Applying Visual Object Categorization and Memory Colors for Automatic Color Constancy

Esa Rahtu[1], Jarno Nikkanen[2], Juho Kannala[1],
Leena Lepistö[2], and Janne Heikkilä[1]

[1] Machine Vision Group
University of Oulu, Finland
[2] Nokia Corporation
Visiokatu 3, 33720 Tampere, Finland
esa.rahtu@ee.oulu.fi,jarno.nikkanen@nokia.com

Abstract. This paper presents a framework for using high-level visual information to enhance the performance of automatic color constancy algorithms. The approach is based on recognizing special visual object categories, called here as memory color categories, which have a relatively constant color (e.g. the sky). If such category is found from image, the initial white balance provided by a low-level color constancy algorithm can be adjusted so that the observed color of the category moves toward the desired color. The magnitude and direction of the adjustment is controlled by the learned characteristics of the particular category in the chromaticity space. The object categorization is performed using bag-of-features method and raw camera data with reduced preprocessing and resolution. The proposed approach is demonstrated in experiments involving the standard gray-world and the state-of-the-art gray-edge color constancy methods. In both cases the introduced approach improves the performance of the original methods.

Keywords: object categorization, category segmentation, memory color, color constancy, raw image.

1 Introduction

Color constancy is a characteristic feature of human visual system which causes the perceived color of objects to remain relatively constant under varying illumination conditions. It is also a desired property of digital cameras, which typically aim to reproduce the colors of the scene to look similar as they appeared to a human observer standing behind the camera when the image was taken. However, the response of digital camera sensors depends on the chromaticity of the illumination and this effect has to be compensated in order to achieve visually pleasing reproduction of colors. Therefore most cameras apply computational color constancy algorithms, also known as automatic white balancing algorithms, which estimate the illumination of the scene so that color distortions can be compensated [1,2].

P. Foggia, C. Sansone, and M. Vento (Eds.): ICIAP 2009, LNCS 5716, pp. 873–882, 2009.

The existing computational color constancy algorithms can be divided into two categories: the ones that require characterization of camera sensor response and the ones that do not. Examples of the former category are color by correlation [3] and gamut mapping algorithms [4], whereas the gray-world [5] or gray-edge [6] algorithms exemplify the other class. In large scale mass production of camera sensors, the color response of the sensors can vary from sample to sample. Having very strict limits for the color response would mean reduced yield and hence higher cost per sensor. Sample specific characterization is also possible, but that would have an impact on the sensor price as well. Consequently, the color constancy algorithms which do not rely heavily on accurate characterization information are useful in such cases in which the cost of the camera sensor is a very critical parameter. On the other hand, the accuracy of illumination estimates is typically better for the algorithms which utilize sensor characterization.

The common factor in the most of previous works on color constancy is that they are based on low-level image information. The use of object recognition in color constancy is considered in [7], but their approach requires that one or more of the exact training objects appear in the analyzed image. According to our knowledge the use of visual object categories is considered only in [8]. However the estimation method they present is based on purely utilizing the mean color values of the categories without any further analysis in the chromaticity domain. Moreover the evaluation method introduced there is rather expensive to compute.

The color constancy application that is considered in this paper is consumer photography with digital cameras, including mobile phone cameras. In this application visually pleasing color quality is more important than very precise color reproduction. Therefore, instead of sensor characterization, we investigate an approach which is based on analyzing the semantic content of images. That is, we aim to detect such object categories from the images which have memory colors associated with them. Such categories are, for example, foliage, grass, sky, sand and human skin [9]. Each of such objects have a limited range of chromaticities associated with them, referred to as memory color clusters hereinafter. Consequently, the initial estimate of white point, which can have error due to inaccurate characterization or poor algorithm performance, can be improved by modifying the white point in such way that the chromaticities of detected objects or surfaces fall closer to their corresponding memory color clusters.

In addition, many color constancy algorithms have difficulties in estimating the illumination chromaticity when there are only a few colors present in the image. This is the case for example for gamut mapping algorithms or gray-world and similar algorithms for obvious reasons. By utilizing the approach proposed in this paper it is possible to increase robustness also in these kinds of situations.

