

Random Spray Retinex: a new Retinex implementation to investigate the local properties of the model

Edoardo Provenzi*, Massimo Fierro, Alessandro Rizzi, Luca De Carli, Davide Gadia and Daniele Marini

Abstract—In order to investigate the local filtering behavior of the Retinex model, we propose a new implementation in which paths are replaced by 2-dimensional pixel sprays, so the name ‘Random Spray Retinex’. A peculiar feature of this implementation is the way its parameters can be controlled to perform spatial investigation. The parameters tuning is accomplished by an unsupervised method based on quantitative measures. This procedure has been validated via user panel tests. Furthermore, we prove that the spray approach has faster performances than the path-wise one. Tests and results are presented and discussed.

Index Terms—Retinex, pixel sprays, locality of color perception.

EDICS categories: MOD-PERC, COL-COLR.

I. INTRODUCTION

The human visual system (HVS) does not perceive the color of an area independently from the visual scene in which it lies, instead it is heavily influenced by the chromatic content of the other areas of the scene. This psychophysiological phenomenon is the *locality of color perception*. One of the earliest models able to deal with locality of perception has been Retinex by E. Land and J.J. McCann [1]. The scientific community has always been interested in this model and its various applications, as reported in [2].

In the basic Land and McCann implementation of Retinex, locality is achieved through paths scanning images.

A great amount of implementations and analysis followed after this first work. These can be divided into two major groups, differing from the way they achieve locality: *sampling* the chromatic content around a pixel [3], [4], [5], [6], [7], [8], or *integrating* it [9], [10], [11], [12], [13], [14], [15].

All the sampling implementations that use a path-wise approach have to deal with the following problems: strong dependency on paths geometry, high computational cost and sampling noise.

On the basis of a recent mathematical analysis of path-wise Retinex algorithms [16] we will prove the intrinsic redundancy of this approach. Consequently, we will propose an extension that allows to keep the sampling approach highly reducing the typical problems related to the use of paths.

Edoardo Provenzi, Massimo Fierro and Alessandro Rizzi are with the Department of Information Technology, University of Milano, Via Bramante 65, 26013, Crema (CR), Italy (e-mail: {provenzi, fierro, rizzi}@dti.unimi.it).

Luca De Carli, Davide Gadia and Daniele Marini are with the Department of Information and Communication, University of Milano, Via Comelico 39, 20135, Milano, Italy (e-mail: {decarli, gadia, marini}@dico.unimi.it).

This alternative technique is constructed replacing paths with random sprays, i.e. two-dimensional point distributions across the image, so the name ‘Random Spray Retinex’ (RSR). We will show how it is possible to change the spray density around a pixel and how this leads to the ability of finding out information about locality of color perception within the Retinex model.

The structure of the paper is the following: in Section II we remind the basic information contained in [16] about the mathematical formulation of path-wise Retinex implementations and its consequences on the intrinsic properties of the model. In Section III we motivate the passage from paths to sprays, that is be used in Section IV to implement RSR and discuss its properties. Section V is dedicated to the tuning of RSR parameters. Finally, in Section VI we prove that RSR has faster performances than those of path-wise Retinex implementations.

II. MATHEMATICAL DESCRIPTION OF RETINEX

The starting point of our analysis is the mathematical description of Retinex given in [16] and briefly recalled here. Consider a RGB digital image and a collection of N paths $\gamma_1, \dots, \gamma_N$ composed by ordered chains of pixels starting in j_k and ending in i (called *target pixel*). Let n_k be the number of pixels traveled by the k -th path γ_k and let $t_k = 1, \dots, n_k$ be its parameter, i.e. $\gamma_k : \{1, \dots, n_k\} \rightarrow \text{Image} \subset \mathbb{R}^2$, $\gamma_k(1) = j_k$ and $\gamma_k(n_k) = i$. We indicate two subsequent pixels of the path as $\gamma_k(t_k) = x_{t_k}$ and $\gamma_k(t_k + 1) = x_{t_k + 1}$, for $t_k = 1, \dots, n_k - 1$. In Retinex the computations are performed separately in every chromatic channel $c \in \{R, G, B\}$, to avoid a cumbersome notation we omit its specification. So we write $I(x_{t_k})$, $I(x_{t_k + 1})$ to indicate the intensities of x_{t_k} and $x_{t_k + 1}$ in the three separated chromatic channels. We write their ratio as $R_{t_k} = \frac{I(x_{t_k + 1})}{I(x_{t_k})}$, with $R_0 \equiv 1$. For technical reasons it is useful to normalize the intensities to take their values in the real unit interval $(0, 1]$, we omit the zero value because we must perform divisions. The usual way to do this is to add a small positive constant to all pixel intensities.

In [16] it has been proved that the intensity of i is re-computed by Retinex to give the (normalized) *lightness* $L(i)$ through the function that follows:

$$L(i) = \frac{1}{N} \sum_{k=1}^N \prod_{t_k=1}^{n_k-1} \delta_k(R_{t_k}) \quad (1)$$

where $\delta_k : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, $k = 1, \dots, N$, are functions defined in this way: $\delta_k(R_0) = 1$ and, for $t_k = 1, \dots, n_k - 1$,

$$\delta_k(R_{t_k}) = \begin{cases} R_{t_k} & \text{if } 0 < R_{t_k} \leq 1 - \varepsilon \\ 1 & \text{if } 1 - \varepsilon < R_{t_k} < 1 + \varepsilon \\ R_{t_k} & \text{if } 1 + \varepsilon \leq R_{t_k} \leq \frac{1+\varepsilon}{\prod_{m_k=0}^{t_k-1} \delta_k(R_{m_k})} \\ \frac{1}{\prod_{m_k=0}^{t_k-1} \delta_k(R_{m_k})} & \text{if } R_{t_k} > \frac{1+\varepsilon}{\prod_{m_k=0}^{t_k-1} \delta_k(R_{m_k})} \end{cases} \quad (2)$$

being $\varepsilon > 0$ a fixed *threshold*.

