Color Constancy Algorithm Selection Using CART

Simone Bianco, Gianluigi Ciocca, and Claudio Cusano

DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione),
Università degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy
{bianco,ciocca,cusano}@disco.unimib.it

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 selection is made by a decision forest composed by several trees that vote for one of the illuminant estimation algorithm. The most voted algorithm is then applied to the input image. Experimental results on the widely used Ciurea and Funt dataset demonstrate the accuracy of our approach in comparison to other algorithms in the state of the art.

1 Introduction

Computational color constancy aims to estimate the actual color in an acquired scene disregarding its illuminant. A scene can be modelled as a collection of Lambertian surfaces illuminated by a single, constant illuminant. 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 $\mathbf{RGB} = (R, G, B)$ 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 $\mathbf{C}(\lambda)$:

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

A. Trémeau, R. Schettini, and S. Tominaga (Eds.): CCIW 2009, LNCS 5646, pp. 31–40, 2009.

Since the only information available are the sensor responses $\boldsymbol{\rho}$ across the image, color constancy is an under-determined problem [1]; and thus further assumptions and/or knowledge are needed to solve it. Typically, some informati3on 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.

We investigated here if it is possible to automatically derive the suitability of a illuminant estimation algorithm for a given image by analyzing a set of visual features. To validate this hypothesis we developed a illuminant estimation framework and evaluated its performance on a public available dataset of images. Given a set of illuminant estimation algorithms, the 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 have been 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. Within this framework, a illuminant estimation strategy has been evaluated which selects for each image, a single algorithm on the basis of the responses of the trees.

Several computational color constancy algorithms exist in the literature which may be included in our framework, each based on different assumptions. Hordley [2] gives an excellent review of illuminant estimation algorithms. In this work, we chose five algorithms, but different algorithms can be used, or added to the set. Recently Van de Weijer et al. [3] have unified a variety of algorithms. These algorithms correspond to instantiations of the following equation:

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

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 I has unit length. 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 [4], 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 Equation 3.
2. White Point (WP) algorithm [5], 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 Equation 3.
3. Gray Edge (GE1) algorithm [3], 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 Equation 3.

4. Second Order Gray Edge (GE2) algorithm [3], 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 Equation 3.

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

2 Classification and Regression Trees for Algorithm Selection

To perform algorithm selection we used decision trees built according to the Classification and Regression Trees (CART) methodology [6]. Briefly, the classifiers are produced by recursively partitioning the feature space, each split being formed by conditions related to the features values. In tree terminology subsets are called nodes: the feature space is the root node, terminal subsets are terminal nodes, and so on. Once a tree has been built, a class is assigned to each of the terminal nodes, and when a new case is processed by the tree, its predicted class is the class associated with the terminal node into which the case finally moves on the basis of its features values. The construction process is based on training sets of cases of known class.

Tree classifiers provide a clear understanding of the conditions that drive the classification process. Moreover, they imply no distributional assumptions for the features. To improve generalization accuracy we decided to perform the classification by also using what is called a "perturbing and combining" method [7]. Methods of this kind, which generate in various ways multiple versions of a base classifier and use these to derive an aggregate classifier, have proved very successful in improving accuracy. We used bagging (bootstrap aggregating), since it is particularly effective when the classifiers are unstable, as trees are, that is, when small perturbations in the training sets, or in the construction process of the classifiers, may result in significant changes in the resulting prediction. With bagging the multiple versions of the base classifier are formed by making bootstrap replicates of the training set and using them as new training sets. The aggregation is made by majority vote. In any particular bootstrap replicate each element of the training set may appear repeated times, or not at all, since the replicates are obtained by resampling with replacement.

Our classifier is trained on a training set of images labeled with the corresponding best algorithm. 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: i) some algorithms generally perform better than others; ii) 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. For the second point, each pair of algorithms is considered and the average difference in performance obtained when one of the two algorithms corresponds to the best choice is computed. In other words, we computed the expected cost (i.e. degradation in performance) caused by the choice of an algorithm when another algorithm is the best choice. These costs are used during training to influence label assignment in such a way that the tree is optimized to minimize the expected misclassification cost instead of the number of errors.

3 Image Features

In the literature many features exist to be used in describing the image content [8,9,10,11]. For our problem, we have limited the choices of the features within the low level features category, since they do not require prior knowledge of the image content and are able to describe some aspects of the image in a compact and efficient way.

RGB color histogram is one of the most widely used image descriptors [12,13] 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. We quantized the RGB color space by uniformly dividing each color axis into 3 intervals for a total of 27 histogram bins.

