From samples to populations in retinex models

Gabriele Gianini

From samples to populations in retinex models

Gabriele Gianini*
Università degli Studi di Milano, Computer Science Department, Crema, Cremona, Italy

Abstract. Some spatial color algorithms, such as Brownian Milano retinex (MI-retinex) and random spray retinex (RSR), are based on sampling. In Brownian MI-retinex, memoryless random walks (MRWs) explore the neighborhood of a pixel and are then used to compute its output. Considering the relative redundancy and inefficiency of MRW exploration, the algorithm RSR replaced the walks by samples of points (the sprays). Recent works point to the fact that a mapping from the sampling formulation to the probabilistic formulation of the corresponding sampling process can offer useful insights into the models, at the same time featuring intrinsically noise-free outputs. The paper continues the development of this concept and shows that the population-based versions of RSR and Brownian MI-retinex can be used to obtain analytical expressions for the outputs of some test images. The comparison of the two analytic expressions from RSR and from Brownian MI-retinex demonstrates not only that the two outputs are, in general, different but also that they depend in a qualitatively different way upon the features of the image. © *2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.26.3.031206]

Keywords: color vision; image enhancement; retinex; sampling.

Paper 170026SSP received Jan. 10, 2017; accepted for publication Mar. 6, 2017; published online Mar. 21, 2017.

1 Introduction

The retinex model by Land and McCann[1,2] provides an abstract model of the mechanism of color sensation, relying on the observation that the color appearance at a point does not depend only on its tristimulus value but rather on the comparison among itself and other pixels. According to the model, separately for each chromatic channel, a pixel receives suitably filtered information about the brightness of other image regions and determines a reference white value, based on which its own brightness is eventually rescaled.

Variants of the algorithm have been developed, characterized by different ways of gathering information from the region surrounding the target pixel (the pixel whose appearance has to be computed). Several models use a process based on repeated sampling. A model—from a family of retinex variants called Milano retinex (MI-retinex)—uses memoryless random walks (MRWs)[3] (this is called hereafter Brownian MI-retinex), whereas a computationally efficient variant, random spray retinex (RSR)[4]—from the same family—uses samples of points, called "sprays." The Brownian MI-retinex determines the white reference level for a target pixel as the average of the collection of maxima, each obtained by an MRW (starting from a random point of the image and stopping the first time it encounters the target); RSR determines the white reference level for a target pixel as the average of the collection of maxima, each obtained by a spray, sampled from the neighborhood of the target, according to a given spatial intensity, decreasing with the distance from the target. The fact that both algorithms are based on "samples of samples" represents a drawback, because it entails the presence of statistical noise. When using the two algorithms, one can limit the noise by increasing the number of samples, but only at the expense of the computational efficiency.

In recent works,[5–7] we argued that the process—consisting in drawing samples from the population of pixels' and then computing averages of intensity extrema—can be replaced by the direct computation of suitable quantities out of the whole population of the pixel intensities in the image. This gave rise to two algorithms: ReMark,[6] based on Brownian MI-retinex and RSR-P,[7] based on RSR. Those algorithms compute the reference white for a target as the weighted sum of the intensities of all the image pixels: the weight represents the probability that the contributing pixel becomes the maximum, respectively, of an MRW ending at the target (in ReMark) and of a spray (in RSR-P). In ReMark, this formalization entails representing the process as an absorbing Markov chain and computing its absorption probabilities; in RSR-P, this, more simply, involves computing sampling distributions of the maxima out of the whole pixel population.

In Ref. 7, the same approach used for RSR has been applied to the visual contrast enhancement algorithm STRESS;[8] in the original algorithm formulation, both the maximum and minimum of a spray are used to rescale the input intensity value of a pixel, then the rescaled values from all the sprays are averaged; the corresponding population-based algorithm, called STRESS-P, computes the output as a weighted sum over all the pixel pairs in the image, the weight of a pair is the probability that it represents a minimum–maximum pair for a spray.