2 Memory Color Categories

The concept of memory color refers to such colors that are associated with familiar object categories in long term memory [10]. This concept is particularly useful in our application, where the goal is to provide visually pleasing colors and it is preferred to reproduce colors close to the corresponding memory colors. An essential characteristic of a memory color is the fact that it is defined in a relatively compact domain in the chromaticity space. In the following we describe how the memory colors used in this paper are learned from correctly white balanced sample images.

We collected a training dataset of 53 images illustrating the tested categories in different locations, time instants, and illuminations. For each training image we also associated a reference white point, which was based on illumination chromaticity measurements with Konica-Minolta CL-200 chroma meter. The reference points were used to white balance each training image according to the von Kries model [11,12]:

$$x_{wb} = Gx_{raw} = s \begin{bmatrix} \frac{1}{w_R} & 0 & 0 \\ 0 & \frac{1}{w_G} & 0 \\ 0 & 0 & \frac{1}{w_B} \end{bmatrix} x_{raw}, \tag{1}$$

where w_R, w_G and w_B are the corresponding RGB-coordinates of the reference white point, $s = \max(w_R, w_G, w_B)$, and $x_{raw} = [R_{raw}, G_{raw}, B_{raw}]^T$ and $x_{wb} = [R_{wb}, G_{wb}, B_{wb}]^T$ are RGB-vectors of the raw and the white balanced pixels respectively. The scaling s is introduced to prevent the colorization of the saturated areas of the image.

After the white balancing the images need to be further converted from the sensor color space to a sensor-independent reference color space, which in our case is the linear RGB space (RGB_{lin}) with $sRGB$ [13] primaries (i.e. transformation from RGB_{lin} to $sRGB$ is obtained by applying the gamma correction [12]). The conversion is done using 3×3 sensor specific conversion matrix C_C as $x_{lin} = C_C x_{wb}$, where $x_{lin} = [R_{lin}, G_{lin}, B_{lin}]^T$ is the vector of the resulting RGB_{lin} values.

The RGB_{lin} training images were roughly hand segmented by defining a set of bounding boxes that capture the memory color categories. The pixels in these segments were converted to chromaticity space $[R_{lin}/G_{lin}, B_{lin}/G_{lin}]^T$ [12], where a mean value was computed for each category in each image. The final memory color domain was defined by an ellipse $(x - m_{ell})^T C_{ell}^{-1} (x - m_{ell}) = r_m^2$, where m_{ell} is the weighted mean of the segmented pixels over all training images and C_{ell} is the corresponding covariance matrix. The size r_m of the ellipse remained as a parameter. The weighting used was the number of segmented pixels per image. Figure 1(a) illustrates some examples of the training images and segmentations.

3 Proposed Framework

In this section we describe the details of the proposed approach. We start from the recognition of the memory color categories and then continue by introducing a method for the refinement of the initial color constancy estimate. The overall process is illustrated in Figure 1(b).

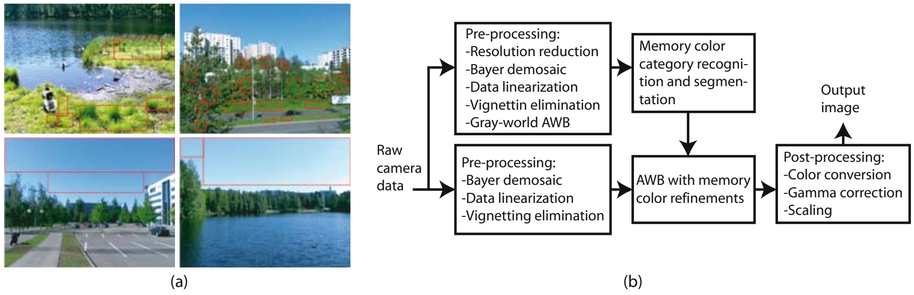

Fig. 1. (a) Examples of the training images and segmentations. (b) The overall image processing pipeline proposed in this paper (cf. [12]).

3.1 Category Recognition

The first step in the proposed process is to recognize the memory color categories. For this task we apply the widely used bag-of-words (BOW) method combined with a SVM classifier [14]. In BOW approach the image is described as a distribution over visual words, which are learned from the local visual descriptors of the training images using vector quantization. The local descriptors are computed from circular patches with radius 4, 8, and 12 pixels extracted on a regular grid with 10 pixel spacing. Each patch is described by one of the following three descriptors, gray scale SIFT [15], W-SIFT [16], or Centile [17], depending on the experiment. In the case of SIFT and W-SIFT the feature vectors were further reduced to 40 dimension using principal component analysis.

