Pattern Recognition 43 (2010) 695–705

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Automatic color constancy algorithm selection and combination

S. Bianco, G. Ciocca, C. Cusano*, R. Schettini

DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione), Università degli Studi di Milano–Bicocca, Viale Sarca 336, 20126 Milano, Italy

ARTICLE INFO

Article history:
Received 31 July 2008
Received in revised form 11 June 2009
Accepted 6 August 2009

Keywords:
Color constancy
Image indexing
Classification
Decision forests

ABSTRACT

In this work, we investigate how illuminant estimation techniques can be improved taking into account intrinsic, low level properties of the images. We show how these properties can be used to drive, given a set of illuminant estimation algorithms, the selection of the best algorithm for a given image. The algorithm selection is made by a decision forest composed of several trees on the basis of the values of a set of heterogeneous features. The features represent the image content in terms of low-level visual properties. The trees are trained to select the algorithm that minimizes the expected error in illuminant estimation. We also designed a combination strategy that estimates the illuminant as a weighted sum of the different algorithms' estimations. Experimental results on the widely used Ciurea and Funt dataset demonstrate the effectiveness of our approach.

1. Introduction

Computational color constancy aims to estimate the actual color in an acquired scene disregarding its illuminant. The different approaches can be broadly classified into color invariant and illuminant estimation [2]. The former approaches derive invariant color descriptors from the image data without estimating explicitly the scene illuminant. The latter is actually a two stage procedure: the scene illuminant is estimated from the image data, and the image colors are then corrected on the basis of this estimate to generate a new image of the scene as if it were taken under a known, canonical illuminant. Many illuminant estimation solutions have been proposed in the last few years, although it is known that the problem addressed is actually ill-posed as its solution lacks uniqueness and stability. Moreover, we have recently shown that on large datasets of both synthetic and real images, the best and the worst algorithms do not exist at all (in a set of well known and widely used color constancy algorithms) [1].

Different solutions usually exploit some assumptions about the statistical properties of the expected illuminants and/or of the object reflectances in the scene. Hordley in his survey [2] gives an excellent review of illuminant estimation algorithms and highlights two research areas that are important in the context of improving the performance of color constancy algorithms by making additional measurements at the time of image capture (i.e. use more color pixel information), and by algorithm combining (i.e. using two or more estimations of the illuminants). Another recent research area which shows promising results aims to improve illuminant estimation by using high level visual information on the image content.

The use of content-based image analysis for automatic color correction has been originally proposed by Schröder and Moser [3]. They classify the images into several signal-oriented generic classes (e.g. scene with high color complexity) and, after the class-specific application of a set of color correction algorithms (White Patch and Gray World), they combine the results in a way that they take into account the class-specific reliabilities of each algorithm. Their proposal is based on a hierarchical Bayesian image content analysis consisting of feature extraction and unsupervised clustering. They also suggest that semantic classes (e.g. indoor, outdoor, vegetation scene, mountain scene, etc.) and specific image degradation classes (e.g. under-exposure, strong color cast, etc.) could be used in a similar way. Gasparini and Schettini [4] applied an adaptive mixture of the white balance and Gray World procedures. In order to avoid the mistaken removal of an intrinsic color, regions identified as probably corresponding to skin, sky, sea or vegetation, are temporarily removed from the analyzed image. Van de Weijer et al. [5] proposed high-level visual information to improve illuminant estimation. They modeled the image as a mixture of semantic classes such as grass, skin, road and building and exploited this information to select the best illuminant out of a set of possible ones. They applied several illuminant estimation approaches to compute a set of possible illuminants. For each of them an illuminant color corrected image is evaluated on the likelihood of its semantic content, and the illuminant resulting as the most likely semantic composition of the image

* Corresponding author.
E-mail addresses: bianco@disco.unimib.it (S. Bianco), ciocca@disco.unimib.it (G. Ciocca), cusano@disco.unimib.it (C. Cusano), schettini@disco.unimib.it (R. Schettini).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.08.007

is selected as the illuminant color. They tested their method on a small subset of the Ciurea and Funt database [6], which is composed of a variety of both indoor and outdoor scenes and has shown that their top-down approach on outdoor images works better than any other tested algorithms.

A recent work [7] classifies the images into indoor and outdoor categories. Several state-of-the-art illuminant estimation algorithms are compared against the proposed classification based strategy. Three instantiations of each algorithm are used: one with the algorithm parameters optimally tuned for images in the indoor class, one with parameters tuned for images in the outdoor class and one with parameters tuned to be used indiscriminately on both classes. Each image is classified as indoor or outdoor and according to the class label, the best algorithm with the parameters setting for that class is used in the estimation of the image illuminant. The images for which the classifier is not confident enough are processed using the general purpose parameters. Results show that the performance in the illuminant estimation improves when the image semantic class is taken into account.

The above approaches show that the illuminant estimation can be improved using semantic information on the image content. One drawback of the approaches is that the semantic classes must be accurately chosen in advance; and in no way can these classes be exhaustive or considered representative for all the images, but only for a given task.

Gijsenij and Gevers [8] used natural image statistics to identify the most important characteristics of color images and achieve selection and/or combination of color constancy algorithms. For this, they used the Weibull parameterization on the first-order derivative filter in the x and y direction to capture the image characteristics, they applied a k-means algorithm to cluster the parameters in a set of an heuristically defined number of clusters and then they associated the best-suited color constancy algorithm to each cluster. Unseen images are assigned to the computed clusters, and the best color constancy algorithm for that image is chosen. The authors show that a correlation between the clusters and image categories can be found. A combining strategy which combines the results of color constancy algorithms assigned to neighbor clusters is also considered. Slightly better results with respect to a baseline algorithm are obtained by the combination strategy applied on the five algorithms considered, while the most relevant increment in the performance is obtained with the combination strategy applied on 75 different instantiations of color constancy algorithms.

We investigated here if it is possible to automatically derive the suitability of an illuminant estimation algorithm for a given image by analyzing a set of visual features. To validate this hypothesis we developed an illuminant estimation framework and evaluated its performance on a public available dataset of images. Given a set of illuminant estimation algorithms, the proposed framework determines how the estimation of the illuminant of a given image should be computed. The prediction of the suitability of each algorithm is carried out by an image classifier based on an ensemble of decision trees. The trees are trained to identify the best algorithm in the set considered, on the basis of the values of a set of low-level visual features. For the most part these are general purpose features taken from the pattern recognition and image analysis fields. Some features have been specifically designed for the illuminant estimation problem.

Two illuminant estimation strategies are evaluated: the first selects a single algorithm on the basis of the responses of the trees; the second combines the algorithms according to the ensemble consensus. Using a small set of simple, well known, illuminant estimation algorithms we obtained a significant improvement with respect to the performance of the algorithms considered.

2. Proposed framework

The image values for a Lambertian surface located at the pixel with coordinates (x,y) can be seen as a function $\boldsymbol{\rho}(x,y)$, mainly dependent on three physical factors: the illuminant spectral power distribution $I(\lambda)$, the surface spectral reflectance $S(\lambda)$ and the sensor spectral sensitivities $\mathbf{C}(\lambda)$. Using this notation $\boldsymbol{\rho}(x,y)$ can be expressed as

$$\boldsymbol{\rho}(x,y)=\int_{\omega} I(\lambda)S(x,y,\lambda)\mathbf{C}(\lambda)\,\mathrm{d}\lambda, \tag{1}$$

where ω is the wavelength range of the visible light spectrum, $\boldsymbol{\rho}$ and $\mathbf{C}(\lambda)$ are three-component vectors. Since the three sensor spectral sensitivities are usually, respectively, more sensitive to the low, medium and high wavelengths, the three-component vector of sensor responses $\boldsymbol{\rho}=(\rho_1,\rho_2,\rho_3)$ is also referred to as the sensor or camera $\mathrm{RGB}=(R,G,B)$ triplet.