Edge direction histogram can be used to determine the edge structures within an image and thus allow us to distinguish between different image classes. For example, strong edges can be found in buildings, roads, and other man-made structures.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. Edge direction histogram Edges are computed applying a Derivate of the Gaussian filter. The orientations are then thresholded and quantized into 18 bins each corresponding to angles of intervals of 10 degrees.

Wavelet statistics provide information at different levels of resolution about the textures and structures within the image. Wavelet multiresolution analysis is often used in content-based retrieval for similarity retrieval, target search, compression, texture analysis, biometrics, etc... [14,15,16]. For our purposes the wavelet statistics features are extracted from the luminance image using a three-iteration Daubechies wavelet decomposition, producing a total of ten bands. The energy i.e. the amount of information within each band, expressed in terms of the mean and variance of the absolute values in each band, provides a concise description of the image's content. This feature is thus composed by 20 components.

YCbCr Color Moments are used to describe the color distribution of an image. 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 [17]. We computed the first two central moments, mean, and standard deviation of each color channel of the YCbCr color space for a total of 9 values. 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 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. 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.

The percentage of clipped color components takes into account the extent of highly saturated color pixels i.e. pixels having the maximum value that can be represented on the device. We discriminate between pixels with zero, one, two or all three color components clipped (8 different cases). The values are accumulated in a histogram normalized with respect to the total number of pixels in the image, so that the histogram represents a probability density distribution.

The cast index is aimed at identifying the presence of a relevant cast within the image. This is important since a strong cast may be an indication that a particular illuminant is present. This feature is inspired by the work done in [18], where the cast is detected and classified into several classes according to its relevance. In this work, we do not consider the class of the cast but instead its distribution statistics (2 components) computed as in [18]. We modify the original formulation by changing the color space representation from the CIELAB to YCbCr which does not require the knowledge of the white point of the scene.

Edge Strengths is an important feature since many color illuminant estimation algorithms rely on statistics about the edges in the images. These estimations are reliable if computed on strong edges otherwise they are less accurate. We compute a histogram of edge magnitudes in order to capture the strength of the edges. The edges are detected as in the case of the edge direction histogram and the magnitudes are quantized into 5 intervals.

All the features have been chosen uniquely for their ability to describe the content of an image. 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.

4 Experimental Results

To evaluate our approach we measured its performance on a subset of the dataset of images presented by Ciurea and Funt [19] 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 two hours of videos. From each clip, a set of images was extracted, resulting in a dataset of more than 11000 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. For more details about the dataset extraction see [20]. The final dataset so extracted consisted of 1135 images. These have 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.

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.

4.1 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 [21], 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 ($\hat{\boldsymbol{\rho}_w}$):

$$e_{ANG} = \arccos\left(\frac{\boldsymbol{\rho}_w^T \hat{\boldsymbol{\rho}_w}}{\|\boldsymbol{\rho}_w\|\|\hat{\boldsymbol{\rho}_w}\|}\right). \tag{4}$$

Hordely and Finlayson [21] 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) [22].

4.2 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 [20] 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 [23,24]. 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 [25].

4.3 Training and Evaluation of the Classifier

We used cross validation to evaluate the performance of the classifier. The angular error of the illuminant estimation algorithms on the whole dataset is computed. This allows 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 Table 1 and in Table 2.

At this point, a ten-fold cross validation is used to train and to evaluate the algorithm selection strategy. Table 3 shows the confusion matrix obtained on the test set. Each row corresponds to an algorithm and reports the distribution of the output of the classifier estimated on the subset of the test set for which that algorithm is the best choice. Most of the images for which the DN

Table 1. 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 2. Matrix of the estimated misclassification costs estimated on the images of the training set

	Predicted Algorithm				
Best 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

Table 3. Confusion matrix of the classifier used for algorithm selection, estimated on the images of the test set

	Predicted Algorithm				
Best 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

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 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 classifier 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 classifier should not be evaluated in terms of classification accuracy, but in terms of the angular error of the selected algorithms. In fact, in more than 70% of test cases the loss of performance due to the choice of a suboptimal algorithm is below one degree of angular error with respect to the best algorithm.

The average angular error of our algorithm selection strategy is about 4.76 degrees, while the median angular error is about 3.21 degrees. These results are compared in Table 4 with those obtained by the five single algorithms and by three combining algorithms: AVG, which simply averages the results of the

Table 4. Summary of the results obtained on the test set by our algorithm selection strategy (AS), compared with the performance of the five simple algorithms. The best score for each column are reported in bold.

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
AVG	4.66	5.99	3
N2M	4.79	5.82	3
LMS	4.12	5.29	7
AS	**3.21**	**4.76**	**8**

estimations given by the five algorithms considered [26]; LMS which consists in a weighted average of the outputs of the individual algorithms [26]; N2M which averages the outputs of the three individual algorithms which gave the closest illuminant estimations, automatically excluding the two that gave the furthest estimations [27]. The performance of our approach is clearly superior to that of single and combined algorithms, at least on the dataset we considered.