The vector quantization is performed by K-means clustering resulting in a vocabulary of 1000 words. In the SVM classifier we used Chi-squared kernel defined as $K(x,y) = e^{-\gamma\chi^2(x,y)}$, where γ is a learned parameter and $\chi^2(x,y) = \sum_j (x_j - y_j)^2/(x_j + y_j)$. The three different descriptors were chosen to examine the effects of different modalities in the recognition performance with our image data. SIFT, W-SIFT, represent two state-of-the-art texture based descriptor, where the first one applies only gray scale information and the second one includes also color modality. The Centile feature represents a simple method that exploits only color information.

Since we are interested in applying high-level color constancy estimation as an integral part of the camera's image processing pipeline (Fig. 1), we aim to make a fast recognition using the original raw data with clearly downscaled resolution. The data for the recognition may be achieved from the corresponding viewfinder image that is captured before the final image is taken. However due to the properties of raw camera data, some preprocessing is still essential.

The reduced pipeline we applied was the following: 1) remove the possible offset from the pixel raw values, 2) perform linear interpolation based Bayer pattern demosaicing, 3) downscale the image to 240 × 320, 4) perform gray-world white balancing. The normalization in step four was introduced in order to equalize the differences in color response between different models of camera sensors. The size 240×320 was selected to match the size of the viewfinder image in our camera system. One can refer to [12] for more comprehensive discussion and examples of camera systems.

3.2 Refining Color Constancy Approximation

The color constancy refinement takes place in the automatic white balancing stage of the processing pipeline illustrated in Figure 1(b). There we first make an initial estimation of the balance with some standard method, which in our experiments was taken to be either gray-world or gray-edge. These two reference algorithms were selected since they do not need any sensor characterization, which could be prohibitive in the case of low cost equipment. Furthermore the gray-edge algorithm [6] has been reported to achieve comparable results with the state of the art methods like gamut mapping [4] and color-by-correlation [3].

After the white balancing with the reference method, we take the category recognition result into account. If a memory color category is found, we perform fast approximate segmentation to the image. This is done by first converting the image pixels to RGB_{lin} and then to the chromaticity space $[R_{lin}/G_{lin}, B_{lin}/G_{lin}]^T$. In this space we take all pixels into the segment that lay inside the extended memory color ellipse. The ellipse is achieved from the corresponding memory color domain by extending the original ellipse size from r_m to r_s. For a satisfactory segmentation we must assume that chromaticities of the pixels are not too far from their true values. However we are only refining the result of the reference method, and we can assume that the solution is already reasonably close to the correct one. The experiments later illustrate that usually even the simple gray-world method produces an initial estimate that is close enough. Figure 2(a) illustrates results of the segmentation step.

Before refining the color constancy estimate, we verify that the memory color category covers more than given proportion p of the image area. The limitation is set in order to have enough support for the memory color for reliable refinement and to detect some of the missclassifications. If the support of the category is larger than the limit p, we compute the mean value m_{sRB} of the segmented pixels in $[R_{lin}/G_{lin}, B_{lin}/G_{lin}]^T$. The initial white balancing is then refined so that m_{sRB} moves to the closest point e_{sRB} at the corresponding memory color ellipse, if not inside the ellipse already. Given m_{sRB} and e_{sRB} the refined white balancing matrix is calculated as

$$p_1 = C_C^{-1} \left[e_{sRB}(1)\ 1\ e_{sRB}(2) \right]^T, \quad p_2 = C_C^{-1} \left[m_{sRB}(1)\ 1\ m_{sRB}(2) \right]^T, \tag{2}$$

$$G_{ref} = s \begin{bmatrix} p_1(1)/p_2(1) & 0 & 0 \\ 0 & p_1(2)/p_2(2) & 0 \\ 0 & 0 & p_1(3)/p_2(3) \end{bmatrix} G_{init}, \tag{3}$$

where $x(i)$ refers to i-th component of a vector x, G_{init} and G_{ref} are the initial and refined white balance matrices respectively, and s is such constant that the minimum value at the diagonal of G_{ref} is equal to one. From here the processing pipeline continues normally using now the estimated G_{ref} instead of G_{init} as a white balancing transformation.

