Evolutionary Color Constancy Algorithm Based on the Gamut Mapping Paradigm

Cristian Munteanu[1,2], Agostinho Rosa[1], Manuel Galan[2],
and Enrique Rubio Royo[2]

[1] Instituto de Sistemas e Robotica, Instituto Superior Tecnico, Av. Rovisco Pais 1,
Torre Norte 6.21, 1049-001, Lisboa, Portugal
`{cmunteanu, acrosa}@isr.ist.utl.pt`
`http://laseeb.isr.ist.utl.pt/`
[2] CICEI, Universidad de Las Palmas de Gran Canaria, Edificio Central del Parque,
Científico y Tecnológico, Campus Universitario de Tafira, 35017 – Las Palmas, Spain
`rubio@cicei.com`

Abstract. In recent years, extensive work has been done to design algorithms that strive to mimic the robust human vision system which is able to perceive the true colors and discount the illuminant from a scene viewed under light having different spectral compositions (the feature is called "color constancy"). We propose a straightforward approach to the color constancy problem by employing an Interactive Genetic Algorithm [1] (e.g. a Genetic Algorithm [2], [3] guided by the user) that optimizes a well known and robust variant of color constancy algorithm called "gamut mapping" [4]. Results obtained on a set of test images and comparison to various color constancy algorithms, show that our method achieves a good color constancy behavior with no additional knowledge required besides the image that is to be color-corrected, and with minimal assumptions about the scene captured in the image.

1 Introduction

Color Constancy algorithms are fundamental both in Computer Vision whenever recognition of objects in a scene is performed based on the color of the respective objects, as well as for color correction of digital pictures with applications in medicine, remote sensing, arts and media. When compared to the human visual system, algorithms that perform Color Constancy fail short to achieve the same effectiveness in recuperating the colors in scenes taken under different illuminants. This is due to the fact that such algorithms make restrictive assumptions about the world from which the image was taken. More-over, several algorithms require additional information about the scene and the technical characteristics of the camera that captures the image, knowledge that is not always readily available. We tackle the color constancy problem by employing an Interactive Genetic Algorithm [1] (e.g. a Genetic Algorithm guided by the user) that optimizes a well known and robust variant of color constancy algorithm called "gamut mapping" [2]. Results are given on a set of test images and comparison is made to various color constancy algorithms that proved efficient and that pertain to the following categories: Gray World (GW) methods, Gamut Mapping methods, and Neural Networks methods [2].

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 404 – 409, 2005.

1.1 Gamut Mapping as Basis Transformation for IGA-GM

The set of all possible responses due to known or expected surface reflectances, as seen under a known, canonical illuminant (usually taken to be the daylight, or an approximation of it), is a convex set, referred to as the canonical gamut. Similarly, the set of responses due to an unknown illuminant is also a convex set. Illumination change is assumed to be modeled by a simple diagonal transformation according to the research of von Kries which was later confirmed to hold as a reasonable assumption in [3]. It was shown that under the von Kries adaptation assumption the two gamuts are within a diagonal transformation of each other. Due to the fact that GM proposes a simple diagonal transformation to each pixel in the image, and because it has been shown that under not very restrictive assumptions such diagonal transformations suffice for good color constancy [3], in this paper we adopt the GM method as the basis for the color constancy transformation applied to correct a given image. The main drawbacks of the classical GM methods are: a) they use extensive knowledge about the camera used to take the picture; b) they use a priori information about the world in which the image was taken in order to construct the gamuts.

2 IGA-GM. Algorithm Presentation

In the present paper we employ an Interactive Genetic Algorithm (IGA) in which a human evaluator iteratively gives a subjective score for each result of a diagonal mapping (corrected image), until the best corrected image (and implicitly the best diagonal mapping) is discovered. Thus IGA-GM adapts to the "best" gamut mapping (diagonal transformation) according to human subjective criteria. This is done without making use of any a priori knowledge about the surrounding world conditions, or the camera. The Genetic Algorithm [4] is given in pseudocode in Fig.1. Each individual x_i in the population (chromosome) codes a diagonal transformation which is next applied to each pixel in the original (input) image, to give the corrected (output) image:

$$\forall i, j : (R^{\text{corrected}}_{\text{pixel}(i,j)}, G^{\text{corrected}}_{\text{pixel}(i,j)}, B^{\text{corrected}}_{\text{pixel}(i,j)}) = (x_{i,1} \cdot R^{\text{original}}_{\text{pixel}(i,j)}, x_{i,2} \cdot G^{\text{original}}_{\text{pixel}(i,j)}, x_{i,3} \cdot B^{\text{original}}_{\text{pixel}(i,j)}) \tag{1}$$