The goal of color constancy is to estimate the color $\mathbf{I}$ of the scene illuminant, i.e. the projection of $I(\lambda)$ on the sensor spectral sensitivities $\mathbf{C}(\lambda)$:

$$\mathbf{I}=\int_{\omega} I(\lambda)\mathbf{C}(\lambda)\,\mathrm{d}\lambda. \tag{2}$$

Since the only information available are the sensor responses $\boldsymbol{\rho}$ across the image, color constancy is an under-determined problem [9]; and thus further assumptions and/or knowledge are needed to solve it. Typically, some information about the camera being used is exploited, and/or assumptions about the statistical properties of the expected illuminants and surface reflectances. When these assumptions are not fulfilled, the illuminant estimation is expected to be very inaccurate and leads to an erroneous color correction.

In this paper we propose a classification approach which improves the performance of existing color constancy algorithms. Since it is difficult to select an exhaustive and comprehensive set of image categories (using either supervised or unsupervised classification), our approach does not classify the images into high level categories and then process each image with an ad hoc algorithm for that class, but it learns from the images themselves some intrinsic, low level properties that can be used to drive the selection of the best algorithm for that image. That is, the selection of the algorithm is not class-based but feature-based.

Several computational color constancy algorithms exist in the literature, each based on different assumptions. Recently Van de Weijer et al. [10] have unified a variety of algorithms. These algorithms approximate the illuminant color $\mathbf{I}$ by implementing instantiations of the following equation:

$$\mathbf{I}(n,p,\sigma)=\frac{1}{k}\left(\iint |\nabla^n \boldsymbol{\rho}_\sigma(x,y)|^p\,\mathrm{d}x\,\mathrm{d}y\right)^{1/p}, \tag{3}$$

where n is the order of the derivative, p is the Minkowski norm, $\boldsymbol{\rho}_\sigma(x,y)=\boldsymbol{\rho}(x,y)\otimes G_\sigma(x,y)$ is the convolution of the image with a Gaussian filter $G_\sigma(x,y)$ with scale parameter σ, and k is a constant to chosen such that the illuminant color $\mathbf{I}$ has unit length. The integration is performed over all pixel coordinates. In this work, varying the three variables (n,p,σ) we have generated four algorithm instantiations that correspond to well known and widely used color constancy algorithms:

1. Gray World (GW) algorithm [11], which is based on the assumption that the average reflectance in a scene is achromatic. It can be generated setting $(n,p,\sigma)=(0,1,0)$ in Eq. (3).
2. White Point (WP) algorithm [12], also known as Maximum RGB, which is based on the assumption that the maximum reflectance in a scene is achromatic. It can be generated setting $(n,p,\sigma)=(0,\infty,0)$ in Eq. (3).

3. Gray Edge (GE1) algorithm [10], which is based on the assumption that the p-th Minkowski norm of the first order derivative in a scene is achromatic. It can be generated setting $(n,p,\sigma)=(1,p,\sigma)$ in Eq. (3).
4. Second Order Gray Edge (GE2) algorithm [10], which is based on the assumption that the p-th Minkowski norm of the second order derivative in a scene is achromatic. It can be generated setting $(n,p,\sigma)=(2,p,\sigma)$ in Eq. (3).

A fifth algorithm has been considered—the Do Nothing (DN) algorithm—which gives for every image the same estimation for the color of the illuminant, $\mathbf{I}=[1\ \ 1\ \ 1]$.

To select the algorithm to be used with a given image, we used a decision forest tree approach based on the CART methodology where each classifier in the forest votes for one of the illuminant estimation algorithms to be used. The most voted algorithm is then applied to the input image. In the training of the classifier we considered the error costs of the erroneous algorithm's selection. That is, the algorithm selected by the classifier is the one that minimizes the expected error in the illuminant estimation. We also designed a combination approach that estimates the illuminant as a weighted sum of the algorithms' estimations. For each algorithm, its weight is computed using the votes of the classifiers in the forest and is proportional to the number of classifiers that have selected that algorithm. The features used in the classification process are heterogeneous and representative of the image content and can be related to visual properties of the images such as color, texture, composition, etc. For the most part they are taken from the content-based image retrieval research field and have been selected because they are widely used in different applications. We added a few features that are related to the illuminant estimation problem that may be helpful in the algorithm's selection process.

The following two sections describe the classifier and the low-level visual features more in detail.

3. Algorithm selection by decision forests

We decided to adopt classification trees as classifiers because they present several advantages which make them particularly suitable for our problem:

- classification trees can be trained to distinguish an arbitrary high number of classes, so that we are free to consider any set of illuminant estimation algorithms;
- since they do not require any feature normalization or decorrelation, they allow feature vectors to be composed of several heterogeneous features;
- classification trees can exploit information about the a priori probabilities of the classes and their misclassification costs, making it easy to integrate information about the correlation between the errors of different algorithms in the algorithm selection model.

In our approach, decision trees are built according to the Classification and Regression Trees (CART) methodology [13] which has proven to be effective for image classification tasks [14,15]. CART trees are produced by recursively partitioning a set of feature vectors $T=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ labeled with the corresponding correct class $\{y_1,\ldots,y_N\}$, $y_j\in\{1,\ldots,K\}$. The partition is driven by an impurity function which measures the diversity of the classes associated to a set of feature vectors on the basis of the estimated distribution of the classes in that set. The Gini diversity index can be used as the impurity function: $i_{Gini}(P_1,\ldots,P_K)=1-\sum_{j=1}^{k}P_j^2$. The process starts by considering the whole training set T. For each feature j and for each value of the threshold τ, the subsets $T_L=\{\mathbf{x}\in T|\mathbf{x}_j\leqslant\tau\}$, $T_R=T\backslash T_L$ are defined and the decrease in impurity is computed as

$$\Delta I(j,\tau)=i(T)-i(T_L)P_L-i(T_R)P_R, \tag{4}$$

where P_L and P_R are the resubstitution estimates of the probabilities that an element of T falls into the subsets T_L and T_R, respectively; $i(T)$ represents the application of an impurity function i to the resubstitution estimates of the distribution $P(y=1|T),\ldots,P(y=K|T)$ of the classes of the elements of the set T. Among all possible splits (j,τ) the one which maximizes the decrease in impurity is selected. In tree terminology T represents the parent node of the nodes T_L and T_R. The process is recursively repeated for the new nodes T_L and T_R until no further decrease in impurity is possible.

At this point, each terminal node L of the tree is labeled with the class which minimizes the misclassification error:

$$\hat{y}=\arg\max_{j\in\{1,\ldots,K\}}P(y=j|L). \tag{5}$$

New cases are classified by choosing the classes associated to the terminal nodes in which they fall on the basis of the values of the features. Since the trees almost certainly overfit the training data, generalization accuracy is expected to be low. To avoid overfitting, we applied a pruning procedure. Instead of minimizing the misclassification error on the training set, a cost-complexity criterion is considered: given a tree T, its performance is measured by

$$R_\alpha(T)=R(T)+\alpha|T|, \tag{6}$$

where $R(T)$ is the probability of misclassification estimated on the training set, $|T|$ is the size of the tree (the number of terminal nodes), and α is a parameter which weights prediction errors and complexity of the tree. Starting from an initial tree T_0 (corresponding to $\alpha=0$) built as described above and increasing the value of α, the sequence of best subtrees $T_0,T_1,\ldots,T_M$ is considered. Each subtree corresponds to the optimal subtree of T_0 for a range of values of the parameter α. The subtree of the sequence which minimizes the misclassification error on an independent validation set is finally selected.