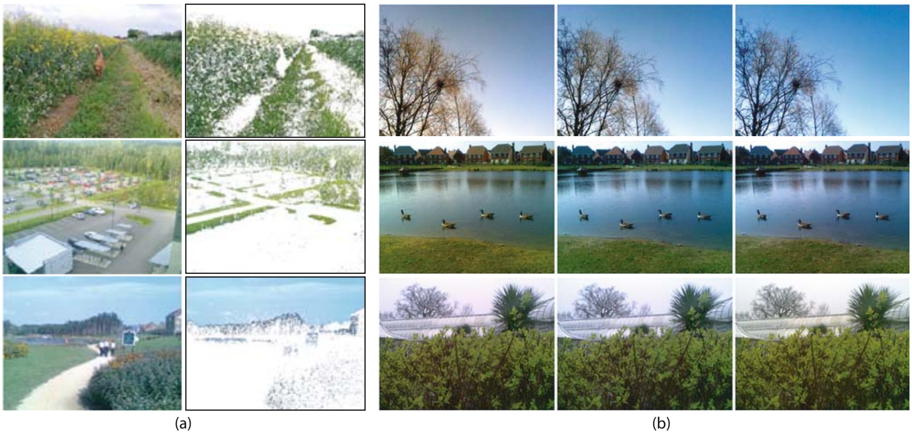

Fig. 2. (a) Example results of the category segmentation. The pixels not included into segment are shown as white. (b) Samples of the final white balanced images in *sRGB*. The columns from left to right illustrate the white point estimation using gray-edge, refined gray-edge, and ground truth, respectively.

4 Experiments

To demonstrate the performance of our framework, we performed two kind of experiments. First we evaluated the memory color categorization and then the method for color constancy refinement. We begin with the categorization experiments.

4.1 Memory Color Categories

In these experiments we evaluate the method using two memory color categories, namely grass & foliage and sky. We collected seven datasets of raw images, each taken by different person with different sensor in a wide variety of time instants and places. The number of images in these sets were 377, 518, 508, 506, 319, 108, and 508. The images in the training sets were processed using the pipeline in Section 3.1 and hand labeled so that if the image contains significant portion of the memory color category it was tagged with the category label and otherwise not. This differs slightly from the traditional categorization, but since our goal at the end was to refine the color constancy estimation we were only interested in images, where the support for the category was large enough. Figures 2 and 3 illustrate some images used in the experiment.

The categorization system described in Section 3.1 was trained using six of the image sets, and then tested using the seventh one. For each of the descriptors, SIFT, W-SIFT, and Centile, we calculated the mean performance over the all seven combinations of test and training sets. The resulting classification performances are listed in Table 1. Each column gives the result for a train-test-split where the given set is used for testing.

Table 1. Mean classification performances for grass & foliage and sky categories. Each column gives the results for a train-test-split where given set is used as a test set.

Grass & foliage category:								
SIFT descriptor	set 1	set 2	set 3	set 4	set 5	set 6	set 7	**average**
True positive	74.3 %	83.7 %	85.9 %	88.0 %	74.2 %	66.7 %	92.6 %	**86.2 %**
False positive	8.8 %	4.8 %	2.7 %	15.7 %	7.5 %	13.1 %	12.6 %	**7.7 %**
W-SIFT descriptor								
True positive	82.9 %	80.6 %	93.7 %	96.4 %	81.8 %	54.2 %	96.3 %	**91.5 %**
False positive	3.3 %	2.6 %	1.1 %	15.2 %	5.9 %	8.3 %	15.1 %	**5.7 %**
Centile descriptor								
True positive	69.5 %	81.6 %	88.0 %	91.3 %	74.2 %	58.3 %	75.6 %	**81.2 %**
False positive	4.0 %	5.5 %	5.2 %	12.7 %	15.4 %	3.6 %	5.7 %	**7.4 %**