5 Conclusions

In this work we have presented a framework for automatic illuminant estimation based on the selection of simple algorithms. To improve illuminant estimation accuracy, a decision forest is trained to identify the best algorithm within a set, for a given image. The choice of the best algorithm is based on a set of low-level features representing the pictorial content of the images.

Experimental results, performed on subset of uncorrelated images extracted from the widely used Funt and Ciurea dataset, demonstrate that our approach is able to improve the results compared with some state of the art algorithms. From our experiments the approach proposed reduced the median angular error by 22.1% with respect to the best illuminant estimation algorithm considered (LMS).

References

1. Funt, B., Barnard, K., Martin, L.: Is machine colour constancy good enough? In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 445–459. Springer, Heidelberg (1998)
2. Hordely, S.D.: Scene illuminant estimation: Past, present, and future. Color Research & Application 31(4), 303–314 (2006)
3. van de Weijer, J., Gevers, T., Gijsenij, A.: Edge-based Color Constancy'. IEEE Transactions on Image Processing 16(9), 2207–2214 (2007)
4. Buchsbaum, G.: A spatial processor model for object color perception. Journal of Franklin Institute 310, 1–26 (1980)
5. Cardei, V., Funt, B., Barndard, K.: White point estimation for uncalibrated images. In: Proc. IS&T/SID 7th Color Imaging Conference, pp. 97–100 (1999)
6. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks/Cole (1984)
7. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
8. Schettini, R., Ciocca, G., Zuffi, S.: Indexing and retrieval in color image databases. Color Imaging Science: Exploiting Digital Media, 183–211 (2002)
9. Antani, S., Kasturi, R., Jain, R.: Survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. Pattern recognition 35, 945–965 (2002)
10. Eakins, J.P.: Towards intelligent image retrieval. Pattern Recognition 35, 3–14 (2002)
11. Sikora, T.: The MPEG-7 visual standard for content description - An overview. IEEE Transaction on circuits and system for video technology 11(6), 696–702 (2001)

12. Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision 7(1), 11–32 (1991)
13. Gong, Y., Chuan, C.H., Xiaoyi, G.: Image indexing and retrieval using color histograms. Multimedia Tools and Applications 2, 133–156 (1996)
14. Idris, F., Panchanathan, S.: Storage and retrieval of compressed images using wavelet vector quantization. Journal of Visual Languages and Computing 8, 289–301 (1997)
15. Scheunders, P., Liven, S., Van de Wouwer, G., Vautrot, P., Van Dyck, D.: Wavelet-based texture analysis. International Journal Computer Science and Information management 1(2), 22–34 (1997)
16. Mojsilovic, A., Rackov, D., Popovic, M.: On the selection of an optimal wavelet basis for texture characterization. IEEE Transaction on Image Processing 9(12), 2043–2050 (2000)
17. Stricker, M.A., Orengo, M.: Similarity of color images. In: Proc. SPIE Storage and Retrieval for Image and Video Databases III Conference, pp. 381–392 (1995)
18. Gasparini, F., Schettini, R.: Color balancing of digital photos using simple image statistics. Pattern Recognition 37(6), 1201–1217 (2004)
19. Ciurea, F., Funt, B.: A Large Image Database for Color Constancy Research. In: Proc. IS&T/SID 11th Color Imaging Conference, pp. 160–164 (2003)
20. Bianco, S., Ciocca, G., Cusano, C., Schettini, R.: Improving Color Constancy Using Indoor-Outdoor Image Classification. IEEE Transactions on Image Processing 17(12), 2381–2392 (2008)
21. Hordley, S.D., Finlayson, G.D.: Re-evaluating Color Constancy Algorithms. In: Proc. 17th International Conference on Pattern Recognition, pp. 76–79 (2004)
22. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics 1, 80–83 (1945)
23. Lewis, R.M., Torczon, V.: Pattern search algorithms for bound constrained minimization. SIAM Journal on Optimization 9, 1082–1099 (1999)
24. Lewis, R.M., Torczon, V.: Pattern search methods for linearly constrained minimization. SIAM Journal on Optimization 10, 917–941 (2000)
25. Lewis, R.M., Torczon, V.: On the convergence of pattern search algorithms. SIAM Journal on Optimization 7, 1–25 (1997)
26. Cardei, V.C., Funt, B.: Committee-Based Colour Constancy. In: IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications, pp. 311–131 (1999)
27. Bianco, S., Gasparini, F., Schettini, R.: A Consensus Based Framework For Illuminant Chromaticity Estimation. Journal of Electronic Imaging 17(02), 023013 (2007)