The pruning process improves the generalization accuracy of the trees. However, a low misclassification error cannot be ensured due to the instability of the training procedure (slightly different training sets could produce completely different trees). To avoid instability, we adopted a multiple classifier approach. Several CART trees are built to form a decision forest [16]. The trees of the forest are first built using different bootstrap replicates of the training set. The complement of each replicate is then used as a validation set to prune the corresponding tree. New cases are classified by majority vote on the output of the trees of the forest.

In our approach, the classifier is used to select the most appropriate illuminant estimation algorithm on the basis of the content of the images. Illuminant estimation algorithms are modeled as classes: an image is of class j if the j-th algorithm is the one which produces the lowest estimation error on that image among the algorithms considered. The feature vector which describes each image is built by computing a set of low-level features (see Section 4) and by concatenating the values obtained.

The straightforward application of the CART training process to this problem leads to poor results. This is due to the fact that some properties of the problem are not taken into account in the formulation:

- some algorithms generally perform better than others;
- the performance of the algorithms are correlated so that the consequences of a non-optimal choice may present a high variability.

The first point is addressed by estimating the a priori probability for each algorithm that it is the best algorithm. They are estimated as

the ratio N_j/N, where N_j is the number of training cases for which the j-th algorithm is the best choice and N is the size of the training set. The a priori probabilities are then used during training to compute the resubstitution estimates of conditional probabilities in (4)–(6).

For each pair of algorithms, Class correlation is modeled by considering, the average difference in performance obtained when one of the two algorithms corresponds to the best choice:

$$c(k|h) = \frac{\sum_{j:y_j=h} e_j^{(k)} - e_j^{(h)}}{|\{j : y_j = h\}|}, \quad h,k \in \{1,\ldots,K\}, \tag{7}$$

where $e_j^{(k)}$ is the error of the k-th algorithm on the j-th training sample. In other words, $c(k|h)$ is the expected cost (i.e. degradation in performance) caused by the choice of algorithm k when algorithm h is the best choice.

Misclassification costs are used during training to influence pruning and label assignment. Eq. (5) is replaced by

$$\hat{y} = \arg\min_{j\in\{1,\ldots,K\}} \sum_{h=1}^{K} c(j|h)P(y=h|L), \tag{8}$$

while $R(T)$ is now the estimated average cost of the decisions of tree T:

$$R(T) = \sum_{L\in\tilde{T}} \left(\min_{j\in\{1,\ldots,K\}} \sum_{h=1}^{K} c(j|h)P(y=h|L) \right) P(L), \tag{9}$$

where $\tilde{T}$ is the set of the leaves of T and $P(L)$ is the resubstitution estimate of the probability that a case falls in the leaf L.

Class correlation is also exploited by using the *twoing* criterion as the impurity function: the impurity of a node is computed by dividing the set of the classes into two macro-classes, and then by applying the Gini diversity index to the distribution of the two macro-classes. In practice, the twoing criterion is implemented by substituting (4) with the following expression:

$$\Delta I_{twoing}(j,\tau) = \frac{P_L P_R}{4} \left[\sum_{k=1}^{K} |P(y=k|T_L) - P(y=k|T_R)| \right]^2. \tag{10}$$

The effect of the twoing criterion is that during the first steps of the tree growing process few resources are wasted trying to discriminate between similar classes (i.e. highly correlated algorithms). The discrimination of such classes occurs only near the leafs.

4. Image features

Since an image conveys information at different levels, the use of different features at the same time is a necessary requisite if we want to capture most of the image information. There is no single "best" representation of the content of an image, but only multiple representations which characterize the content from different perspectives. In the literature many features exist to be used in describing the image content [17–20].

We have considered two groups of low level features: general purpose features and problem-dependent features. The general purpose features are features that can be used on a large range of applications since they do not capture characteristic of the images that are problem specific. The features in this category that we have selected are: color histogram, edge direction histogram, statistics on the wavelet coefficients, and color moments.

Problem-dependent features try to capture properties of the images that can be useful in improving the performance of the task under consideration. Since several algorithms of illuminant estimation rely on some image characteristics or make certain assumptions about the color of the images, some of the problem dependent features have been chosen to exploit these properties. For example, the extent of a color cast in an image is a feature that may be useful in the color balancing problem. A strong cast may be an indication that a particular illuminant is present. The number of different colors is an indication of color variability, and thus that the Gray World assumption can be justified for the image under consideration. Conversely, an image with very few colors may not be reliably processed for balancing since it may lack a sufficient amount of information to estimate the White Point. Several algorithms rely on the edges found in the image, so we have included features that extract edge information at different perspectives.

Summarizing, the problem-dependent features chosen are: the number of different colors, the percentage of color components that are clipped to the highest and lowest value that can be represented in the image color space, a cast index representing the extent of the presence of a color cast in the image and the magnitudes of the edges.

Note that all these features have been chosen independently from their usefulness in the image classification process. They have been chosen uniquely for their ability to describe the content of an image. Some of them are not strictly independent in the sense that similar properties of the images are evaluated using different features. The aim of the classifier is to choose the features as well as which specific components in a feature are more relevant to discriminate between the classes selected for the problem under analysis. Moreover, while all the features must be computed for the images in the training sets, only the features actually chosen and used by the classifier need to be computed for the images in the test sets and for new images to be processed. This approach is made possible by the use of CART trees as classifiers. Other classification methodologies (such as support vector machines and neural networks) would have required a complex feature selection (and normalization) step. Table 1 summarizes the features considered.

Table 1
Summary of the features used to describe the images.

Name	No. components	Category
YCbCr color moments	6	Color
RGB color histogram	27	Color
Number of colors	1	Color
Cast indexes	2	Color
Color clipping	8	Color
Edge magnitude histogram	5	Edges
Edge direction histogram	18	Edges
Wavelet statistics	20	Texture

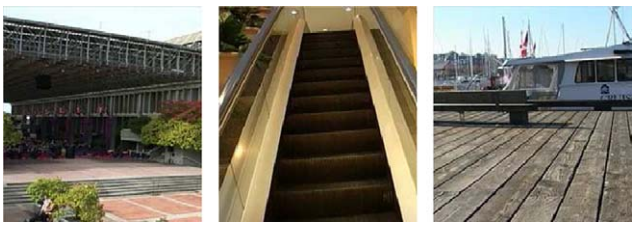

Fig. 1. Examples of images of man-made structures.

4.1. Edge direction histogram

Edges are a clue about the subject depicted in an image. Strong edges can be found in buildings, roads, and other man-made structures. These edges usually have directions in a definite pattern (see Fig. 1). On the other hand, pictures of natural scenes usually do not show strong edges and since the subject has no clear structure they do not show a specific pattern (see Fig. 2). Edge direction histogram

Fig. 2. Examples of images of natural scenes.

Fig. 3. Examples of images with weak (left) and strong (right) edge magnitudes.

can be used to determine the edge structures within an image and thus allow us to distinguish between different image classes. Edges are computed applying a Derivate of the Gaussian filter with $\sigma=1$ on the luminance image in both the x and y directions (G_x, G_y) and then the edge orientation at edge position (x,y) are computed as follows:

$$\theta(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right). \tag{11}$$

The orientations are quantized into 18 bins each corresponding to angles of intervals of 10°. The quantized orientations are then used to compute an edge direction histogram of 18 components. Only the orientations belonging to edges whose gradient's magnitude is above a given threshold (0.50 in our case) are taken into account. This ensures that only edges with sufficient strength are used in computing the direction histogram.