Sky category:								
SIFT descriptor	set 1	set 2	set 3	set 4	set 5	set 6	set 7	**average**
True positive	68.2 %	59.2 %	76.5 %	66.4 %	63.4 %	77.8 %	87.6 %	**73.7 %**
False positive	5.7 %	3.2 %	4.5 %	3.2 %	3.0 %	9.7 %	11.6 %	**5.1 %**
W-SIFT descriptor								
True positive	78.8 %	75.0 %	67.9 %	77.3 %	68.3 %	94.4 %	93.6 %	**81.1 %**
False positive	4.9 %	2.5 %	2.3 %	1.9 %	2.5 %	4.2 %	12.7 %	**4.1 %**
Centile descriptor								
True positive	50.8 %	68.4 %	71.6 %	64.8 %	61.0 %	80.6 %	91.4 %	**71.9 %**
False positive	3.8 %	2.5 %	4.5 %	4.2 %	3.8 %	5.6 %	8.4 %	**4.3 %**

The overall performance with the sky category seems to be lower than with grass & foliage category. This probably follows from the characteristic texture of the latter category compared to almost textureless sky. Furthermore we can observe that texture based features are performing better, but still computationally simple Centile features result relatively high recognition rate especially with grass & foliage category. In some cases one can also observe near 10% false positive rate. A closer look reveals that almost all of these images contain a little portion of the memory color category, but not enough to be labeled as positive in the ground truth. These images however rarely cause problems in the color constancy refinement, because of the limit in the segment size.

4.2 White Balance Refinements

For the fifth image set in the categorization experiment we also measured the illumination chromaticity with similar methods as in Section 2. These values were

Table 2. Relative improvements achieved. The left result refers to gray-world and right to gray-edge method as initial approximation.

Category	Δerr_{mean}	Δerr_{median}	number of images improved
grass & foliage	2.6 % / 5.6 %	14.3 % / 2.7 %	51.9 % / 61.4 %
sky	25.2 % / 1.9 %	29.1 % / 15.0 %	76.9 % / 57.5 %

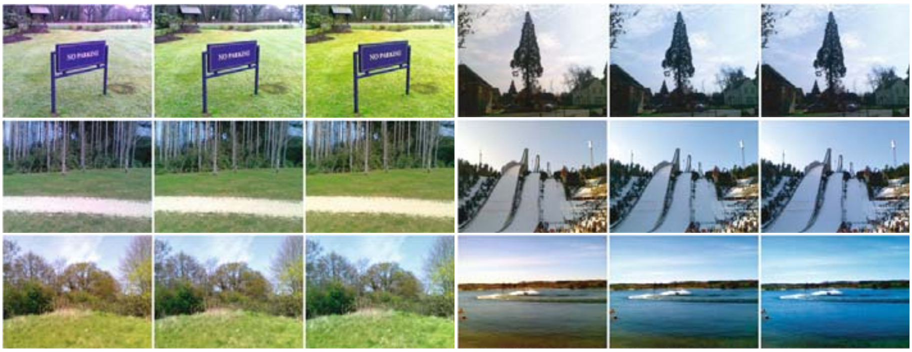

Fig. 3. Samples of the final white balanced images in *sRGB*. The columns from left to right illustrate the white point estimation using gray-world, refined gray-world, and ground truth, respectively.

used as a ground truth in the following automatic white balancing experiment, where the initial color constancy, provided by a low-level reference method, was refined for those images, which were recognized to contain a memory color category. As reference methods we used both gray-world and gray-edge algorithms. The category labeling for the test set was taken from the results achieved with SIFT descriptor in the previous section. We selected SIFT instead of W-SIFT for this experiment since it was faster to evaluate. The framework used in the experiment was the one described in Section 3.2 with parameter values $r_m = 0.5$, $r_s = 3.0$, and $p = 10\%$, for the grass & foliage category and $r_m = 0.6$, $r_s = 2.0$, and $p = 25\%$, for the sky category. For the gray-edge method we applied parameter values $n = 1$, $p = 1$, and $\sigma = 6$, according to [6].

