On pre-filtering with Retinex in color image retrieval

Gianluigi Ciocca*, Daniele Marini[+], Alessandro Rizzi[§], Raimondo Schettini*, Silvia Zuffi*[1]

*Istituto Tecnologie Informatiche Multimediali
Consiglio Nazionale delle Ricerche Milano, Italy

[+]Dip. Scienze dell'Informazione
Università di Milano, Italy

[§]Dip. Elettronica per l'Automazione
Università di Brescia, Italy

ABSTRACT

We have examined the performance of various color-based retrieval strategies when coupled with a pre-filtering Retinex algorithm to see whether, and to what degree, Retinex improved the effectiveness of the retrieval, regardless of the strategy adopted. The retrieval strategies implemented included color and spatial-chromatic histogram matching, color coherence vector matching, and the weighted sum of the absolute differences between the first three moments of each color channel. The experimental results are reported and discussed.

Keywords: image retrieval, image indexing, Retinex, image color features

1. INTRODUCTION

The use of color as an object descriptor has been introduced in the field of object recognition and image retrieval with the aim of overcoming the limitation of the dependence on the positioning of object and sensor, that shape-based methods present[1]. While the color distribution of the object could be considered a sort of invariant with respect to the camera's viewing point and to the position, orientation, and partial occlusion of the object, how the object appears to a viewer and how it is recorded by a sensor is strongly dependent on the illumination. A change in the illumination of the scene produces a modification in the RGB colors of the corresponding image that may determine the failure of the recognition strategies. While in many cases the user may still be able to recognize the colors in the scene, we can only guess to what extent an image search engine can perform the same task, and at what cost when retrieval is based purely on color. This problem arises when we can not assume constant imaging conditions during data acquisition, as in the case of a heterogeneous image database (IDB), the images of which are collected from many different sources. A further source of color shifts may also come from the use of different device color spaces.

Previous works in the field of object recognition and image retrieval have drawn attention to how changes in illumination conditions can affect results, and proposed the application of color constancy algorithms, or color invariant indexing to cope with this problem[1, 2, 3, 4, 5].

We examine here the performance of various color-based retrieval strategies when coupled with a pre-filtering Retinex algorithm to see whether, and to what degree, the effectiveness of the retrieval is improved, regardless of the strategy adopted.

2. THE RETINEX MODEL

Retinex was originally proposed by Land and McCann[6] in order to understand and emulate the color perception of the human vision system. Many variants of the Retinex method of computing lightness have been presented in the last thirty years. These methods model visual perception by exploiting the spatial distribution of the colors in the scene. Since the present

[1] Email: ciocca@itim.mi.cnr.it, marini@dsi.unimi.it, rizzi@ing.unibs.it, centaura@itim.mi.cnr.it, zuffi@itim.mi.cnr.it

Internet Imaging II, Giordano B. Beretta, Raimondo Schettini, Editors,
Proceedings of SPIE Vol. 4311 (2001) © 2001 SPIE · 0277-786X/01/$15.00

investigation intends only to verify whether the preprocessing of images using Retinex can somehow make color-based image retrieval more stable, on the basis of previous studies and comparison of some of the existing Retinex methods[7], we have decided to implement the algorithm described briefly here below.

We let I be an image to be processed; for each pixel I^x at position *x*, a number *N* of random paths starting from different points I^{j_i} with *i=1..N* and ending at pixel I^x are generated (see Figure 1). The type of random paths differs depending upon the implementation of the Retinex algorithm; here we have chosen a Brownian-paths generator[7] approximated with a mid-point displacement technique[8].

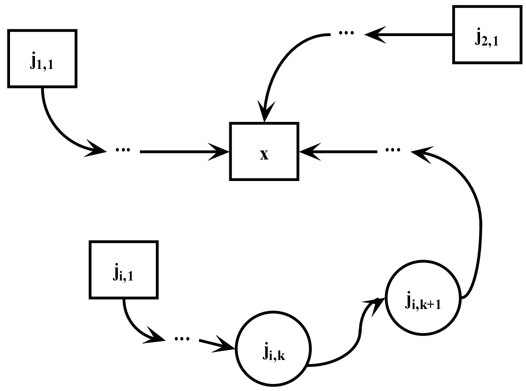


Fig. 1 Random paths to pixel I^x.