4.2. Edge strengths

In order to capture the relevance of the edges we compute a histogram of edge magnitudes. The edges are detected as in the case of the edge direction histogram. The magnitude is quantized into 5 bins corresponding to the following intervals: $[0, 0.25)$, $[0.25, 0.50)$, $[0.50, 0.75)$, $[0.75, 1.0)$, $[1.0, \infty)$. Two examples of images with different edge strengths are shown in Fig. 3.

4.3. Color histogram

Color histogram is one of the most widely used image descriptors [21,22] and represents the color distribution of the image. It possesses several useful properties that make it a robust visual feature such as compactness, invariance and robustness with respect to the geometric transformation of the original image like rotation and scale. In order to compute the histogram we quantized the RGB color space by uniformly dividing each color axis into three intervals. The RGB cube is thus subdivided into 27 smaller cubes and each of the original colors is mapped to the cube which it falls into.

4.4. Wavelet statistics

Information about the textures and structures within the image can be obtained using a wavelet decomposition. This technique is often used in content-based retrieval for similarity retrieval, target search, compression, texture analysis, biometrics, etc. [23,24]. In multiresolution wavelet analysis, at each level of resolution (i.e. at each application of the wavelet decomposition) we have four bands containing different information obtained by applying a combination of a low pass filter (L) and a high pass filter (H). Specifically, the information corresponds to a low-pass filtered version of the processed image (LL band), and three bands of details that roughly correspond to the horizontal edges (LH band) of the original images, the vertical edges (HL band) and the diagonal edges (HH band). Each band is a matrix of values, one fourth the size of the original image.

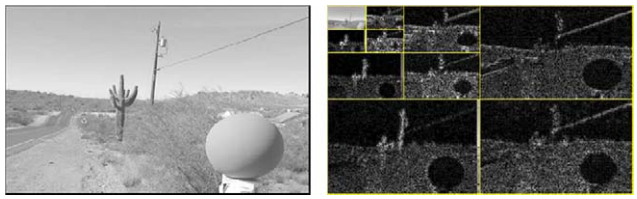

Fig. 4. A three-iteration Daubechies wavelet decomposition.

Wavelet decomposition is applied recursively to the LL band. The resultant decomposition will contain information, i.e. details, at the lower resolution. The process can be repeated until the LL sub-band cannot be further processed or until a given number of wavelet decomposition applications is reached. Different filters can be used to produce the bands of the wavelet analysis [25] e.g. Harr, Daubechies, Symlet, Biort, etc.

For our purposes the wavelet statistics features are extracted from the luminance image using a three-iteration Daubechies wavelet decomposition, producing a total of 10 bands as shown in Fig. 4. The mean and variance of the absolute values in each band are then computed as band statistics. These feature values represent the energy i.e. the amount of information within each band and provide a concise description of the image's content. This feature thus composed of 20 (two energy values for each of the 10 bands) components.

4.5. YCbCr color moments

To describe the color distribution of an image, we computed the first two central moments, mean, and standard deviation of each color channel of the YCbCr color space, derived by transforming the R, G, and B color coordinates. The color distribution of an image can, in fact, be considered a probability distribution and can therefore be characterized uniquely by its central moments alone, as can any probability distribution [26]. The choice of the YCbCr color space allows the separation of the luminance component from the chrominance components in a simple way using a linear transformation. The color transformation used is that defined in the ITU-R Recommendation BT.601 [27]:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} + \begin{bmatrix} 65.74 & 129.06 & 25.06 \\ -37.95 & -74.50 & 112.44 \\ 112.44 & -94.15 & -18.29 \end{bmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}, \tag{12}$$

where r, g, b are the RGB coordinates normalized in the range $[0, 1]$. This feature is composed of six values (two statistics for each of the three color channels).

4.6. Number of colors

The number of distinct colors is related to the color range of the image. Since several illuminant estimation algorithms are based on the Gray World assumption, the color range is an indication of whether this assumption holds true for the given image or not. The

Fig. 5. Examples of images with many different colors. Left image: 10 782 colors with average color (122, 123, 121). Right image: 13 882 colors with average color (107, 106, 110). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. Examples of images with few different colors. Left image: 5380 colors with average color (150, 97, 47). Right image: 7538 colors with average color (96, 126, 150). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

actual values of the pixels colors may impede the occurrence of the assumption, but if an image contains many different colors then the average color is likely to be a gray value. Two examples can be seen in Figs. 5 and 6. To remove small variations in the color appearance and thus limit the influence of noise in the computation of the feature, the RGB color channels are quantized by considering only the six most significant bits. Thus, the maximum number of different colors that can be discriminated is $2^6 \times 2^6 \times 2^6 = 262\,144$.

4.7. Clipped color components

To take into account the extent of highly saturated color pixels, we compute the percentage of pixels whose color components are clipped to the maximum value that can be represented. For digital images with eight-bit color channel representation, the maximum value is 255. We discriminate between pixels with zero, one, two or all three color components clipped to the maximum value. In total the histogram of clipped color components is composed of eight bins: 0 clipped components, 1 clipped component (either R, or G or B), 2 clipped components (either R and G, or R and B, or G and B) and 3 clipped components (R and G and B).

The values in the histogram bins are normalized with respect to the total number of pixels in the image, so that the histogram represents a probability density distribution.

4.8. Cast indexes

The cast index is aimed at identifying the presence of a relevant cast within the image; and it is inspired by the work done in [4], where the cast is detected and classified into several classes according to its relevance. The basic idea of the cast detection is that the color distribution of an image can be analyzed by converting it into a suitable color space and using statistical tools to characterize the presence of the cast. In [4] the presence of a color cast is used to process images producing more pleasing images; that is, images that users perceive as more natural than the original one.

In this work, we made small modifications of the original formulation since the problem we face is different from the one in [4]. We changed the color space representation from the CIELAB to YCbCr, since the former depends on the knowledge of the White Point of the scene. Moreover, we only considered the cast indexes disregarding the final cast classification.

An image with a very strong cast will show one definite peak within the CbCr plane, far away from the neutral axis corresponding to the color cast.

The means and variances of the Cb and Cr components (μ_{Cb} and μ_{Cr}, σ^2_{Cb} and σ^2_{Cr}) are used to compute the color equivalence circle center $C=(\mu_{Cb},\mu_{Cr})$ and its radius $r=(\sigma^2_{Cb}+\sigma^2_{Cr})^{1/2}$, as well as the two cast indexes $D=\mu-\sigma$ and $D_\sigma=D/\sigma$ where $\mu=(\mu^2_{Cb}+\mu^2_{Cr})^{1/2}$. D is a measure of how far the color distribution is from the neutral axis (i.e. from $(0,0)$ in the CbCr coordinates), indicating thus the presence of a cast. D_σ quantifies the strength of the cast. An example of an image showing a strong cast is depicted in Fig. 7. Fig. 8 shows an example of an image without cast.

5. Experimental results

To evaluate our approach we measured its performance on a subset of the dataset of images presented by Ciurea and Funt [6] which is commonly used in the evaluation of color constancy algorithms as it is labeled with the ground truth illuminants. In this dataset 15 digital video clips were recorded (at 15 frames per second) in different settings such as indoor, outdoor, desert, markets, cityscape, etc. for a total of 2 h of videos. From each clip, a set of images was extracted, resulting in a dataset of more than 11 000 images. A gray sphere appears in the bottom right corner of the images and was used to estimate the true color of the scene illuminant. Since the dataset sources were video clips, the images extracted show high correlation. To remove this correlation, only a subset of images should be used from each set. Taking into account that the image sets came from video clips, we applied a two stage video-based analysis to select the image to be included in the final illuminant dataset.