As an error measure we calculated angle difference of the white point coordinates $err = \cos^{-1}(\hat{w}_{true} \cdot \hat{w}_{estim})$, where $\hat{a} = a/||a||_{L^2}$, and w_{true} and w_{estim} are vectors containing the ground truth and the estimated coordinates of the white point respectively. The results are shown in Table 2. The relative improvements reported there are calculated as follows $\Delta err_{mean} = (mean(err_{init}) - mean(err_{ref}))/mean(err_{init}) \cdot 100\%$, where err_{init} and err_{ref} refer to errors of the initial and refined approximations respectively, and *mean* is the mean over all positively classified images. The median error Δerr_{median} is achieved by replacing *mean* with median operator. Finally the number of images improved indicates the portion of the positively classified images, that resulted in the same or better estimation than with the reference method. Some images of the results are also illustrated in Figures 2(b) and 3.

It can be observed, that according to all measures, the application of memory color correction achieves a considerable improvement in both categories, and especially in the case of sky. This is probably due to the fact that the memory color domain for sky is more compact than that of grass & foliage. Further improvements may be achieved by dividing the grass & foliage category in several sub classes for which more compact memory color clusters are available. Finally also visual results indicate a clear improvement in the subjective quality of the white balancing.

5 Conclusions

In this paper we presented a framework for applying visual category recognition results to improve automatic color constancy. The approach was based on so called memory color categories, which are known to occupy a compact region in the chromaticity space. The category recognition was performed by using the bag-of-features approach for low resolution input images which were first roughly white balanced with the simple and fast gray-world algorithm. Then, the categorization was used for adjusting the white balance produced by a low-level method, such as the gray-world or gray-edge algorithms. The experiments indicate that the proposed approach constantly improves the result of both low-level methods. Hence, utilizing semantic information of object categories is a promising new direction for automatic color constancy.

Acknowledgments

The Academy of Finland is gratefully acknowledged for providing the funding for this research.

References

1. Barnard, K., Martin, L., Coath, A., Funt, B.: A comparison of computational color constancy Algorithms. II. Experiments with image data IEEE Transactions on Image Processing 11(9), 985–996 (2002)
2. Barnard, K., Cardei, V., Funt, B.: A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data. IEEE Transactions on Image Processing 11(9), 972–984 (2002)
3. Finlayson, G., Hordley, S., Hubel, P.: Color by correlation: a simple, unifying framework for color constancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), 1209–1221 (2001)
4. Forsyth, D.: A novel algorithm for color constancy. International Journal of Computer Vision 5(1), 5–36 (1990)
5. Buchsbaum, G.: A spatial processor model for object color perception. J. Frank. Inst. 310 (1980)
6. van de Weijer, J., Gevers, T., Gijsenij, A.: Gray edge Edge-based color constancy. IEEE Transactions on Image Processing 16(9), 2207–2214 (2007)
7. Obdrzalek, Š., Matas, J., Chum, O.: On the Interaction between Object Recognition and Colour Constancy. In: Proc. International Workshop on Color and Photometric Methods in Computer Vision (2003)
8. van de Weijer, J., Schmid, C., Verbeek, J.: Using high-level visual information for color constancy. In: Proc. International Conference on Computer Vision, pp. 1–8 (2007)

9. Fairchild, M.: Colour appearance models, 2nd edn. John Wiley & Sons, Chichester (2006)
10. Bodrogi, P., Tarczali, T.: Colour memory for various sky, skin, and plant colours: effect of the image context. Color Research and Application 26(4), 278–289 (2001)
11. Barnard, K.: Practical color constancy. PhD Dissertation, School of Computing Science, Simon Fraser Univ., Bumaby, BC, Canada (1999)
12. Nikkanen, J., Gerasimow, T., Lingjia, K.: Subjective effects of white-balancing errors in digital photography. Optical Engineering 47(11) (2008)
13. Stokes, M., Anderson, S., Chandrasekar, S., Motta, R.: A standard default color space for the internet-sRGB (1996), http://www.w3.org/Graphics/Color/sRGB
14. Csurka, G., Dance, C., Fan, L., Williamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Proc. European conference on Computer Vision, pp. 59–74 (2004)
15. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
16. van de Sande, K., Gevers, T., Snoek, C.: Evaluation of color descriptors for object and scene recognition. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (2008)
17. Silvén, O., Kauppinen, H.: Color vision based methodology for grading lumber. In: Proc. 12th International Conference on Pattern Recognition, vol. 1, pp. 787–790 (1994)