For each chromatic channel *c*, let us indicate the i-th path as a sequence of *n* pixels $\left\{I_c^{j_{i,1}}, I_c^{j_{i,2}}, \ldots, I_c^{j_{i,n}}\right\}$ with $I_c^{j_{i,n}} = I_c^x$ and $I_c^{j_{i,1}} = I_c^{j_i}$. Along each path, a sequence of ratio products called *chain value* (*CV*) is computed according to the following rules:

$$CV_c^{i,1} = 1$$

$$\text{for } k = 2 \text{ to } n \qquad CV_c^{i,k} = CV_c^{i,k-1}\delta_c^{i,k} \qquad \text{where } \delta_c^{i,k} = \begin{cases} \dfrac{I_c^{j_{i,k}}}{I_c^{j_{i,k-1}}} & \text{if} \quad \left|I_c^{j_{i,k}} - I_c^{j_{i,k-1}}\right| > threshold \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Along the path, if all the previous pixels are darker than the first one, the value of *CV* is lower than the unit. If a lighter pixel is found, the *CV* value overtakes the unit and the following *reset mechanism* is applied:

$$\text{If } CV_c^{i,k} > 1 \text{ then } CV_c^{i,k} = 1 \tag{2}$$

In this way the *CV* computation starts again from that pixel.
The output pixel value $\bar{I}_c^x$ for the channel *c* of the original pixel I^x is the mean of all the *CVs,* computed along all the *N* random paths ending at I^x:

$$\bar{I}_c^x = \frac{1}{N}\sum_{i=1}^{N} CV_c^{i,x} \tag{3}$$

These computations must be executed independently on the three RGB channels considered as an approximation of the *l*, *m*, and *s* retinal wavebands[6].

The threshold value in Equation 1 allows us to ignore the smooth luminance shading and non-uniform illumination. It has been found that this threshold value should be in the [0,10] percent range of the channel color depth[9]; we have set it at 5 percent.

The implementation of the Retinex algorithm described so far has been expedited with the use of Look Up Tables (LUT)[7]. The algorithm is applied to a small sub-sampled version of the original image; three LUT remapping functions are then constructed, one for each channel, between the sub-sampled and its filtered image. These LUT functions are then applied on the original full size image to obtain the final output (see Figure 2).

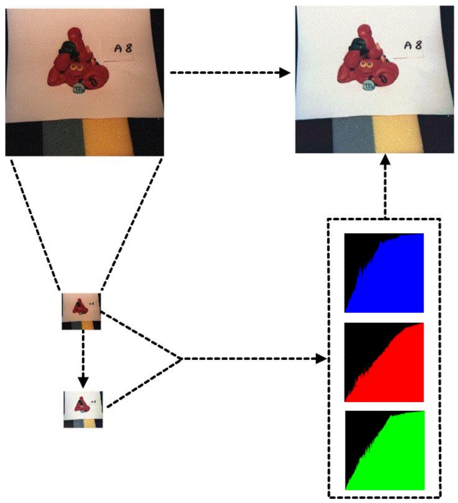


Fig. 2 The LUT Retinex algorithm

The local effects of the Retinex filter may transform pixels with the same value into different values: in other words, $I_c^x = I_c^y$ does not imply $\bar{I}_c^x = \bar{I}_c^y$. Therefore in building the LUT, each value from the input image will be mapped into the average of the corresponding output values. For values not present in the sub-sampled image, a linear interpolation between the two nearest neighboring output values is used.

3. EXPERIMENTS

The experiments were performed on two databases, containing 310 paintings and 387 ceramic objects respectively. We have randomly selected 15 images from each database, and for each of these simulated a changes in the imaging conditions using one of 8 illuminants. These images have then been used to query the corresponding database, using one strategy at time, first without any pre-filtering, and then applying the Retinex algorithm to both the query and the database images (see Figure 3). The experiments have been performed using the Quicklook[2] Image Search Engine[10].