In the first stage, a video clip is reconstructed from each set of images, removing the right part of the images containing the gray sphere. The video clip is fed to a key frame extraction algorithm [28], which dynamically selects a set of candidate images by analyzing the visual content of consecutive frames. Clips showing high variability in their pictorial content will have a high number of images extracted, while clips showing little or no variability will have only a single image extracted.

In the second stage, we further processed the extracted images with a visual summary post-processing algorithm [29]. A hierarchical clustering algorithm further removes redundancies within the set by iteratively eliminating pictorially similar images until the number of remaining images is equal to the one required.

As a trade-off between the number of images to be included in the dataset and the correlation problem, we set the parameters of the key frame extraction algorithm so that it over-extracts images from the video clip, and so that the number of images that must be included in the final dataset corresponds to 10% of the clip size. The parameters of the visual post-processing algorithm are thus set accordingly. With these settings the final dataset consists of 1135 images. More details about the dataset extraction can be found in [7].

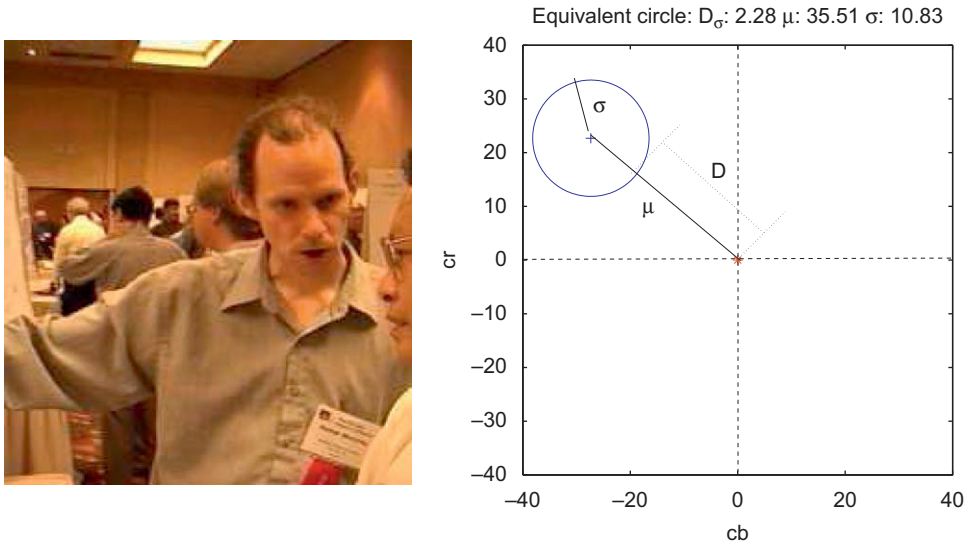

Fig. 7. Example of an image with a strong color cast. The equivalence circle is compact and far from the neutral axis.

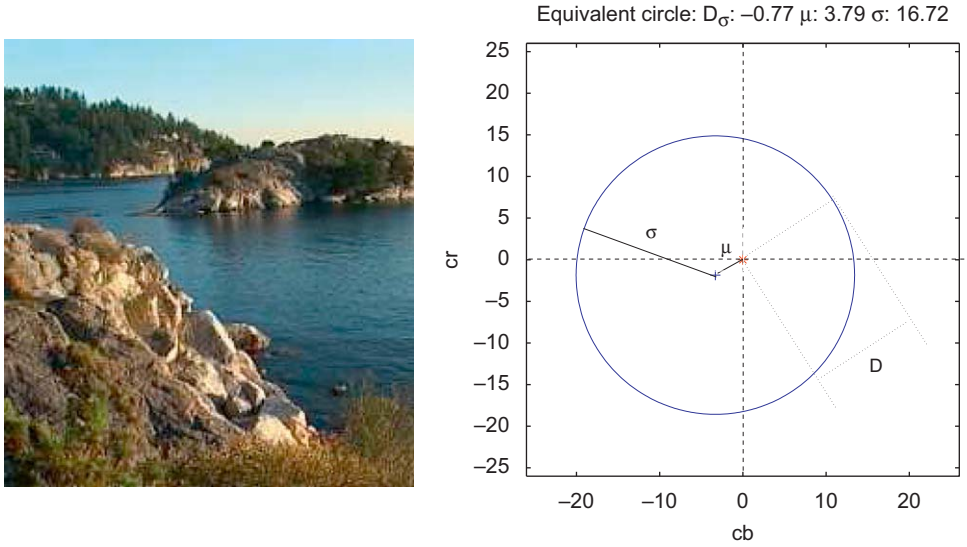

Fig. 8. Example of an image without color cast. The equivalence circle is large and close to the neutral axis.

The final dataset has been randomly subdivided into a training set of 340 images (about 30% of the dataset) and a test of 795 images. The training set has been used to:

- find the best parameters of the illuminant estimation algorithms;
- make an estimate of the a priori related to the algorithms (i.e. the probability that an algorithm is the best one);
- estimate the matrix of misclassification costs (7).

A cross validation on the test set has been adopted to train and evaluate the decision forest and to assess the overall performance of the strategy.

5.1. Performance evaluation

In order to evaluate the performance of the algorithms considered, we have to define an error measure. Since in estimating the scene illuminant it is more important to estimate its color than its overall intensity, the error measure has to be intensity-independent. As suggested by Hordley and Finlayson [30], we use as error measure the angle between the RGB triplets of the illuminant color ($\boldsymbol{\rho}_w$) and the algorithm's estimate of it ($\hat{\boldsymbol{\rho}_w}$):

$$e_{ANG} = \arccos\left(\frac{\boldsymbol{\rho}_w^T \hat{\boldsymbol{\rho}_w}}{\|\boldsymbol{\rho}_w\| \|\hat{\boldsymbol{\rho}_w}\|}\right). \tag{13}$$

Hordely and Finlayson [30] showed that a good descriptor for the angular error distribution is the median error. To verify if the performances of different algorithms are statistically different, a test which is able to compare the whole error distribution of different algorithms is needed. Since standard probability models cannot represent underlying errors well, we need a test that does not make any a priori assumptions about the underlying error distributions. To compare the performance of two color constancy algorithms in addition to the median angular error, we have used the Wilcoxon sign test (WST) [31]. Let X and Y be random variables representing the angular errors between the illuminant estimations of the two algorithms and the real illuminants; let μ_X and μ_Y be the median

values of such random variables. The Wilcoxon signed-rank test can be used to test the null hypothesis $H_0 : \mu_X = \mu_Y$. To test H_0, we consider the differences of independent error pairs $(X_1 - Y_1), \ldots, (X_N - Y_N)$ for N different images. We rank these error pairs according to their absolute differences. Ranks are signed considering whether the corresponding error pair difference is positive or negative. If H_0 is correct, the sum of the signed ranks W will approximate zero. If W is much larger or smaller than zero, the alternative hypothesis $H_1 : \mu_X \neq \mu_Y$ is true. We can test H_0 against H_1 at a given significance level α. We reject H_0 and accept H_1 if the probability of observing the error differences we obtained is less than or equal to α. In this work, we have used the alternative hypothesis $H_1 : \mu_X < \mu_Y$ as implemented in the Matlab statistical package, with a significance level $\alpha = 0.01$. Comparing each color constancy algorithm with all the others, we generated a score representative of the number of times that the null hypothesis H_0 has been rejected for the given algorithm, i.e. the number of times that the performance of the given algorithm has been considered to be better than the others.