Hereafter, we refer to those algorithms that prescribe to enact the repeated sampling as "sampling-based models," and to those that compute the output based on the whole population of all the possible samples as "population-based models." Population-based approaches have the advantage of yielding outputs that, by construction, are free from statistical noise; more importantly, they provide further insight into the respective sampling-based models. Indeed, sampling- and population-based models are complementary to one another

*Address all correspondence to: Gabriele Gianini, E-mail: gabriele.gianini@unimi.it

(in a sense, as a tree-view and a forest-view of a wooded area). Sampling-based models allow researchers to clearly specify fundamental mechanisms (furthermore they are generally easier to implement); population-based models are appropriate for reasoning about model comparisons and lend themselves well to the design and tuning of model approximations.[9]

In Ref. [7], the latter capability has been demonstrated by the design of efficient approximations of RSR-P and STRESS-P based on space quantization, whereas the former has been demonstrated discussing the halo artifact formation mechanisms in RSR-P and in an earlier population-based algorithm, QBRIX.[5]

In the present paper, we review the above-mentioned contributions and further develop the relationships among models. Specifically, we probe the similarities and differences of Brownian MI-retinex and RSR by building an analytical expression for the ReMark and RSR-P output from a specific test image. We observe that the choice of a radially symmetric profile from which to draw the points cannot lead to a faithful substitute for the random walks, for at least two reasons.

- The individual random walk starting from a chosen point and ending at the target has a shape that is typically not radially symmetric around the target; thus using a spray in place of a (two fixed ends) random walk implies exploring the space according to different geometries.

- The random walk in two-dimensional (2-D) cannot take "jumps" from a point to another disconnected point, whereas points in a spray can fall in any position of the image (although with different probability). This implies that the spatial comparisons follow different rules. A random walk cannot compare two regions characterized by different intensities if they are disconnected by an intervening region, without comparing them also with the intervening region. This, in general, changes the distribution of the spray maxima, eventually harmonically averaged to yield the white reference level.

Those two elements can in general produce differences between the processing of Brownian MI-retinex and of RSR, both in their sampling-based version and their population-based version. Although those differences can go unnoticed in the case of most natural images, on specific test images the computation by the two algorithms yields significantly different corrections.

In the present work, we develop this point using the population-based formalization, which can yield to analytic expressions for the output, in specially simple cases. After introducing RSR and RSR-P, we use Brownian MI-retinex and RSR-P to compute analytically the value at a point of a specific test image and highlight the differences. The comparison of the two analytic expressions shows not only that the two outputs are, in general, different, but also that they depend in a qualitatively different way from the features of the image. This result points to the fact that RSR, more than an efficient approximation of the Brownian MI-retinex, should be more soundly considered as an algorithm on its own.

A couple of clarifications are in order, about the scope of the present paper. Retinex is usually used in two ways: as a computational model of human color sensation or as an

unsupervised image enhancer. For the first purpose, a careful procedure of input and output calibration is necessary, whereas for the second task, those phases are not required. In the present work, we consider the use of retinex for image enhancement; therefore, we disregard the issue of the calibrations. Furthermore, the present work does not cover the comparison with the related retinex models based on variational methods such as Refs. [10] and [11].

The paper is structured as follows: first, we recall the definition of Brownian MI-retinex and of ReMark; then, the definition of RSR and RSR-P; finally, we compute analytically their output for the central pixel of a specific test image and show how the two definitions imply qualitatively relevant differences in the output.

2 Memoryless Random Walks Samples and Populations

Here, we describe the MRW sampling-based algorithm Brownian MI-retinex and its corresponding population-based version ReMark.

The original retinex algorithm was defined as consisting in three components: the chromatic channel separation, the reset mechanism, and the thresholding mechanism. The first component is motivated by the observation that the human visual system (HVS) operates through three independent retinal-cortical subsystems, processing the low, middle, and high frequencies of the visible spectrum, respectively; algorithmically, this feature translates into the prescription that each chromatic channel has to be processed independently of the others. The reset and thresholding mechanisms are motivated by the observation that color sensation, and consequently color appearance, is not based on the color stimulus at the point, but rather on the relative spatial arrangement of the stimuli in the observed scene;[12] both intensities and gradients take part into the determination of the output. Each retinex system attenuates smooth changes in intensity—arising for instance from gradients of the illuminant—and enhances sharp changes, i.e., the edges; furthermore, each retinex system estimates the appearance of a point by relating the value at the point to a "local reference white" found within the image and rescales all the intensities of the region with respect to that reference. Algorithmically, in the original path-based formulation of retinex, these observations were rendered by a "path-level" mechanism, called "reset," which would change the current reference white whenever the path exploring the image met a new maximum, and by a "step-level" mechanism, called "thresholding," that would discount small intensity gradients found along the path.