3.1. Simulating the change in illumination

If we consider two images representing the same scene under two different illuminants, a simple model for describing the changes between corresponding pixels is a linear transformation (a 3x3 matrix of coefficients) regardless of the pixel's location. An even simpler model is one considering a diagonal matrix. The diagonal model has been proposed by Von Kries as a model for human adaptation[11]. For it to hold exactly, sensor filters must be assumed to have narrow band properties. Although this hypothesis has not been completely verified in practice, it is frequently assumed: for example, in the Retinex theory, and in the definition of illuminant-invariant image descriptors[2, 3, 6].
To simulate a change in illuminant conditions we have used Von Kries' chromatic adaptation model. The images were coded in Standard RGB color space. sRGB values were first converted to CIE XYZ coordinates, and the Von Kries transform was then applied to map tristimulus values from the sRGB reference white to the target illuminant, chosen from a set of 8 different and standard illuminants (A, B, C, D50, D55, D75, D93, F2)[12]. Resulting XYZ values were then converted into sRGB values.

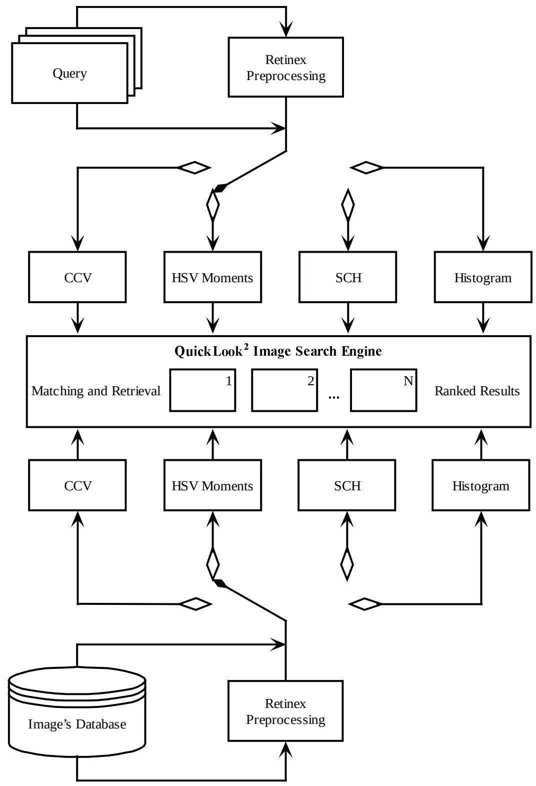


Fig. 3. The experiment scheme.

3.2. Color Features and Quantization

Each query strategy can be based on different image features. Since this work focused only on the use of color information, only color features have been considered. Among all the available color features we have chosen: the HSV color Moments, Color Histogram, Color Coherence Vector and Spatial Chromatic Histogram.

3.2.1. Quantization

The effective and efficient computation of the color features has required a drastic reduction in the number of colors used to represent the contents of our 24-bit images. Formally, we let C be a color space and $P = \{c_1, c_2, \ldots, c_i, \ldots, c_n \mid c_i \in C,\ n << \|C\|\}$, a subset of C, called quantization space. A function Q that maps each color in C to an element in P is called a *quantizer,* and is defined as:

$$Q : C \to P \tag{4}$$

We let I be a 24-bits $n \times m$ image. Colors in I, defined on the color space C, are reduced by applying the quantizer Q with respect to a palette of c chosen colors (P). We have chosen the HSV color space here, as color features in the HSV color space were reported, in[13], to yield better results during image retrieval than color features in other spaces. The subset P was selected by non-uniformly sampling the HSV color space at 64 intervals chosen to include colors with the same appearance. To reduce the quantization noise, and preserve the chromaticity contents, a vector majority filter was applied to the quantized image, using a working window W of 3x3 pixels. Letting $\widetilde{I}$ be the quantized image, at each pixel $\widetilde{I}(x,y)$ was assigned a value according to the following rule:

$$\widetilde{I}'(x,y) = c_k \quad with \ \ c_k = \underset{c_k}{argmax}\left\{\left\|A_{c_i}\right\|\right\} \ and \ \ A_{c_i} = \left\{\widetilde{I}(x,y) \middle| \widetilde{I}(x,y) = c_i, \widetilde{I}(x,y) \in W\right\} \tag{5}$$