5.2. Tuning of the color constancy algorithms

Two of the color constancy algorithms considered, (GE1 and GE2), needed a training phase to opportunely tune the parameters (n, p, σ). As a training set, we used the same 300 images used in [7] in order to make the results easily comparable. Starting from the 340 training images, 40 have been discarded in order to balance the frequency of indoor and outdoor images. The performances of the algorithms are evaluated using the median angular error. Since the median error is a nonlinear statistic, we needed a multidimensional nonlinear optimization algorithm: our choice was to use a Pattern Search Method (PSM). PSMs are a class of direct search methods for nonlinear optimization [32]. PSMs are simple to implement and do not require any explicit estimate of derivatives. Furthermore, global convergence can be established under certain regularity assumptions of the function to minimize [33].

The general form of a Pattern Search Method can be described in the following way. At each step k, we have the current iterate $\mathbf{x}_k$, a set D_k of search directions, and a step-length parameter Δ_k. Usually the set D_k is the same for all iterations. For each direction $\mathbf{d}_k \in D_k$, we set $\mathbf{x}^+ = \mathbf{x}_k + \Delta_k \mathbf{d}_k$ (the "pattern") and we examine $f(\mathbf{x}^+)$ where f is the function to be minimized. If $\exists \mathbf{d}_k \in D_k : f(\mathbf{x}^+) < f(\mathbf{x}_k)$, we set $\mathbf{x}_{k+1} = \mathbf{x}^+$ and $\Delta_{k+1} = \alpha_k \Delta_k$ with $\alpha_k > 1$; otherwise, we set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and $\Delta_{k+1} = \beta_k \Delta_k$ with $\beta_k < 1$. The algorithm stops when step Δ_k is smaller than a fixed threshold, or when the maximum number of iterations has been reached. In this work we have chosen to fix the maximum number of iterations $n = 50$, $\alpha_k = \alpha = 2$, $\beta_k = \beta = 0.5$, $D_k = \{NW, N, NE, E, SE, S, SW, W\}$, and $\Delta_0 = 0.1$. The same starting point has been chosen for the two algorithms that needed a training phase (GE1 and GE2): $\mathbf{x}_0 = (p_0, \sigma_0) = (1, 0)$. The optimal values of the parameters found by the PSM are $(p, \sigma) = (1.10, 1.08)$ for the GE1 and $(p, \sigma) = (1.55, 1.83)$ for the GE2.

5.3. Decision forest training and evaluation

To train the decision forest and to evaluate the performance of our strategy we adopted a cross validation procedure. First, the five illuminant estimation algorithms are applied to the whole dataset and their angular errors are computed. The first step of the training process for the decision forest consists in the estimation of the a priori probability for each algorithm that is the best choice, and of the matrix of misclassification costs. These values, estimated on the 340 images of the training set, are reported in Tables 2 and 3. In more than one third of the images, the Gray World algorithm corresponds to the best choice. In another 30% of the cases the images are already well balanced, and thus the "Do Nothing" algorithm produces the most accurate illuminant estimation. The remaining one third of the images are best processed by the GE1 or the GE2 algorithms. For less than 4% of the images the WP algorithm corresponds to the best choice. The matrix of misclassification costs tell us that the results of the GE1 and GE2 algorithms show a high correlation. Detecting the images for which the GW algorithm should be used is crucial. In fact, in these cases the errors of the other algorithms are rather high.

Table 2
A priori probabilities, corresponding to the five illuminant estimation algorithms, estimated on the images of the training set.

Algorithm	Probability
DN	0.33
GW	0.34
WP	0.04
GE1	0.12
GE2	0.17

Table 3
Matrix of the misclassification costs estimated on the images of the training set (7).

Best algorithm	Predicted algorithm				
	DN	GW	WP	GE1	GE2
DN	0.00	10.90	1.98	6.41	4.10
GW	8.43	0.00	5.67	4.13	6.28
WP	0.50	10.19	0.00	4.93	2.68
GE1	2.80	5.48	2.29	0.00	0.77
GE2	2.86	6.18	1.89	0.67	0.00

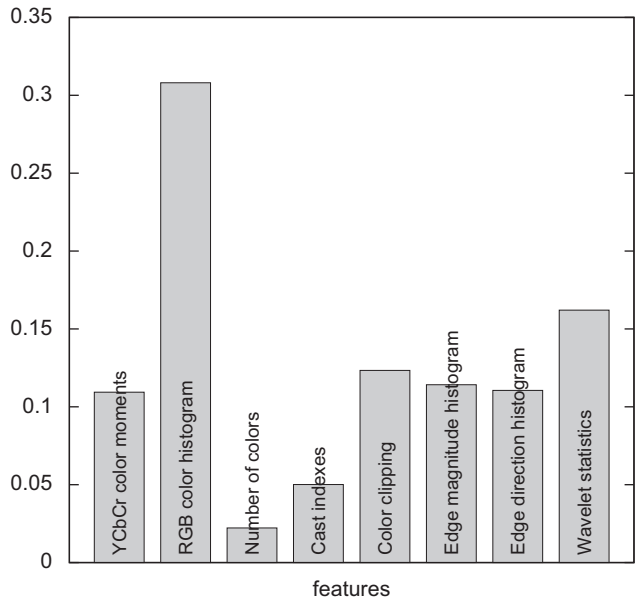

Fig. 9. Histogram of the occurrences of the features in the splits of the trained trees.

At this point, a 10-fold cross validation is used to train and to evaluate our strategy. The test set is randomly partitioned into 10 subsets. Then a decision forest composed of 30 classification trees is trained on all the images of the dataset (including the training set) with the exclusion of a single subset of the test set, which is finally used to measure the performance of the decision forest. The procedure is repeated 10 times, one for each subset of the test set. The results of the 10 forests are finally merged.

Fig. 9 reports the distribution of the occurrences of the features within the splits of the 10 forests. All the features have been used. The RGB histogram is used in more than 30% of the splits; however,

Table 4
Confusion matrix of the decision forest used for algorithm selection, estimated on the images of the test set.

Best algorithm	Predicted algorithm				
	DN	GW	WP	GE1	GE2
DN	0.85	0.06	0.01	0.04	0.04
GW	0.24	0.61	0.01	0.10	0.05
WP	0.37	0.00	0.11	0.37	0.15
GE1	0.39	0.29	0.04	0.17	0.11
GE2	0.45	0.15	0.02	0.13	0.26

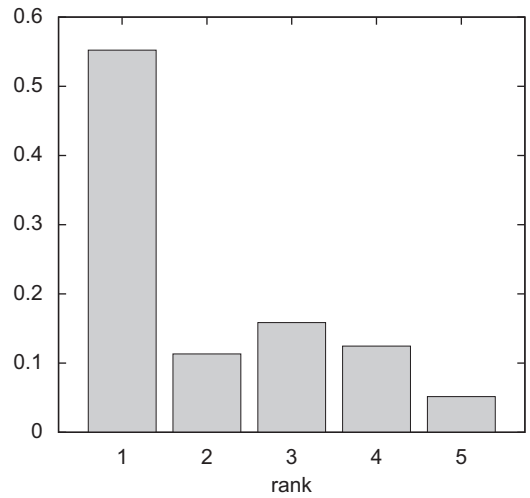

Fig. 10. Distribution of the rank of the algorithm selected by the decision forest on the images of the test set. The first bar represents the fraction of test images for which the best algorithm is selected; the second bar represents the fraction of cases in which the second best is selected, and so on.

it must be considered that it is the features with the highest number of components. Very compact features, such as the number of colors and cast indexes are important as well, if we consider that they are formed by only one and two components, respectively.