When the first or the third option are satisfied, δ_k acts simply as the identity function and the formula implements the typical ‘chain of ratios’ of Retinex [1].

The second option occurs when only a very small change of intensity is measured between two subsequent pixels. In this case $\delta_k(R_{t_k}) = 1$, so that the product of ratios remains unchanged with respect to the previous step. This implements in a mathematical fashion the so-called *threshold mechanism*.

Finally, when the fourth situation is satisfied, i.e. when $\delta_k(R_1)\delta_k(R_2)\cdots\delta_k(R_{t_k-1})R_{t_k} > 1 + \varepsilon$, then δ_k resets the chain of product to 1 so that the last pixel traveled becomes the local white reference. This is the mathematical implementation of the *reset mechanism*. The *white patch behavior* of the algorithm is determined by this operation.

All the options together realize the well known ratio-threshold-reset mechanism of Retinex.

This description of the original Retinex algorithm is exhaustive but not predictive because of the presence of the threshold mechanism. However, it is well known that the best Retinex performances correspond to very small values of ε . This fact has been used in [16] to show that, for such values of ε , the difference between Retinex with and without threshold has an upper bound which has negligible effects for the final lightness computation. In [16] it has been proved that the really important mechanism of Retinex is not the threshold one, instead it is the reset that determines the Retinex filtering properties. Hence it is not reductive to study Retinex fixing $\varepsilon = 0$, as we are going to do, from now on, until the rest of the paper.

The great advantage to neglect the threshold mechanism is that the mathematical formulation undergoes a significant simplification and it can be proved [16] that the lightness value can be written with this closed formula:

$$L(i) = \frac{1}{N} \sum_{k=1}^N \frac{I(i)}{I(x_{H_k})} \quad (3)$$

where x_{H_k} is the pixel with highest intensity traveled by the path γ_k , for every $k = 1, \dots, N$.

It is immediate to see that $L(i)$ is the average of the single path contributions $L_k(i) = \frac{I(i)}{I(x_{H_k})}$, hence, the only information that characterizes every contribution is the value of the pixel x_{H_k} with highest intensity traveled by the path γ_k .

It follows that the problem of finding $L_k(i)$ reduces to the problem of finding the maximum value assumed by the image function $I(x, y)$ along γ_k , for every $k = 1, \dots, N$. We stress that *this result is independent from paths geometry*.

Finally, we remember that the analysis of formula (3) performed in [16] revealed *three intrinsic characteristics of Retinex*: the first is that its filtering properties are strongly image-dependent; the second is that, when paths length tend to very big values (compared with the image size), Retinex loses its local properties and becomes a global white-patch algorithm. Finally, the third intrinsic property is that Retinex is not an idempotent operator, i.e., in general, $L(L(i)) \neq L(i)$, but subsequent iterations of Retinex converge to a fixed image qualitatively characterizable as follows: every path travels at least one pixel with intensity 1 in every fixed chromatic channel. The visual consequence is a sort of speckling in the image of convergence (more details about this topic can be found in [16]).

III. FROM PATHS TO PIXEL SPRAYS

The information given by the mathematical formulation of Retinex have strong consequences on the structure of $\mathcal{P}_i(\text{Image})$: the set of paths embedded in the image and *ending in the point i*. After formula (3), on this set it is natural to define this equivalence relation: given $\gamma, \eta \in \mathcal{P}_i(\text{Image})$,

$$\gamma \sim \eta \Leftrightarrow \max_{(x,y) \in \gamma^*} \{I(x,y)\} = \max_{(x,y) \in \eta^*} \{I(x,y)\} \quad (4)$$

where γ^* and η^* are the codomain of the paths, i.e. the collections of pixels traveled by γ and η , respectively.

Paths belonging to different equivalence classes give different contributions to the lightness computation, while every path in a given equivalence class is characterized by the same value of $L_k(i)$. It immediately follows that, for the purposes of Retinex, $\mathcal{P}_i(\text{Image})$ contains redundant paths and so the really interesting set of paths is the *quotient set* $\mathcal{P}_i(\text{Image})/\sim$, whose elements are the equivalence classes of paths with respect to the equivalence relation defined in (4). In each equivalence class one can choose a single representative path to compute $L_k(i)$, in particular, the shortest one is the two-points path whose codomain is simply $\{x_{H_k}, i\}$. It follows that the ordering operations needed to generate the paths are perfectly uninfluential for the final lightness computation.

Moreover, by a mathematical point of view, paths are topological manifolds of dimension 1 embedded in the image, which is a topological manifold of dimension 2, so paths do not really scan local neighborhoods of a pixel, but rather *particular directions* in these neighborhoods. This directional extraction of information can lead to halos or artifacts in the filtered image.

The classical implementations of Retinex try to remedy this problem using a large number of paths, but this increases the filtering time and does not really overcome the problem.