How well a chromosome performs (that is, how good the diagonal mapping is in recuperating the original colors and how well it corrects the input image) is judged mainly by the human evaluator who looks at the corrected image and gives a numeric score, called fitness value: $f(x_i)$ ranging from 0 (worst appearing image) to 10 (best appearing image). The user shouldn't evaluate all images corrected by each chromosome in the population, in each generation of the algorithm, because such a process would become too tedious (there would be a lot of images to evaluate). Many of the chromosomes to be evaluated are actually given a fitness value automatically, using a clustering algorithm over fitness values previously allocated by the user. Clustering is performed by two distinct procedures (see Fig. 1): `evaluate_1` and `evaluate_2`. `evaluate_1` is applied in the first generation of the GA, when the population is initialized randomly. The population is clustered and for each cluster an individual is picked up at random. The respective chromosome is evaluated by the user. The rest of the members of the respective cluster get the same fitness value as the one allocated by the user. In `evaluate_2` again the population is first clustered, then for each individual in the population we identify the cluster in which this individual lies. There is a fixed probability P_E that this individual will be evaluated by the user. If not evaluated by the user we proceed as follows: if the size of the respective cluster is greater than 2, the fitness of the respective individual is computed as the average of the fitness of all individuals in the cluster. If the cluster contains just one individual (e.g. the individual for which we calculate the fitness), then the respective chromosome is evaluated by the user.

```
t : = 0
initialise P(t)
evaluate_1 P(t)
while (terminate(P(t)) ≠ true) do
{
        P'(t) := select_tournament (P(t) | q)
        P''(t):= crossover (P'(t) | P_c)
        P'''(t):= mutate (P''(t) | P_m)
        P(t+1):= elitist (P'''(t) ∪ P(t) | K)
        evaluate_2 P(t+1)
        t := t+1
}
endwhile
```

Fig. 1. Pseudocode of IGA-GM. t represents the time/generation index; $P(t)$ is the population at generation t, $P'(t)$ is the population at generation t after selection, $P''(t)$ is the population at generation t after selection and crossover, and $P'''(t)$ is the population at generation t after selection, crossover and mutation. K is the number of elites.

The selection strategy has been adopted to insure a steady convergent behavior of the algorithm. The trade-off we had to make is the well-known trade-off between exploration and exploitation present in any search method including EA. The convergent exploitation assured by selection and crossover should well-balance the wide exploration effect achieved by our mutation operator [5]. The selection method (`select_tournament` in Fig. 1) was chosen as a combination between binary tournament for which groups of $q = 2$ individuals are randomly formed and the chromosome with better fitness in the group is selected into the next generation [5], and a K-elitist scheme that attempts the preservation of the K best individuals in the population [6]. Binary tournament was chosen because it has a constant and relatively high selection pressure [7], and it is simple to implement and computationally light. The elitist scheme does not assure with probability 1 the preservation of the best individuals in the population, it just attempts this preservation. This is due to the subjective nature of fitness allocation by the user: the human evaluator might score the same image (corresponding to the best chromosome) with different fitness values on different occasions, and when this score diminishes in value the respective chromosome (i.e. the best solution) might be lost from the population due to selection effects.

The crossover operator should efficiently exploit the search space around the parents (i.e. the pair of chromosomes that undergo crossover). The advantage of using SBX is that it generates children (i.e. the pair of chromosomes resulted after crossing-over the parents) that are spread symmetrically around the parents and proportionally to the spread of the parents, as discussed in [8]. SBX was implemented as described in [8], the parameter that controls the spread of the children around the parents being η, and the probability of applying crossover is P_c.

The mutation operator should counter-balance the reduction of diversity due mainly to selection (associated with exploitation of the search space) and reintroduce diversity into population and thus increase exploration of the search space. The reduction of diversity is a cause of premature convergence [5] and it becomes more apparent when small populations of short in length chromosomes are used. This is precisely the case of our application. For such cases we have designed a novel mutation operator based on Principal Component Analysis, called PCA-mutation in [9] which maintains high levels of diversity in the population and increases the probability of discovering better solutions, as shown in [9]. The parameter of PCA-mutation (see [9]) is denominated c, and the probability of applying mutation is P_m.

3 Experimental Results

The experimental part comprises two main sections: a) firstly we perform a comparison between IGA-GM and other Color Constancy algorithms for a single unknown illuminant (e.g. different than canonical or daylight illuminant). b) secondly, we check the effectiveness of IGA-GM on a set of scenes taken under two quite different than daylight illuminants. In Table 1 we give the parameters of IGA-GM used in both experimental parts and for all images in the test set.

Table 1. IGA-GM parameters: N - population size, l- number of genes in each chromosome that undergo the evolution process, Pc - crossover probability , Pm - mutation probability, P_E - user evaluation probability, q – size of the tournament selection, c – parameter of PCA-mutation (see [35]), η – parameter of the SBX crossover, T_{max} -number of generations the GA is allowed to run, vlb_j - lower bound of the genes, vub_j - upper bound of the genes, χ - inconsistency threshold for the clustering algorithm

N	l	P_c	P_m	P_E	K	q	c	η	T_{max}	vlb_j	vub_j	χ
50	3	0.9	0.25	0.04	1	2	100	2	10	0	100	0.95

Fig. 2. Comparison to Color Constancy algorithms (Detergent image). Upper-row from left to right: original image (to be color-corrected), target image (image under canonical illuminant), GW, Retinex; Lower-row from left to right: NN, GM, IGA-GM.