Table 4 shows the confusion matrix obtained on the test set. Each row corresponds to an algorithm and reports the distribution of the output of the decision forest estimated on the subset of the test set for which that algorithm is the best choice. Most of the images for which the DN algorithm is the best choice are correctly classified (85% of accuracy). For the other algorithms the correct classification rate ranges from 61% (GW) to 11% (WP). However, considering the a priori distribution of the five algorithms, the aggregated classification accuracy is about 55%, as shown in Fig. 10 which reports the histogram of the rank corresponding to the choice of the decision forest on the test set. The best algorithm is chosen 55% of the time, the second best algorithm is chosen 11% of the time; and the frequency of the selection of the third, the fourth, and the worse algorithm are 16%, 12%, and 5%, respectively. It should be considered that the forest has not been trained with the aim of finding the best algorithm, but with the aim of finding the algorithm with the lowest expected error, taking into account the errors determined by misclassifications. This means that the performance of the decision forest should not be evaluated in terms of classification accuracy, but in terms of the angular error of the selected algorithms. Fig. 11 reports the distribution of the loss determined by the choice of the decision forest with respect to the best algorithm. In more than 70% of test cases this loss is below 1° of angular error with respect to the best algorithm. The average angular error of our Classification-based Algorithm Selection (CAS) strategy is about 4.76°, while the median

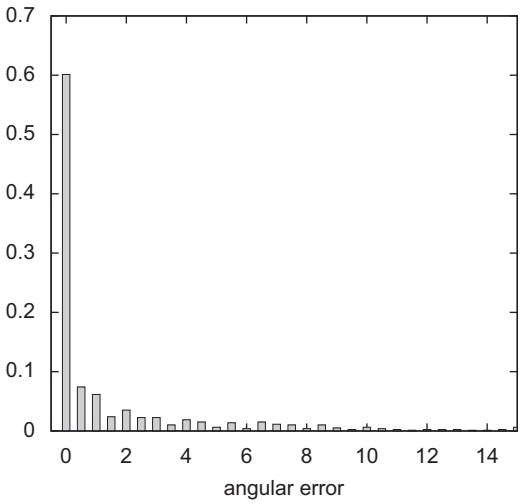

Fig. 11. Distribution of the difference in angular error between the algorithms selected by the decision forest and the best choice for each image of the test set.

Table 5
Summary of the results obtained on the test set by the Classification-based Algorithm Selection (CAS) strategy, compared with the performance of the five simple algorithms and with the results obtained by an algorithm selection strategy based on semantic classification [7].

Algorithm	Median	Mean	WSTs
DN	6.05	8.07	0
GW	5.95	7.27	0
WP	5.48	7.45	2
GE1	4.47	5.84	4
GE2	4.65	6.23	3
CAS	**3.21**	**4.76**	**6**
Semantic	3.54	4.89	5
Ideal classifier	2.31	3.27	–

The best score for each column are reported in bold.

angular error is about 3.21°. These results are compared in Table 5 with those obtained by the five single algorithms. A comparison with the results of a semantic driven approach [7] (on the same data) is also reported. The performance of our approach is clearly superior to that of the single algorithms and of the semantic-based approach, at least on the dataset we considered. It is interesting to note that DN is the worst algorithm on average. However, in about 33% of the cases it is the best choice. This means that in the remaining 67% of the images its error is very high. Thus, from a color correction point of view, detecting which images need to be corrected and which do not is crucial. Our selection strategy seems quite effective in doing this (see the confusion matrix in Table 4).

In order to determine which part of the error is due to the illuminant estimation algorithms and which part should be accounted to classification errors, we compared our selection strategy with a strategy based on an ideal classifier. The ideal classifier selects for each image the best algorithm among the five considered. Using the ideal classifier we obtained a median angular error of 2.31° on the test set. This means that the performance of our strategy may be improved up to 0.9° of median angular error.

In order to obtain better illuminant estimations, a common approach is to combine the results of several different algorithms. Cardei and Funt [34] obtained good illuminant estimation by combining the results of Gray World, White Patch and neural net methods, considering both linear and non-linear committee methods. Schaefer et al. [35] introduced a combined physical and statistical color constancy algorithm that integrates the statistics-based Color by

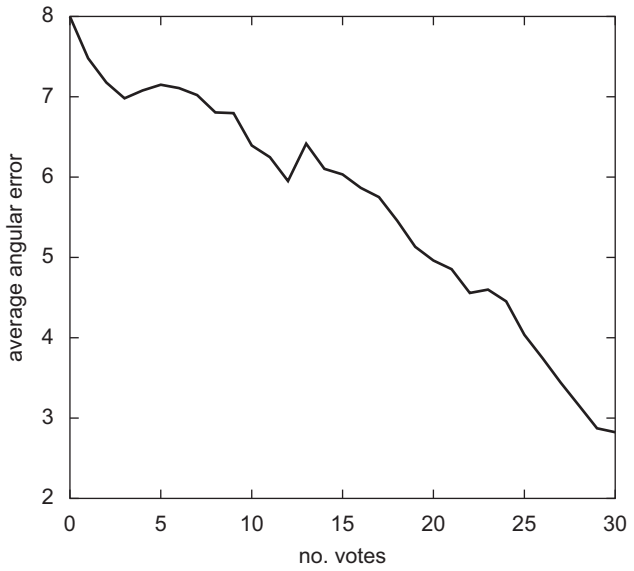

Fig. 12. Average angular error obtained by the five illuminant estimation algorithms on the images of the set, as a function of the number of votes received by the trees of the decision forest.

Table 6
Summary of the results obtained on the test set by the Classification-based Combining (CAC) strategy compared with the performance of other popular combining methods.

Algorithm	Median	Mean	WSTs
AVG	4.66	5.99	0
LMS	4.12	5.29	2
N2M	4.79	5.82	0
CAC	**3.04**	**4.46**	**3**

The best score for each column are reported in bold.

Correlation method with a physics-based technique, based on the dichromatic reflectance model, using a weighted combination of their likelihoods for a given illumination set and taking the maximum likelihood entry.

In our Classification-based Algorithms Combination (CAC) strategy the five algorithms are linearly combined, using as weights the consensus of the decision forest; that is, each algorithm is weighted proportionally to the number of trees which "voted" for it. The number of votes is, in fact, related to the error of the algorithms, as shown in Fig. 12 which reports the average angular error (on the test set) obtained by the algorithms as a function of the votes that they received. The higher the number of votes is, the lower the error is. Table 6 reports the results obtained by the combination strategy and compares it with other combining methods. The first combining method considered (AVG), simply averages the results of the estimations given by the five algorithms considered [34]. The second one (LMS) is a weighted average of the outputs of the individual algorithms [34]. The weights were optimized in the least mean squares sense. This combining method was trained and tested using the same 10-fold cross validation used before. The last combining method considered (N2M), averages the outputs of the three individual algorithms which gave the closest illuminant estimations, automatically excluding the two that gave the furthest estimations [1]. Combining is more effective than simple selection, in fact, it obtained a median angular error of 3.04° (versus 3.21 of the selection strategy). Simple combining methods are clearly outperformed.

6. Conclusions

We presented a framework for automatic illuminant estimation based on the selection or combination of simple algorithms. In this work we used a set of state of the art algorithms. The proposed strategies can be used as it is with any set of color constancy algorithms.

To improve illuminant estimation accuracy, a decision forest is trained to identify the best algorithm for a given image. The solutions investigated here included: a Classification-based Algorithm Selection strategy which applies the algorithm with minimal expected error, and a Classification-based Algorithms Combination strategy which linearly combines the algorithms by weighting them on the basis of the consensus of the trees of the decision forest. The decisions of the classifier are based on a set of low-level features taken from the literature or specifically designed for the problem.

We tested the strategies on a suitable subset of the widely used Funt and Ciurea dataset. For this, a method for extracting uncorrelated images from the dataset is used. Our results demonstrate that our approach outperforms the other algorithms selection and combination strategies considered.

We plan to further investigate this topic, including additional illuminant estimation algorithms and to experiment with other machine learning tools. We are also considering to extend the framework in order to exploit both visual and semantic information.