In addition to those well established elements, the early retinex formulations contain components that have been often reinterpreted and given different algorithmic formalizations. The process proposed in the early retinex formulations consisted in a pixel (or region) sampling guided by a path whose deterministic or random character varies from an implementation to another. Some relevant variants defined by means of path-based sampling are Brownian MI-retinex,[3] termite-retinex (TR),[13] energy-driven TR,[14] and the algorithm by Montagna and Finlayson.[15]

In Brownian MI-retinex,[3] paths are defined as MRWs starting from a randomly chosen pixel and stopping the first time the path meets the target pixel.

In addition to the formalizations of the paths, other defining elements contribute to the diversity of retinex variants; the combination of mechanisms is also open to different choices. For instance in Brownian MI-retinex, the quantity computed along the path is used to update the target pixel only; on the contrary, in the classical version by Land and McCann[1,2] and in the version by Montagna and Finlayson,[15] each pixel along the path is updated with the information collected by the path up to that point.

2.1 Brownian MI-retinex

Let $i^{(1)}$, $i^{(2)}$, $i^{(3)}$ be the three chromatic channels of the input color image $\bar{i}$ with size $(n_{row} \times n_{col})$, where $n_{row}, n_{col} \in \mathbb{N}$ represent the number of rows and columns, respectively. Each channel $i^{(c)}$ can be seen as a function $i^{(c)}: \mathfrak{I} \to [0,1]$, where $\mathfrak{I}$ is the image support, i.e., the set of spatial coordinates (u,v) of image pixels, with $u \in \{1, \ldots, n_{row}\}$ and $v \in \{1, \ldots, n_{col}\}$. Since the retinex algorithms process the three channels independently, we drop the apex c and from now on denote the channel function simply by i, and the value of the function at a pixel of index p by $i_p \equiv i(p)$. The (single-channel) input image is thus given by the pair $(\mathfrak{I}, i)$.

We also denote the output channel function by $o: \mathfrak{I} \to [0,1]$ and its value at p by $o_p \equiv o(p)$; the (single-channel) output image is given by the pair $(\mathfrak{I}, o)$. We denote the reference white level of a pixel p by $w_p \equiv w(p)$ and the overall reference white level of a channel by $w: \mathfrak{I} \to [0,1]$; the (single-channel) reference white image is given by the pair $(\mathfrak{I}, w)$.

All the algorithms described here work by computing the output of a pixel at time, this pixel is called "target" and indexed by τ; they compute, by various means, the reference white level w_τ and then compute the output as $o_\tau = i_\tau/w_\tau$.

In Brownian MI-retinex, the image exploration process is based on repeated statistical sampling; each random walk, connecting the random starting point to the target, provides a sample of pixels; Brownian MI-retinex prescribes to compute a synthesis quantity out of each path and eventually to take an average of those synthesis quantities to determine the output value at the target pixel. The synthesis quantity from each sample is computed (in each chromatic channel) by applying the reset and the thresholding mechanism to the products of the ratios of intensities of subsequent pixels along the path: the thresholding consists in forcing the ratio to one when it is within some small threshold from the unit and the reset consists in forcing the product of the ratios to one whenever it exceeds one (for a formal account see Ref. 6).

More formally, consider a "target pixel" $\tau \in \mathfrak{I}$ and a distinct pixel $p \in \mathfrak{I}$ which we call the "initial reference pixel." A path γ connecting p to τ defines an $(n+1)$ steps-long list of pixels γ_t, where t indicates the path-step number

$$\gamma \equiv (p = \gamma_0, \gamma_1, \ldots, \gamma_t, \ldots, \gamma_n = \tau)$$

and the corresponding list of pixel intensities, with $i_t = i(\gamma_t)$. Now indicated by $r(t)$, the ratio of the intensities i_t and i_{t-1} of two consecutive pixels γ_t and γ_{t-1} along the path, for $t > 0$, i.e., $r_t \equiv i_t/i_{t-1}$. By convention, $r(0) \equiv 1$. If the pixel at step t is darker than the one at $(t-1)$, this ratio is less than 1.