We see that there are three reasons for which paths are not perfectly suitable for the analysis of locality of color perception within the Retinex model: they are redundant, their ordering is completely uninfluential and they have inadequate topological dimension.

Thus, we are lead to use 2-dimensional objects such as *areas* instead of 1-dimensional paths to analyze locality of color perception. More precisely, our idea is to implement the investigation about locality selecting pixels from these

areas with a density sample that changes according to a given function of their distance with respect to the target pixel i . Each function generates a different kind of pixel selection around i , leading to different kind of ‘sprays’, each of which reveals different local filtering properties.

The next sections explain in detail how it is possible to make these ideas concrete, generating the new implementation of Retinex that we have denoted with *RSR*: ‘*Random Sprays Retinex*’.

IV. RSR IMPLEMENTATION

RSR is a new implementation of the original Retinex model [1] which has been inspired from the results of the mathematical analysis of Retinex performed in [16]. In RSR the role of a path γ_k traveling n_k pixels and ending in the target i is played by $\text{Spray}_k(i)$, a spray composed by n_k pixels and centered in i . The typical ratio-reset operation along a path is substituted by the search of the pixel with highest intensity in the whole spray. It will be clear from the following discussion that, once the number of points per spray is chosen, there is no need to vary it with k , hence, from now on, we will write n instead of n_k to denote the number of pixels per spray.

The functional expression of the formula (3) to compute the lightness remains exactly the same in both algorithms, so they share the same intrinsic properties recalled at the end of section II. This is the reason why the results about locality of color perception that we will get thanks to the *RSR implementation* can be referred to the *Retinex model*.

Let us now show how to construct $\text{Spray}_k(i)$. With a random point generator we can get a *uniform random distribution* of n values in the real unit interval $[0, 1]$. Then, by multiplication, we can extend this distribution to any real interval, in particular, we are interested to the intervals $[0, 2\pi]$ and $[0, R]$, where R is a given positive real number that will represent the radius of the spray. We denote, respectively, with $\text{RAND}_n[0, 2\pi]$ and $\text{RAND}_n[0, R]$ the corresponding uniform random distributions.

Now, if (i_x, i_y) are the coordinates of i , we can define the polar coordinates of a generic pixel $j \equiv (j_x, j_y)$ belonging to $\text{Spray}_k(i)$ in this way:

$$\begin{cases} j_x = i_x + \rho \cos(\theta) \\ j_y = i_y + \rho \sin(\theta) \end{cases} \quad (5)$$

where $\rho \in \text{RAND}_n[0, R]$, $\theta \in \text{RAND}_n[0, 2\pi]$.

These are the coordinates of pixels that have an isotropic angular distribution in a circle of radius R centered in the pixel i .

Notice, however, that the radial density is not isotropic, in fact, because of the rotation, the spray results more dense near the target pixel i than far away. We can easily compute the *mean radial density* $\rho(r)$ considering, as in fig. 1, a circle C_r of arbitrary radius r , $0 < r < R$, centered in i and then taking the ratio between the mean number of points inside C_r and its area.

Since we are dealing with uniform random distributions, the mean number of points inside C_r is $n \frac{r}{R}$, hence

$$\rho(r) = \frac{n \frac{r}{R}}{\pi r^2} = \frac{n}{\pi R r}. \quad (6)$$

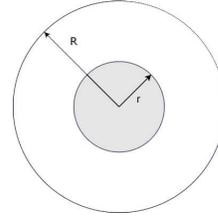


Fig. 1. Computation of the mean areolar density in function of the spray radius.

Thus the mean radial density of spray pixels decreases as the inverse radius.

Fig. 2 shows an example of such a spray with 400 pixels and radius $R = 1$.

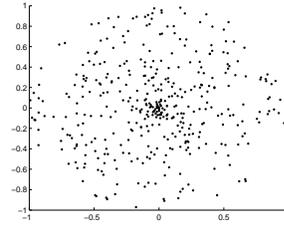


Fig. 2. An example of ‘naturally localized’ spray.

The angular isotropy is a natural requirement that must be satisfied by the spray, since the presence of privileged directions generates artifacts in the filtered image. For example, if we consider a spray with the shape showed in fig. 3, then the corresponding filtered image exhibits haloes independently on how many sprays or pixels per spray we use.

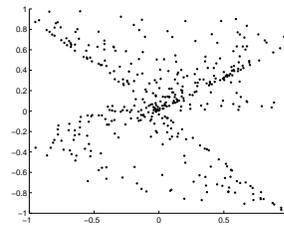


Fig. 3. An example of anisotropic spray.

An example of these haloes can be seen in fig. 5, which shows the picture in fig. 4 filtered with 20 anisotropic sprays per pixel, 400 pixels per spray and $R = \text{DIAG}$, where DIAG is the value of the image diagonal.



Fig. 4. Original ‘gallery’ image.



Fig. 5. Haloes induced by anisotropic sprays.

Now, the local properties of Retinex can be analyzed in a very simple way applying a function on the coordinate ρ to change the radial density of the spray pixels around i . Precisely, given any function $f: \mathbb{R}^+ \rightarrow \mathbb{R}^+$, we can consider the modified spray whose pixels have polar coordinates defined by:

$$\begin{cases} x = x_i + f(\rho) \cos(\theta) \\ y = y_i + f(\rho) \sin(\theta) \end{cases} \quad (7)$$

where, again, $\rho \in \text{RAND}_n[0, R]$ and $\theta \in \text{RAND}_n[0, 2\pi]$. It is useful to distinguish the special case in which $f \equiv id_{\mathbb{R}^+}$, the identity function restricted on non-negative real numbers, calling the corresponding spray ‘naturally localized’.