Our approach defines a framework, which given a set of algorithms for a given imaging problem, seems to be able to devise an effective strategy to solve the problem identifying equivalence classes in the images so that each class can be processed with the best performing algorithm. To verify the generality of the approach, we will evaluate it against other imaging problems.

References

[1] S. Bianco, F. Gasparini, R. Schettini, A consensus based framework for illuminant chromaticity estimation, Journal of Electronic Imaging 17 (02) (2007) 023013.
[2] S.D. Hordely, Scene illuminant estimation: past, present, and future, Color Research & Application 31 (4) (2006) 303–314.
[3] M. Schröder, S. Moser, Automatic color correction based on generic content-based image analysis, in: Proceedings of the Color Imaging Conference, vol. 9, 2001, pp. 41–45.
[4] F. Gasparini, R. Schettini, Color balancing of digital photos using simple image statistics, Pattern Recognition 37 (6) (2004) 1201–1217.
[5] J. van de Weijer, C. Schmid, J. Verbeek, Using high-level visual information for color constancy, in: Proceedings of the IEEE 14th International Conference on Computer Vision, 2007, pp. 1–8.
[6] F. Ciurea, B. Funt, A large image database for color constancy research, in: Proceedings of the IS&T/SID 11th Color Imaging Conference, 2003, pp. 160–164.
[7] S. Bianco, G. Ciocca, C. Cusano, R. Schettini, Improving color constancy using indoor-outdoor image classification, IEEE Transactions on Image Processing 17 (12) (2008) 2381–2392.
[8] A. Gijsenij, T. Gevers, Color constancy using natural image statistics, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[9] B. Funt, K. Barnard, L. Martin, Is machine colour constancy good enough?, in: Proceedings of the 5th European Conference on Computer Vision, 1998, pp. 445–459.
[10] J. van de Weijer, T. Gevers, A. Gijsenij, Edge-based color constancy, IEEE Transactions on Image Processing 16 (9) (2007) 2207–2214.
[11] G. Buchsbaum, A spacial processor model for object color perception, Journal of Franklin Institute 310 (1980) 1–26.
[12] V. Cardei, B. Funt, K. Barndard, White point estimation for uncalibrated images, in: Proceedings of the IS&T/SID 7th Color Imaging Conference, 1999, pp. 97–100.
[13] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth and Brooks/Cole, 1984.
[14] R. Schettini, C. Brambilla, G. Ciocca, A. Valsasna, M. De Ponti, A hierarchical classification strategy for digital documents, Pattern Recognition 35 (2002) 1759–1769.
[15] R. Schettini, C. Brambilla, C. Cusano, G. Ciocca, Automatic classification of digital photographs based on decision forests, International Journal of Pattern Recognition and Artificial Intelligence 18 (5) (2004) 819–845.
[16] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.
[17] R. Schettini, G. Ciocca, S. Zuffi, Indexing and retrieval in color image databases, in: Color Imaging Science: Exploiting Digital Media, 2002, pp. 183–211.
[18] S. Antani, R. Kasturi, R. Jain, Survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video, Pattern recognition 35 (2002) 945–965.

[19] J.P. Eakins, Towards intelligent image retrieval, Pattern Recognition 35 (2002) 3–14.
[20] T. Sikora, The MPEG-7 visual standard for content description—an overview, IEEE Transaction on Circuits and System for Video Technology 11 (6) (2001) 696–702.
[21] M.J. Swain, D.H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.
[22] Y. Gong, C.H. Chuan, G. Xiaoyi, Image indexing and retrieval using color histograms, Multimedia Tools and Applications 2 (1996) 133–156.
[23] F. Idris, S. Panchanathan, Storage and retrieval of compressed images using wavelet vector quantization, Journal of Visual Languages and Computing 8 (1997) 289–301.
[24] P. Scheunders, S. Liven, G. Van de Wouwer, P. Vautrot, D. Van Dyck, Wavelet-based texture analysis, International Journal Computer Science and Information Management 1 (2) (1997) 22–34.
[25] A. Mojsilovic, D. Rackov, M. Popovic, On the selection of an optimal wavelet basis for texture characterization, IEEE Transaction on Image Processing 9 (12) (2000) 2043–2050.
[26] M.A. Stricker, M. Orengo, Similarity of color images, in: Proceedings of the SPIE Storage and Retrieval for Image and Video Databases III Conference, 1995, pp. 381–392.
[27] ITU-R Rec. BT.601, Encoding parameters of digital television for studios, ITU, Geneva, Switzerland, 1995.
[28] G. Ciocca, R. Schettini, An innovative algorithm for key frame extraction in video summarization, Journal of Real-Time Image Processing 1 (1) (2006) 69–88.
[29] G. Ciocca, R. Schettini, Supervised and unsupervised classification post-processing for visual video summaries, IEEE Transactions on Consumer Electronics 2 (52) (2006) 630–638.
[30] S.D. Hordley, G.D. Finlayson, Re-evaluating color constancy algorithms, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 76–79.
[31] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 80–83.
[32] R.M. Lewis, V. Torczon, Pattern search methods for linearly constrained minimization, SIAM Journal on Optimization 10 (2000) 917–941.
[33] R.M. Lewis, V. Torczon, On the convergence of pattern search algorithms, SIAM Journal on Optimization 7 (1997) 1–25.
[34] V.C. Cardei, B. Funt, Committee-based colour constancy, in: IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications, 1999, pp. 311–331.
[35] G. Schaefer, S. Hordley, G. Finlayson, A combined physical and statistical approach to colour constancy, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2005, pp. 148–153.

About the Author—SIMONE BIANCO obtained the B.Sc. and the M.Sc. degrees in Mathematics from the University of Milan-Bicocca, Italy, respectively, in 2003 and 2006. He is currently a Ph.D. student in Computer Science at DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione) of the University of Milano-Bicocca, working on image processing. The main topics of his current research concern digital still cameras processing pipelines, color space conversions, optimization techniques and characterization of imaging devices.

About the Author—GIANLUIGI CIOCCA took his degree (Laurea) in Computer Science at the University of Milan in 1998, and since then he has been a fellow at the Institute of Multimedia Information Technologies of the Italian National Research Council, where his research has focused on the development of systems for the management of image and video databases and the development of new methodologies and algorithms for automatic indexing. He is currently an assistant professor in computer science at DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione) of the University of Milano-Bicocca, working on video analysis and abstraction.

About the Author—CLAUDIO CUSANO is a post-doc student at DISCo, (Department of Information Science, Systems Theory, and Communication), of the University of Milano-Bicocca, where he took his Ph.D. in Computer Science. Since April 2001 he has been a fellow of the ITC Institute of the Italian National Research Council. The main topics of his current research concern 2D and 3D imaging, with a particular focus on image analysis and classification, and on face recognition.

About the Author—RAIMONDO SCHETTINI is an associate professor at DISCo, University of Milano Bicocca where he is in charge of the Imaging and Vision Lab. He has been associated with Italian National Research Council (CNR) since 1987. He has been team leader in several research projects and published more than 160 refereed papers on image processing, analysis and reproduction, and on image content-based indexing and retrieval. He is an associated editor of the Pattern Recognition Journal. He was a co-guest editor of three special issues about Internet Imaging (Journal of Electronic Imaging, 2002), Color Image Processing and Analysis (Pattern Recognition Letters, 2003), and Color for Image Indexing and Retrieval (Computer Vision and Image Understanding, 2004). He was General Co-Chairman of the 1st Workshop on Image and Video Content-based Retrieval (1998), of the First European Conference on Color in Graphics, Imaging and Vision (2002), and of the EI Internet Imaging Conferences (2000–2006).