The "threshold mechanism" is introduced by means of a "threshold filter" $\theta^\varepsilon(.)$, defined for an argument $z \in \mathbb{R}$ as

$$\theta^\varepsilon(z) \equiv \begin{cases} 1 & \text{if } (1-\varepsilon) < z < (1+\varepsilon) \\ z & \text{otherwise} \end{cases}, \tag{1}$$

where $\varepsilon > 0$ is a parameter of the model (its purpose is to represent a level of contrast insensitivity of the HVS). The threshold function removes smoother intensity changes but maintains sharper intensity changes, e.g., the edges in the image.

The "reset mechanism" is introduced through a function $\rho(z)$, which imposes a "cap" level 1

$$\rho(z) \equiv \begin{cases} z & \text{if } s < 1 \\ 1 & \text{otherwise} \end{cases}. \tag{2}$$

Over the path, one can define of a "recursive path-processing function" F_t of the path-ordered intensities, which, at every step, applies the threshold filter to the current intensity ratio, computes the new product $\theta^\varepsilon(r_t) \times F_{t-1}$ and then applies the reset filter to the chain product of ratios; such a function is defined as follows:

$$F_t^\varepsilon = \begin{cases} 1 & \text{for } t = 0 \\ \rho[\theta^\varepsilon(r_t) \times F_{t-1}^\varepsilon] & \text{for } 1 \leq t \leq n \end{cases}. \tag{3}$$

Due to the reset mechanism, it is granted that $F_t^\varepsilon \in [0,1]$. The contribution $c(\gamma)$ of a path $\gamma_{p\tau}$ of n steps to the output for a target is defined as $c(\gamma) = F_n^\varepsilon$. The output for a pixel is computed by averaging this quantity over a large number of random walks, starting from a large number of points sampled all over the image.

Sometimes—since it has been observed[3] that the suppression of gradients has a minor impact on the output from most natural images—the Brownian MI-retinex model is implemented without thresholding (we called this variant reset-only Brownian MI-retinex). In that case, the computation of the synthesis quantity out of a path can be shown to reduce to the computation of the maximum intensity of the set of pixels visited by the path.[3] This quantity is averaged over all the paths and used as a new reference white level for the target, with respect to which the input is rescaled. Hereafter, we refer to the reset-only version of Brownian MI-retinex.

2.1.1 Reset-only Brownian MI-retinex

It is worth pointing out that applying the reset mechanism is equivalent to enforcing a memory-loss to the path every time it finds a new maximum: in the reset-only version, the value contributed to the target for the computation of the reference white level equals the intensity of the last reset point. Indeed, the reset-only (i.e., $\varepsilon = 0$) recursive function $F_t \equiv F_t^{\varepsilon=0}$, for $t > 0$, takes the form

$$F_t = \rho(r_t \times F_{t-1}). \tag{4}$$

Notice that in a path $\gamma_{p\tau}$ along which no reset takes place, that expression implies a contribution

$$c(\gamma_{p\tau}) = F_n = r_n \times F_{n-1}$$

$$= \prod_{t=1}^{n} r_t = \prod_{t=1}^{n} \frac{i_t}{i_{t-1}} = \frac{i_n}{i_0} = \frac{i_\tau}{i_P} \qquad (5)$$

Furthermore, notice that in a segment of a path between the steps $t_P > 0$ and t_Q along which no reset takes place, the contribution cumulated is

$$\prod_{t=1}^{n} r_t = \prod_{t=t_P+1}^{t_Q} \frac{i_t}{i_{t-1}} = \frac{i_{t_Q}}{i_{t_P}}.$$

In other words, in absence of reset, the contribution is always the ratio between the first and the last intensity, which is always less than or equal to 1:;for the reset to be triggered—and the contribution to be reset to one—the ratio has to become greater than 1, which means that i_{t_P} is no longer the maximum of the segment, but a new maximum has been found. Hence, in the reset-only version of retinex, the reset point is the point of the discovery of a new maximum by the path.