Figs. 6-9 show some examples of sprays with $R = 1$, $n = 400$, obtained with different functions f .

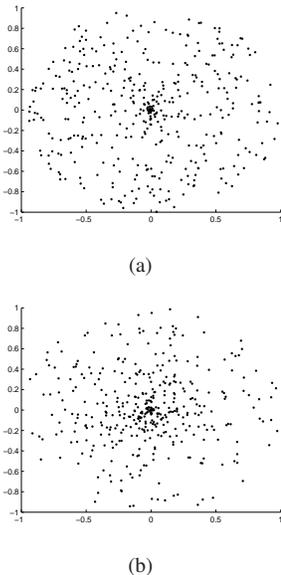


Fig. 6. (a) Spray with $f(\rho) = \frac{\log(1+\rho)}{\log(2)}$. (b) Spray with $f(\rho) = \frac{\sinh(\rho)}{\sinh(1)}$.

It can be seen that the normalized logarithmic and hyperbolic sinus functions keep the spray density quite similar to the one of the naturally localized spray. Instead powers of ρ with exponent greater than 1 and the normalized exponential function tend to increase the density around the center. Finally, powers of ρ with exponent in $(0, 1)$ and the normalized inverse exponential applied on ρ tend to delocalize the spray. The multiplication of ρ by a constant coefficient m simply changes the radial extension of the naturally localized spray, expanding the radius, when $m > 1$, or contracting it, when $0 < m < 1$.

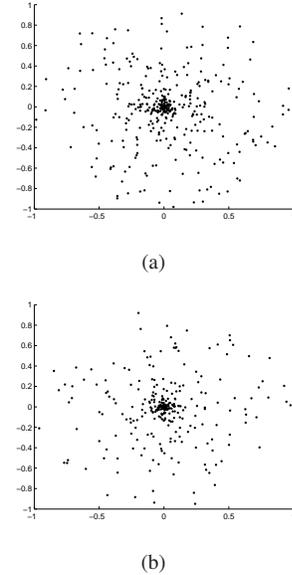


Fig. 7. (a) Spray with $f(\rho) = \rho^2$. (b) Spray with $f(\rho) = \rho^4$.

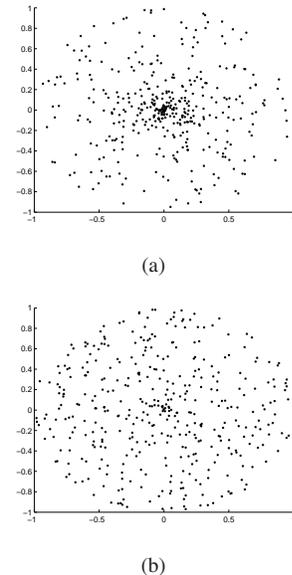


Fig. 8. (a) Spray with $f(\rho) = \frac{e^\rho - 1}{e - 1}$. (b) Spray with $f(\rho) = \frac{e^{-\rho} - 1}{e^{-1} - 1}$.

To perform the analysis of locality in RSR we must tune f and the other parameters of the algorithm. Before showing the results about tuning, we briefly summarize all these parameters and discuss their meaning in the next subsection.

A. RSR parameters and their meaning

RSR depends on four parameters: R (the radius of the sprays), f (the radial density function) N (the number of sprays) and n (the number of pixels per spray).

The radius R of the spray defines the extension of the circular area analyzed around the pixel i . This area must be tuned to get enough information about the color distribution around i .

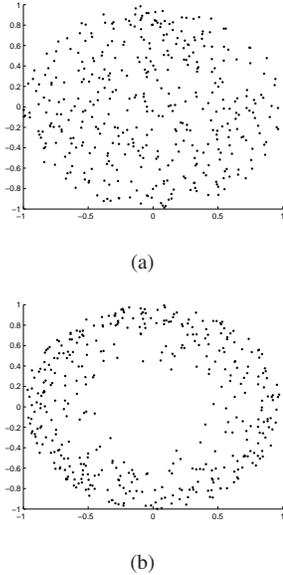


Fig. 9. (a) Spray with $f(\rho) = \sqrt{\rho}$. (b) Spray with $f(\rho) = \sqrt[5]{\rho}$.

As already stated, the function f changes the radial density of the spray pixels. It must be tuned to find out what is the spray pixel distribution that better fits the computational reproduction of color perception performed by the HVS.

For each $\text{Spray}_k(i)$ there is a non zero probability to find the pixel with highest intensity x_{H_k} in an isolated pixel not related to the context. This, of course, would produce chromatic noise in the filtered image. Since the spray pixels are generated by a random point generator, all the N sprays are different and so, statistically speaking, the influence of isolated pixels on the global computation of $L(i)$ decreases when we average many sprays contributions $L_k(i)$. Hence, the higher is the number of sprays, the lower is the chromatic noise in the filtered image. This is confirmed by the tests performed (as will be discussed later), which also shown that, to avoid pattern replication all across the image, the sprays must be taken by a pre-computed set of, at least, a thousand sprays.

Finally, the number n of pixels per spray determines how much information is extracted from the spray area. If we use very large values of n we cover all the spray area, losing the locality of the spray distribution, instead, if we use small values of n , we cannot get enough information to correctly compute $L(i)$.