3.2.2. HSV Color Moments

The color distribution of an image can be considered a probability distribution. Because any probability distribution is uniquely characterized by its central moments, color distribution can be characterized in the same manner. In[14], the first three moments (mean, variance and skewness) of each color channel of the HSV color space were used to evaluate image similarity. The feature entries for the i-th color channel were:

$$E_i(\tilde{I}) = \frac{1}{N}\sum_{x,y}\tilde{I}_i(x,y) \text{ that is, the average color channel values;}$$

$$\sigma_i(\tilde{I}) = \sqrt{\frac{1}{N}\sum_{x,y}\left(\tilde{I}_i(x,y) - E_i(\tilde{I})\right)^2} \text{ that is, the standard deviation;} \tag{6}$$

$$s(\tilde{I}) = \sqrt[3]{\frac{1}{N}\sum_{x,y}\left(\tilde{I}_i(x,y) - E_i(\tilde{I})\right)^3} \text{ that is, the third root of the skewness.}$$

The evaluation function proposed was a user specified weighted L_1 distance. Each feature entry was weighted by a values selected by the user based on the specific application.

$$d_{mom}(\tilde{I}_1,\tilde{I}_2) = \sum_i \left(w_{i1}\left|E_i(\tilde{I}_1) - E_i(\tilde{I}_2)\right| + w_{i2}\left|\sigma_i(\tilde{I}_1) - \sigma_i(\tilde{I}_2)\right| + w_{i3}\left|s_i(\tilde{I}_1) - s_i(\tilde{I}_2)\right|\right) \tag{7}$$

where $w_{i1}, w_{i2}, w_{i3} \geq 0$, and i=*H, S* and *V*. In our experiments we set all the weights to 1.

3.2.3. Color Histogram

Color histograms are frequently used to compare images because they are simple to compute, and tend to be robust against small changes in camera viewpoint. Retrieval using color histograms has been investigated in[1,15] for identifying objects in image databases. An image histogram refers to the probability mass function of image intensities. This is extended for color images to capture the joint probabilities of the intensities of the three color channels. More formally, each entry of the histogram is defined by:

$$h_{c_i}(\tilde{I}) = N \cdot Prob\left(\tilde{I}(x,y) = ci\right)\ x = 1...m, y = 1..n \ \ and \ \ N = m \times n \tag{8}$$

Computationally, the color histogram is formed by counting the number of pixels of each color.
There are several distance formulas for measuring the similarity of color histograms. Techniques for comparing probability distributions are not appropriate for color histograms because visual perception determines similarity rather than closeness of the probability distributions. One of the most commonly used measures for color histogram comparison is the histogram intersection[1]. The intersection formula is given by:

$$d\left(h(I_1),h(I_2)\right) = \frac{\sum min\left(h_{c_i}(I_1), h_{c_i}(I_2)\right)}{min\left(\left\|h(I_1)\right\|, \left\|h(I_2)\right\|\right)} \tag{9}$$

where $\left\|h(I_1)\right\|$ and $\left\|h(I_2)\right\|$ give the area of each histogram.

3.2.4. Color Coherence Vector

Color coherence vector histograms (CCV) are a refinement of color histograms proposed by[16], who classified the pixel of each image as *coherent* or *non-coherent* pixel. A pixel is said to be coherent if it is part of a large similarly-colored region;

otherwise, it is labeled as non-coherent. To extract the color regions, the image is quantized in n colors, and an 8-neighbor connected component algorithm is then applied. Pixels in regions of a size exceeding a predefined threshold (typically, 0.5-1% of the image size) are counted as coherent pixels, and those in smaller regions as non-coherent ones. For each color c_i two values are then computed: the number of coherent pixels, α_{c_i}, and the number of non-coherent pixels, β_{c_i}; each entry in the CCV is thus a pair (α_{c_i},β_{c_i}), and is called *coherence pair*. The whole coherence vector is defined as:

$$CCV(\tilde{I}) = \left\langle (\alpha_{c_1},\beta_{c_1}),\ldots,(\alpha_{c_i},\beta_{c_i}),\ldots,(\alpha_{c_n},\beta_{c_n}) \right\rangle \tag{10}$$

Clearly the sum $\alpha_{c_i}+\beta_{c_i}$ is the number pixels of color c_i present in the image; the set of the sums for i=1..n represent the color histogram. To compare two CCV's the L1 distance can be used:

$$\Delta CCV(\tilde{I},\tilde{I}') = \sum_{i=1}^{n} \left(\left|\alpha_{c_i} - \alpha'_{c_i}\right| + \left|\beta_{c_i} - \beta'_{c_i}\right| \right) \tag{11}$$