Let us consider a path with at least a reset, in a path γ from p to τ, along which the last reset takes place at step $t = R$, the above results imply a contribution

$$c(\gamma_{p\tau}) = 1 \times \prod_{t=R+1}^{n} r_t = \prod_{t=R+1}^{n} \frac{i_t}{i_{t-1}} = \frac{i_n}{i_R} = \frac{i_\tau}{i_R}. \qquad (6)$$

If one thinks of the path without reset of Eq. (5) as of a path where the reset has been forced at the zeroth step, then the output for pixel τ, computed over a collection of N random paths ending at τ can be written as

$$o(\tau) \equiv \frac{1}{N} \sum_{\gamma} c(\gamma) = \frac{1}{N} \sum_{\gamma} \frac{i_\tau}{i_R}, \qquad (7)$$

$$= i_\tau \left(\frac{1}{N} \sum_{\gamma} \frac{1}{i_R} \right), \qquad (8)$$

where i_R is the intensity of the latest reset point of the path, i. e., the latest intensity maximum, i.e., the absolute path intensity maximum. In short, the output is computed by multiplying the input for the harmonic average of the path intensity maxima. That harmonic average takes the role of reciprocal of the white reference level w_τ for the target.

Overall, the reset-only Brownian MI-retinex algorithm works as follows:

1. Every pixel of the image is taken in turn as the target pixel. The image is scanned systematically and every pixel τ of the image becomes, at a given time, a target.
2. Given the target pixel τ, one chooses at random (usually uniformly at random) a starting point p (the initial reference pixel) for a path.
3. Given the reference–target pair (p, τ), one chooses at random, according to a specific sampling procedure, a path $\gamma \in \Gamma_{p\tau}$, i.e., member of the family of paths connecting p to τ. The procedure consists in starting an MRW from p and stopping the walk the first time it meets τ.

4. Given the path γ, one gets the list of the "input" intensities $i_t = i(\gamma_t)$ for $t \in \{0, 1, \ldots, n\}$.
5. The contribution to the output by a reference pixel p to a target pixel τ "through the walk" γ is defined by the maximum intensity $i_\gamma^{\max} = \max i(\gamma_t)$
6. The harmonic average of the paths' intensity maxima is taken

$$\frac{1}{w_\tau} = \left(\frac{1}{N} \sum_{\gamma} \frac{1}{i_\gamma^{\max}} \right).$$

7. The output is computed as $o_\tau^{\text{MIRretinex}} = i_\tau / w_\tau$.

2.2 ReMark

To provide a population-based translation of the reset-only Brownian MI-retinex process, one has to weight the intensity of each pixel by the probability that it becomes the maximum of a reference-to-target path, then take the result as the white reference level, to compute the output. In Ref. 6, the computation of the relevant probability is accomplished by modeling the joint process—consisting in MRW over the image and in the path functional computation—as a suitable Markov chain.

That chain allows to consider, at the same time, two levels of description: the description of the walk displacement in the 2-D geometric space of the image and the evolution of the information carried by the walk (the maximum intensity found up to the current step), which is represented as a third dimension, as illustrated in Fig. 1. Notice that each reset point of the walk (a point where the path has found a new maximum of intensity) corresponds to a level jump in the three-dimensional (3-D) representation: those points are seen as sinks of the absorbing Markov chain by a level and as sources by another. Notice also that the target is represented by different points, the distinct "target representatives" are situated along the same vertical, if the path meets one of those points, it stops and contributes to the target the value of the intensity level on which it is walking.

By using the formalism of the absorbing Markov chains, one can compute for every source point situated on its 2-D coordinate and at its level of intensity, which is the probability that a random walk starting from there will be absorbed in correspondence of a representative of the target. In practice, one aggregates the sources from the same starting level, the 3-D absorbing Markov chain provides the fraction of the population of walkers absorbed at each representative point, that fraction is the sought weight. Weighting the reciprocal of the absorption levels by that weight, one obtains the desired reference white value for the target.