V. TUNING RSR PARAMETERS

We performed our tests on a set of over 100 very different pictures given by real-world images, portraits, landscapes and geometric images.

A. Tuning the spray radius

The easiest parameter to tune has proved to be the radius: for all images and independently from the other parameter of RSR, our tests showed that the optimal value for R is DIAG , the value of the diagonal of the image.

The reason is easily comprehensible: if one uses a smaller radius, then two pixels that lie near the extreme points of the diagonals can never be compared. The effect of using a smaller radius than DIAG can be clearly seen comparing figs. 10 and 11, which have been filtered with $R = \text{DIAG}/2$ and $R = \text{DIAG}$, respectively, keeping all the other parameters constant: ρ as radial coordinate, $N = 30$, $n = 800$.



Fig. 10. An image filtered with spray radius $R = \frac{\text{DIAG}}{2}$.



Fig. 11. The same image as in fig. 10 filtered with spray radius $R = \text{DIAG}$.

Furthermore, it is not useful to use a radius larger than DIAG , since the spray loses part of its density around the target pixel and many spray points lie outside the image area.

B. Tuning the radial density function

The radial density of the spray is responsible for the local property of RSR because the probability to find out the pixel with highest intensity in the spray is greater in the image areas where the spray is denser, than in the image regions where the spray has only few points.

It is well known that tests about human color perception show that the chromatic influence between two pixels decreases with their distance (e.g. [17], [18], [19]). This fact is implemented in every color perception model: path-wise algorithms (e.g. [6]) sample the image content with paths that are denser in the immediate neighborhood of the target pixel than far away, while integrative algorithms (e.g. [11]) use a center/surround technique that weights the surround of the target pixel with monotonically decreasing functions.

Coherently with this, even RSR revealed that delocalized sprays are inadequate to correctly simulate color perception by the HVS. For example, fig. 12 shows an image filtered using $\sqrt[5]{\rho}$ as radial coordinate.

As a consequence, the only interesting radial density functions are those that correspond to monotonically decreasing radial densities. Only such functions will be considered in the next discussion.

We conducted the tuning using both subjective quality match tests and quantitative tests about color constancy.



Fig. 12. Effects of a spray with radial coordinate $\sqrt[5]{\rho}$.

The first kind of tests has been developed as follows: we filtered our test set of images fixing n and N and varying the radial density function. We have displayed the images on a middle gray background of a calibrated monitor in a dark room. Then we asked a collection of users to indicate in a scale between 1 (poor) and 5 (excellent), the degree of naturalness (color plausibility in relation to the personal experience), absence of noise and detail visibility of the filtered images. The results of our tests, averaged on the three questions and on the test set images, are shown in fig. 13.

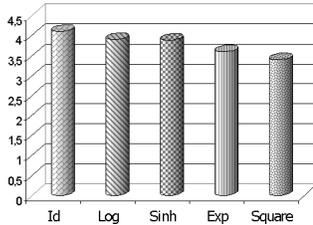


Fig. 13. Quality test for different radial density functions.

The images filtered with the naturally localized spray have always received the best judgement by the users. Starting from ρ^2 , the sprays results too localized and the corresponding filtered images show an increasing amount of noise, as can be seen in fig. 14, that has been filtered with ρ^4 as radial coordinate and with $N = 30$, $n = 800$ (to be compared with fig. 11, which has been filtered with the same values of n and N , but with ρ as radial coordinate).



Fig. 14. Noise induced by a spray with radial coordinate ρ^4 .

Regarding color constancy tests, we considered the pictures of the database described in [20], consisting in a series of photographs taken under different color casts. We filtered each series of pictures with different radial density functions. Then we computed the CIELab differences between the images filtered with every given radial density function to have a measure of the corresponding algorithm ability to reduce color cast.

For more readability we report only the results of our tests on the picture in fig. 15 taken under three different casts: Cast 1 = PHILIPS Neon Neutral Daylight 6500K (TLD965), Cast 2 = PHILIPS Neon Fluotone 4100K (TLD840), Cast 3 = PHILIPS Neon Daylight 5000K (TLD950). Tests with the others casts shown analogous results.



Fig. 15. An image of the database in [20] for color constancy tests.

The values visualized in the graphics of fig. 16 correspond to the parameters $N = 20$ and $n = 400$, when these parameters are varied the numerical values of the differences change, but the relationship between the different radial density functions does not change.

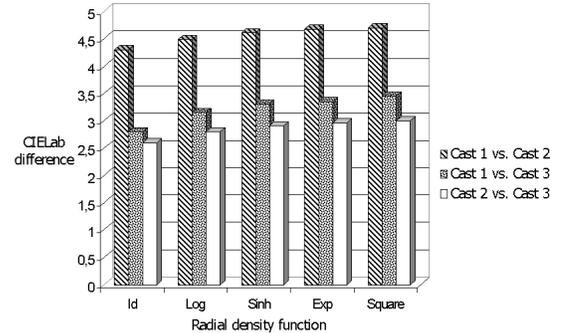


Fig. 16. Color constancy test for different radial density functions.

It can be seen that the density function that minimizes the CIELab difference between the filtered images is the identity function. Tests on the other images exhibit analogous results.

The consequence of our subjective and quantitative tests is that the naturally localized spray is the most suitable to reproduce the behavior of the HVS within the RSR implementation of the Retinex model. From now on, RSR will be considered only with ρ as radial coordinate.