3.2.5. Spatial Chromatic Histogram

Spatial chromatic histograms (SCH)[17] are extended histograms that preserve not only information about the color content of the image, but also the spatial distribution of each color within the image. Each entry in a SCH is composed of three values: $h_{c_i}(\tilde{I})$, the ratio of pixels in $\tilde{I}$ of color c_i; $\mathbf{b}_{c_i}(\tilde{I}) = (\bar{x}_{c_i},\bar{y}_{c_i})$, the baricenter (in relative coordinates) of the spatial distribution of color c_i; and $\sigma_{c_i}(\tilde{I})$, the standard deviation of the distribution of color c_i. The elements of the SCH for an image are then:

$$S_{c_i}(\tilde{I}) = \left(h_{c_i}(\tilde{I}),\mathbf{b}_{c_i}(\tilde{I}),\sigma_{c_i}(\tilde{I})\right) \text{ where i=1…n.} \tag{12}$$

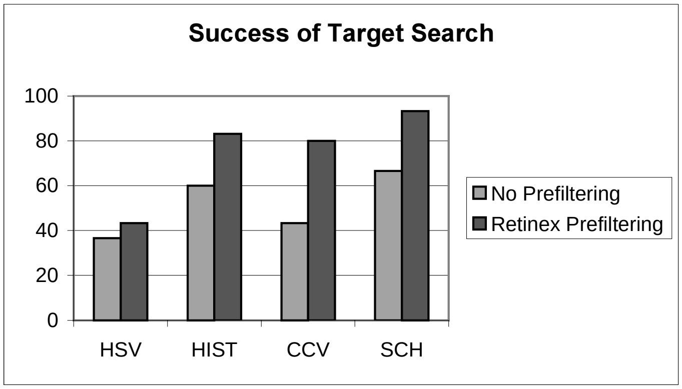
Letting A_{c_k} be the set of pixels in the image having the same color c_k, $A_{c_k} = \left\{\tilde{I}(x,y) \middle| \tilde{I}(x,y) = c_k\right\}$, the three elements of the SCH can then be computed as follows:

$$h_{c_k}(\tilde{I}) = \frac{\left|A_{c_k}\right|}{n \times m}, \text{ where n and m are the width and height of the image;}$$
$$\bar{x}_{c_k}(\tilde{I}) = \frac{1}{n}\frac{1}{\left|A_{c_k}\right|}\sum_{\tilde{I}(x,y)\in A_{c_k}} x, \qquad \bar{y}_{c_k}(\tilde{I}) = \frac{1}{m}\frac{1}{\left|A_{c_k}\right|}\sum_{\tilde{I}(x,y)\in A_{c_k}} y; \tag{13}$$
$$\sigma_{c_k}(\tilde{I}) = \sqrt{\frac{1}{\left|A_{c_k}\right|}\sum_{p\in A_{c_k}} d\left(p,b_{c_k}(\tilde{I})\right)^2}, \text{ where d( ) is the euclidean distance between two pixels.}$$

The similarity function proposed by the authors has been designed to separate color information from spatial information:

$$f(\tilde{I}_1,\tilde{I}_2) = \sum_{c_i}\left[min\left(h_{c_i}(\tilde{I}_1),h_{c_i}(\tilde{I}_2)\right) * \left(\frac{\sqrt{2} - d\left(\mathbf{b}_{c_i}(\tilde{I}_1),\mathbf{b}_{c_i}(\tilde{I}_2)\right)}{\sqrt{2}} + \frac{min\left(\sigma_{c_i}(\tilde{I}_1),\sigma_{c_i}(\tilde{I}_2)\right)}{max\left(\sigma_{c_i}(\tilde{I}_1),\sigma_{c_i}(\tilde{I}_2)\right)} \right) \right] \tag{14}$$

4. RESULTS AND DISCUSSION



Feature	Preprocessing	STS (%)
HSV	None	36,665
	Retinex	43,335
HIST	None	60,000
	Retinex	83,165
CCV	None	43,330
	Retinex	80,000
SCH	None	66,670
	Retinex	93,335

Fig. 4. Summary result of Target Search experiments.