Table 2. RMS error calculated for *Detergent* image on pairs between target image (T) and Color Constancy corrected image. The image that is not corrected (the original - input image) is denoted as O.

	(T, O)	(T,GW)	(T, Retinex)	(T,NN)	(T, GM)	(T, IGA-GM)
RMS$_{(R,G,B)}$	2.34e+04	18.02	49.13	13.38	22,30	16.16
RMS$_{(r,g)}$	16.42	12.88	16.27	7.78	21.74	7.99

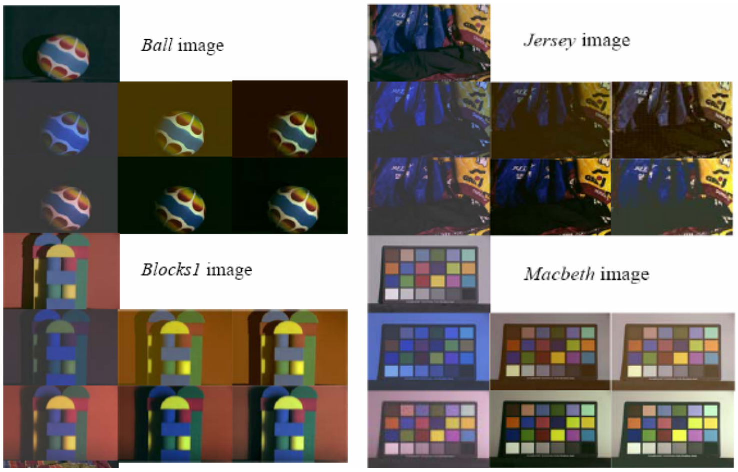

Fig. 3. IGA-GM solutions on different illuminants. Figure is divided in horizontal bands for each scene. First column first line: target image; second line first column: image under “solux_4700+3202”; second line second column: IGA-GM; second line third column: “IGA-GM + intensity correction”; third line first column: image under “syl_wwf”; third line second column: IGA-GM; third line third column: IGA-GM + intensity correction.

4 Conclusions

The main advantages of IGA-GM are the following: the method does not require previous information and knowledge about the scene being captured in the image, it does not make further restrictive assumptions about the world from which the image was taken, it is a simple algorithm with no pre-processing or training phases, it takes into account very subtle subjective criteria when judging the quality of an image, criteria that so far cannot be successfully "coded" in an automatic or machine controlled way. Moreover, results obtained on a wide set of test images show that IGA-GM achieves an effectiveness that is close to that of the best Color Constancy methods operating on the respective images. Consequently, the advantage lies in the robustness of IGA-GM, that is: we may use a single method (i.e. IGA-GM) to correct a wide range of images, instead of testing several Color Constancy methods (such as GM, GW, NN, etc.) and see which performs better on the respective images. Though results are good even when the human evaluator that analysis the outputs of IGA-GM doesn't have any knowledge about the scene captured in the input image, an increase in efficiency is expected when human evaluator experts are used to correct images which pertain to fields for which they have acquired the necessary expertise. Thus, for future work, we will first divide the images into groups pertaining to various fields and letting a human expert on the respective field correct the images using IGA-GM. Such fields of application may include medical images, remote sensing images, consumer and commercial images, or artistic photography.

References

1. Takagi, H.: Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. In: Proceedings of the IEEE 89 (2001) 1275–1296
2. Barnard, K., Cardei, V., Funt, B. V.: A Comparison of Computational Color Constancy Algorithms-Part I: Methodology and Experiments with Synthesized Data. IEEE Trans. on Image Processing 11(9) (2002) 972–984
3. Finlayson, G., Drew, M. S., Funt, B.V.: Diagonal transforms suffice for color constancy. In: Proceedings IEEE Int. Conf. on Computer Vision (1993) 164–171
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
5. Back, T., Fogel, D., Michalewicz, Z., Bäck, T.(eds.): Handbook of Evolutionary Computation. Institute of Physics Publishing (1997)
6. Bäck, T., Hoffmeister, F.: Extended selection mechanisms in Genetic Algorithms. In: Proceedings 4th Int. Conf. Genetic Algorithms (1991) 92–99
7. Miller, B. L., Goldberg, D. E.: Genetic Algorithms, tournament selection and the effects of noise. Complex Systems 9 (1996) 193–212
8. Deb, K., Beyer, H.-G.: Self-Adaptation in Real-Parameter Genetic Algorithms with Simulated Binary Crossover. Evolutionary Computation 9(2) (2001) 197–221
9. Munteanu, C., Lazarescu, V.: Improving mutation capabilities in a real-coded GA. Proceedings of EvoIASP'99. Lecture Notes in Computer Science, Vol. 1596. Springer-Verlag, Berlin Heidelberg New York (1999) 138–149