We recall from formula (6) that the mean areolar density of the naturally localized spray decreases as the inverse distance from the center. It follows that, in RSR, fixed a pixel i , every other pixel of the image, considered as a single entity, has a 'mean chromatic influences' on i that decreases as the inverse distance from i . This fact implies that, statistically speaking, the chromatic influence of pixels close to i is comparable only with that of entire areas of pixels far from i , the wideness of which must increase, according to formula (6). This seems to be a good motivation to study multilevel extensions of RSR.

Finally, we notice that the result of this section corresponds to what found in the tuning experiments of another color perception model: ACE (Automatic Color Equalization) [21]. In that algorithm the target pixel is compared with the other image pixels, each of which is weighted with a coefficient. In

[21] it has been shown that the optimal weight coefficients are the inverse distances from the target.

C. Tuning the number of sprays and pixels per spray

One of the consequences of the mathematical analysis performed in [16] is that, as the paths length of a path-wise Retinex implementation grows to great values, the algorithm loses its local properties showing a global white patch behavior. The tuning of paths length or number is still an open problem for path-wise Retinex implementations.

We are now going to show that, with the RSR implementation, it is possible to perform an unsupervised tuning of the parameters N and n in a self-consistent way, highly reducing the range of their optimal values.

These two parameters are strictly related because the lightness is computed averaging the N contributions of the sprays, each of which depends on how many points are used to find out the pixel with highest intensity.

We carried out the tuning as follows. We filtered the images of our test set increasing N from 5 to 60 with a constant step of 5 sprays and increasing n from 250 to 900 with a constant step of 50 points. Then we calculated $\Delta E_N(n)$ and $\Delta E_n(N)$, the CIELab differences between the images filtered with a fixed value of N and two consecutive values of n , and viceversa, with N playing the role of n . We observed that both $\Delta E_N(n)$ and $\Delta E_n(N)$ decrease monotonically for all images.

Now, since two images are considered chromatically indistinguishable if $\Delta E < 1$, it is natural to tune N and n taking the smallest values of these parameters for which this inequality holds true. In other words, this procedure is a natural compromise between the minimization of filtering time and the maximization of filtering quality.

To have a quantitative example to discuss, let us consider the tests performed on the image in fig. 17.



Fig. 17. Image for the tuning of the parameters n and N .

The interpolation graphic of ΔE , viewed as function of N and n , and its intersection with the hyperplane $\Delta E \equiv 1$, is visualized in fig. 18.

Fig. 19 and fig. 20 represent the interpolation graphics of the functions $\Delta E_N(n)$ and $\Delta E_n(N)$, which are the level curves of the surface in fig. 18. In the horizontal axis are indicated the two consecutive values of n or N corresponding to the CIELab difference values displayed in the graphic. Only the significant part of the curves are visualized.

Since the parameters n and N control two different characteristics of the filtered image, it is not sensed to take high values of n and little values of N or viceversa, because the

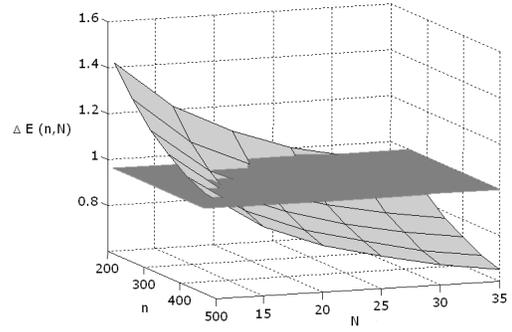


Fig. 18. Surface of $\Delta E(n, N)$ intersecting the hyperplane $\Delta E \equiv 1$.

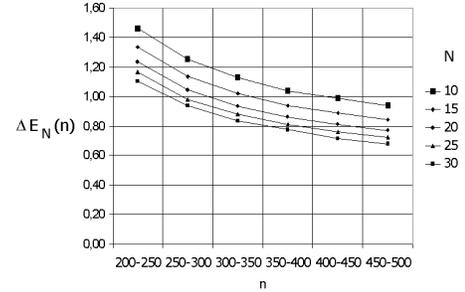


Fig. 19. Graphics of $\Delta E_N(n)$ for different values of N .

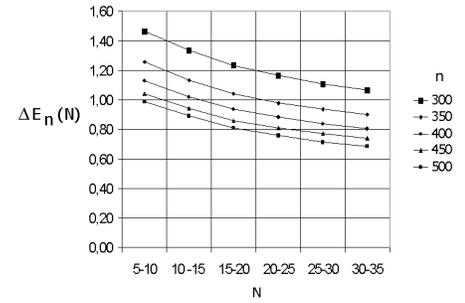


Fig. 20. Graphics of $\Delta E_n(N)$ for different values of n .

corresponding image would have good chromatic quality, but high chromatic noise, or viceversa, respectively. Instead the optimal couple (N, n) must be chosen as the ‘minimal’ couple of intermediate values of N and n such that the surface $\Delta E(n, N)$ lies under the hyperplane $\Delta E \equiv 1$, where with ‘minimal’ we mean the couple that minimizes the product $n \cdot N$. For example, it can be seen from figs. 19 and 20 that both the couples $(N = 25, n = 350)$ and $(N = 20, n = 400)$ correspond to intermediate values of N and n such that the surface lies under $\Delta E \equiv 1$, but $25 \cdot 350 = 8750 > 20 \cdot 400 = 8000$, so that the optimal choice is $(N = 20, n = 400)$, because it corresponds to 750 operations per pixels less than the other couple.