To quantify the performance of each retrieval strategy in a global score, we have defined the Success of Target Search index (STS) as the ratio between the number of images retrieved in the first position and the total number of queries. Figure 4 summarizes the STS values for each feature used in retrieval, first with no filtering and then with Retinex pre-filtering. When the Retinex algorithm is applied, the results outperform those when the pre-filtering is not applied, regardless of the feature used in the retrieval. The improvement is less pronounced in case of HSV moments, than for other features (15%). This result is still under investigation, but at first sight the reason could be that on very dark images (on which the feature fails), Retinex introduces an unwanted shift in lightness[18]. In the other cases, the improvement is 27.85%, 45.83% and 28.56% for the color histogram, the color coherence vector and the spatial chromatic histogram respectively.

In conclusion, Retinex pre-filtering improves the retrieval effectiveness of any of the retrieval strategies implemented. A simple comparison of the improvements in performance suggests that better results can be obtained when Retinex is coupled with color quantization.
The prime advantage of the Retinex algorithm used is its efficiency; future work will include an in-depth comparison with other Retinex algorithms, operating on much larger databases.

5. BIBLIOGRAPHY

1. M. Swain, and D. Ballard, “Color Indexing”, *International Journal of Computer* Vision, **7:1**, pp. 11-32, 1991.
2. B. V. Funt, G. D. Finlayson, “Color Constant Color Indexing”, *IEEE Transactions on PAMI*, Vol. **17**, pp. 522-529, 1995.
3. T.Gevers, A. W. M. Smeulders, “A Comparative Study of Several Color Models for Color Image Invariant Retrieval”, *Proc. IDB-MMS’96*, 1996.
4. T.Gevers, A. W. M. Smeulders, “Color Based Object Recognition”, *Pattern Recognition*, Vol. **32**, pp.453-464, 1999.
5. D. Marini, A. Rizzi, “A Color Appearance Approach to Image Database Visual Retrieval”, EI2000, IS&T/SPIE's Electronic Imaging 2000, San Jose’, California, USA, 24-28/1/2000.
6. E. Land, J. McCann , “Lightness and Retinex Theory”, *Journal of Optical Society of America*, **61:1**, pp. 1-11, 1971.
7. D. Marini, A. Rizzi, L. De Carli, “Multiresolution retinex : comparison of algorithms.” First International Conference on Color in Graphics and Image Processing CGIP'2000, Saint-Etienne, France, October 1-4, 2000.
8. D. Saupe, “Algorithms for random fractals”, Barnsley et all, *The Science of Fractal Images*, Springer Verlag, New York, 1988.
9. D. Marini, A. Rizzi, “A computational approach to color adaptation effects”, *Image and Vision Computing*, Vol. **18** (13), pp. 1005-1014, 2000.
10. G. Ciocca, I. Gagliardi, R. Schettini, “Quicklook^2: An Integrated Multimedia System”, *International Journal of Visual Languages and Computing Special Issue on Multimedia Databases and Image Communication*. (in print)
11. M. D. Fairchild, *Color Appearance Models*, Addison-Wesley, Reading, Massachusetts, 1998.
12. R. W. G. Hunt, *Measuring Colour*, Third Edition, Fountain Press, Kingston-upon-Thames, England, 1998.

13. B. Furht, *The Handbook of Multimedia Computing: Chapter 13 – Content-Based Image Indexing and Retrieval.* LLC: CRC Press, 1998.
14. M. Stricker and A. Dimai, "Color Indexing with Weak Spatial Contraints", *Proc. Storage and Retrieval for Image and Video Databases* IV, 1996.
15. J. R. Smith, and S. F. Chang, "Tools and Techniques for Color Image Retrieval", Columbia University CTR Technical Report, June 1995.
16. G. Pass, R. Zabih, J. Miller, "Comparing Images Using Color Coherence Vectors", *Proc. Fourth ACM Multimedia Conference*, 1996.
17. L. Cinque, S. Levialdi, A. Pellicano', "Color-Based Image Retrieval Using Spatial-Chromatic Histograms", *IEEE Multimedia Systems 99,* Vol. **II**, pp. 969-973, 1999.
18. D. Marini, A. Rizzi, M. Rossi, "Color Constancy Measurement for Synthetic Image Generation", *Jounal of Electronic Imaging*, Vol. **8**:**4**, pp. 394-403, October 1999.