We combined the procedure just described with subjective matches analogous to those performed for the tuning of the radial density function, but now changing every time the values of N and n . The results of the tests performed on the image shown in fig. 17 are presented in fig. 21. The surface is obtained interpolating the values at the nodes (n, N) , the value at each node is calculated averaging the degree of naturalness,

absence of noise and detail visibility indicated by the users.

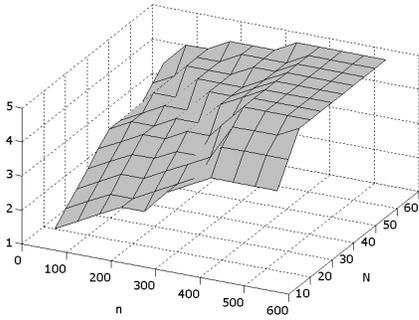


Fig. 21. Quality test for the parameters n and N with respect to fig. 17.

It can be seen from the graphic in fig. 21 that the surface reveals a wide constant area after the couple of parameters (N, n) overcomes $(20, 400)$, as predicted by the quantitative procedure described above. It is evident that there is no reason to increment the filtering time taking greater values for N and n . All the other tests performed has revealed agreement between the unsupervised procedure described above and the subjective tests involving users.

Now that we described the tuning procedure, we show in figs. 22-24 some output results of RSR with tuned parameters.



Fig. 22. ‘Gallery’ filtered with tuned parameters: $N = 25$, $n = 800$.



(a)



(b)

Fig. 23. (a) Original ‘Books’ image; (b) ‘Books’ filtered with tuned parameter $N = 25$, $n = 750$.

As can be seen from the different values of optimal values of N and n for the various images, the tuning of N and n strongly depends on the different image content. The problem to find out a formula to precisely determine the variation of the parameters N and n in relation with the image content still remains open.



(a)

(b)

Fig. 24. (a) Original ‘flowers’ image; (b) ‘Flowers’ filtered with tuned parameter $N = 20$, $n = 550$.

D. Filtering the same image with different sizes

If we consider a given image at different sizes, then we need a formula to extend the optimal values of N and n for a certain size to the other sizes of the same image. Our tests shown that the optimal value of N remains constant, but, as expected, the optimal value of n changes. In fact n determines the amount of information needed to compute the lightness and obviously this amount must increase or decrease in relation with the image size.

We can formalize the problem in this way: suppose we have the same image at the sizes $W_0 \times H_0$ and $W_1 \times H_1$, and suppose that $n^{\text{opt}}(W_0, H_0)$, the optimal value of n for the image of width W_0 and height H_0 , is known. The easiest way to find out $n^{\text{opt}}(W_1, H_1)$, the optimal value of n for the image of width W_1 and height H_1 , is to impose this mathematical proportion:

$$n^{\text{opt}}(W_1, H_1) : W_1 H_1 = n^{\text{opt}}(W_0, H_0) : W_0 H_0 \quad (8)$$

i.e. to impose the fraction of spray pixels per unit of image area to remain constant.

Applying the unsupervised tuning procedure previously described, all the images of our test set shown that, once the optimal value of n is found for a given image size, formula (8) enables to correctly compute the changes of the optimal value of n in function of the new image sizes.

VI. RSR VS. PATH-WISE RETINEX EXECUTION TIME

As we have already recalled in the introduction, one of the major drawbacks of path-wise Retinex implementations is their high computational cost. In this section we are going to show that the RSR implementation, besides the possibility to analyze the locality of color perception and to tune all its parameters, provides another advantage with respect to path-wise Retinex implementations: even though the two methods share the same computational cost, $\mathcal{O}(D^2)$, being D the image dimension, RSR has significantly faster performances than those of path-wise Retinex implementations.

To prove this assertion, let us denote with N the number of paths for Retinex or sprays for RSR and with n the mean number of pixels traveled by the paths of Retinex or composing the spray for RSR.

Since we are interested in comparing the intrinsic operations of Retinex and RSR, we consider the case in which both paths and sprays are pre-computed. The memory required to store paths and sprays is practically the same.

Let us denote with the generic name FLOP (floating point operation) a multiplication, division, or equality test between floating point numbers.

In a path-wise Retinex implementation one has to perform $n - 1$ divisions, $n - 1$ multiplications and $n - 1$ comparisons for each path and each image pixel (the $n - 1$ comparisons are needed to see if the product of ratios overcomes 1 in every step of the path, in order to see if the reset mechanism has to be enabled).

Instead, in RSR, only one division and $n - 1$ equality tests are performed for each pixel and each spray.

So the ratio between the number of floating point operations performed by the two algorithms is:

$$\frac{\text{FLOP(Retinex)}}{\text{FLOP(RSR)}} = \frac{3(n-1)NDFLOP}{nNDFLOP}, \quad (9)$$

since n is, at least, in the order of hundreds, $(n - 1)\text{FLOP} \simeq n\text{FLOP}$ and the whole ratio can be approximated by this new one

$$\frac{\text{FLOP(Retinex)}}{\text{FLOP(RSR)}} \simeq \frac{3n \text{ FLOP}}{n \text{ FLOP}} = 3. \quad (10)$$

If one considers also the threshold mechanism, then the previous ratio becomes 4.

VII. CONCLUSIONS

We presented a new implementation of the Retinex model in which the chromatic information in the image is scanned by 2-dimensional pixel sprays instead of 1-dimensional paths, so the name RSR for ‘Random Spray Retinex’. The passage from paths to pixel sprays is based on a recent mathematical characterization [16] of path-wise Retinex implementations.

While RSR shares the same intrinsic properties of every path-wise Retinex implementation, it has proved to be more suitable to analyze the local Retinex properties, being its parameters easily handled in order to perform spatial investigation.

We have analyzed both quantitatively and qualitatively RSR performances. The analysis regards two groups of parameters, one related to the spatial exploration and the other to the amount of chromatic information considered. To tune these parameters, we proposed an unsupervised method, validated through user panel tests.

Concerning spatiality, the results have shown that the mean chromatic influence between pixels decreases as the inverse of their distance, while the amount of information (number of sprays and pixels per sprays) required for optimal results strongly depends on the image content.

Finally, we proved that RSR significantly reduces the number of operations that path-wise Retinex implementations need to filter an image, thus reducing in a significative way the corresponding filtering time.

ACKNOWLEDGMENT

The authors would like to thank Dr. Carlo Gatta for useful suggestions and interesting discussions about this paper.

REFERENCES

- [1] E. Land, J.J. McCann: “Lightness and Retinex Theory”, *J. Opt. Soc. Am. A*, vol. 61, pp. 1-11, 1971.
- [2] J.J. McCann (chair), A. V. : “Special session on Retinex at 40”, *J. of El. Im.*, vol. 13 (1), pp. 6-145, 2004.
- [3] J.J. McCann, S.P. McKee, T.H. Taylor “Quantitative studies in retinex theory. A comparison between theoretical predictions and observer responses to the ”color mondrian” experiments.”, *Vis. Res.*, vol. 16 (5), pp. 445-58, 1976.
- [4] E. Land: “The Retinex Theory of Color Vision”, *Sc. Am.*, vol. 237 (3), pp. 2-17, 1977.
- [5] J. Frankle, J.J. McCann: “Method and apparatus for lightness imaging”, U.S. Patent 4348336, 1983.
- [6] D. Marini, A. Rizzi: “A computational approach to color adaptation effects”, *Im. and Vis. Comp.*, vol. 13, pp. 1005-1014, 2000.
- [7] T.J. Cooper, F. A. Baqai: “Analysis and extensions of the Frankle-McCann Retinex algorithm”, *J. of El. Im.*, vol. 13 (1), pp. 85-92, 2004.
- [8] B. Funt, F. Ciurea, J.J. McCann: “Retinex in MatLab”, *J. of El. Im.*, vol. 13 (1), pp. 48-57, 2004.
- [9] E. Land: “Recent advances in retinex theory and some implications for cortical computations: Color vision and the natural image”, *Proc. Nat. Acad. Sci. USA*, vol. 80, 1983, pp. 5163-5169.
- [10] D.J. Jobson, Z. Rahman, G.A. Woodell: “A multiscale Retinex for bridging the gap between color images and the human observation of scenes”, *IEEE Trans. on Im. Proc.*, vol. 6 (7), pp. 965-976, 1997.
- [11] D.J. Jobson, Z. Rahman, G.A. Woodell: “Properties and performance of a center/surround retinex”, *IEEE Trans. on Im. Proc.*, vol. 6 (3), pp. 451-462, 1997.
- [12] K.Barnard, B.Funt: “Investigations into multi-scale Retinex” *Colour Imaging: Vision and Technology*, John Wiley and Sons, 1999, pp. 9-17.
- [13] S.O. Huck, C.L. Fales , R.E. Davis, R. Alter-Gartenberg: “Visual communication with Retinex coding”, *Appl. Opt.*, vol. 39 (11), pp. 1711-1730, 2000.
- [14] J.D. Cowan, P.C. Bressloff: “Visual cortex and the Retinex algorithm”, *Proc. SPIE Human Vision and Electronic Imaging VII*, vol. 4662, Eds. B.E. Rogowitz and N.T. Pappas, 2002, pp. 278-285.
- [15] Z. Rahman, D.J. Jobson, G.A. Woodell: “Retinex processing for automatic image enhancement”, *J. of El. Im.*, vol. 13 (1), pp. 100-110, 2004.
- [16] E. Provenzi, L. De Carli, A. Rizzi, D. Marini: “Mathematical definition and analysis of the Retinex algorithm”, *J. Opt. Soc. Am. A*, vol. 22, pp. 2613-2621, 2005.
- [17] O. Creutzfeld, B. Lange-Malecki, K. Wortmann: “Darkness induction, retinex and cooperative mechanisms in vision”, *Exp. Brain Res.*, vol. 67, pp. 270-283, 1987.
- [18] O. Creutzfeld, B. Lange-Malecki, E. Dreyer: “Chromatic induction and brightness contrast: a relativistic color model”, *J. Opt. Soc. Am. A*, vol. 7 (9), pp. 1644-1653, 1990.
- [19] L.M. Hurvich, D. Jameson: “Theory of brightness and color contrast in human vision”, *Vision Res.*, vol. 4, pp. 135-154, 1990.
- [20] A. Rizzi, C. Gatta, D. Marini: “Yaccd: Yet another color constancy database”, *Proc. of El. Im. 2003, IS&T/SPIEs International Symposium, Color Imaging VIII San Jose, California U.S.A.*, vol. 5008, 2003, pp. 24-35.
- [21] A. Rizzi, C. Gatta, D. Marini: “A new algorithm for unsupervised global and local color correction”, *Patt. Rec. Lett.*, vol. 124, pp. 1663-1677, 2003.