Formally, the Markov chain is defined by the tuple $(\mathbb{S}, T, P^{(0)})$, where $\mathbb{S}$ is the state space, T is the transition matrix, and $P^{(0)}$ is the initial probability distribution of the walkers. As the discrete time steps t proceed, the distribution of probability mass $P^{(t)}$ evolves; asymptotically for $t \to \infty$ in this Markov chain, on all the states, the mass will be zero, except for those absorbing states that are associated to the target τ; since there may be one such state for every level i_k, we indicate the array of the corresponding probabilities by $P_k^{(\infty)}(\tau)$, for $k = 1, \ldots, M$. The quantities $P_k^{(\infty)}(\tau)$ are the weights used by the algorithm to perform the weighted harmonic average that yields the whiter reference level w_τ.
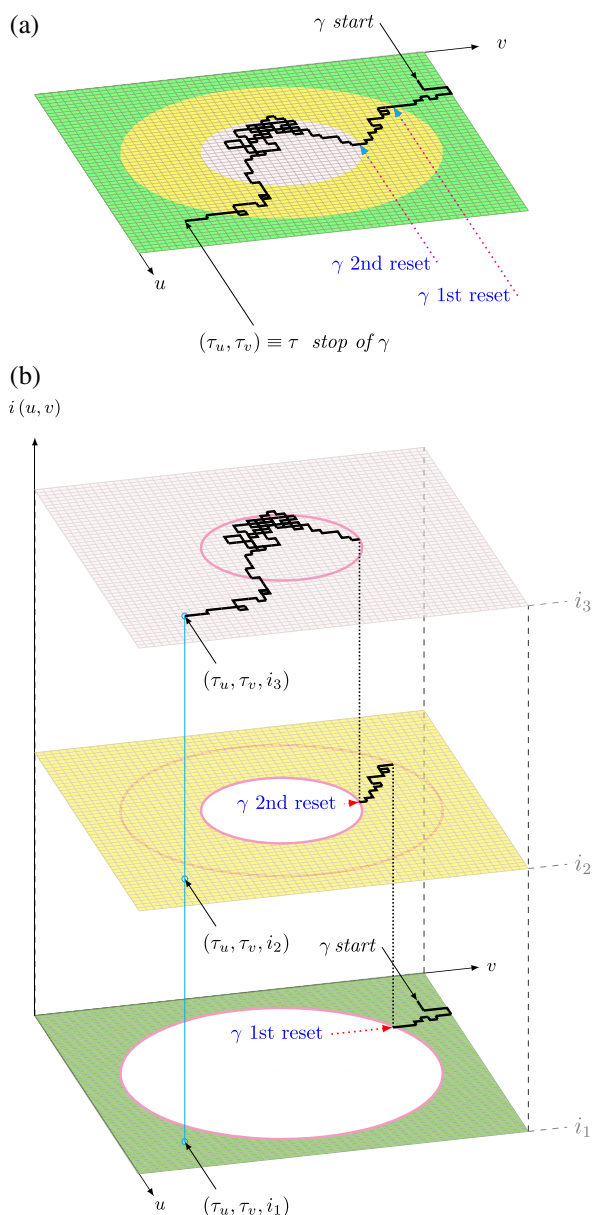
Fig. 1 An example random walk from the 2-D space representation (a) to the 3-D Markov chain representation (b). In the reset-only Brownian MI-retinex, the 2-D random walk carries the information about the latest intensity maximum found in exploring the image. Every time a new maximum is found, the carried information is updated: the point of the image where the update takes place is called reset point. This process can be modeled as a 3-D Markov process: two dimensions are used to model the spatial diffusion, the third dimension to model the carried value: reset points correspond to upward transitions in the 3-D representation. Transitions can take place only isotropically among first neighbors of the same level and—at reset points—unidirectionally, from a coordinate of a level to a coordinate of an upper level. The target pixel τ is pictured by a vertical array of target representatives: each representative absorbs and stops the walk. The fraction of walkers absorbed by target representative located at a level i corresponds to the probability that that level of intensity contributes to the target.

The whole absorbing Markov chain, described hereafter, is a device aimed at computing $P_k^{(\infty)}(\tau)$.

State space $\mathbb{S}$. The detailed structure of the Markov chain depends on the input image $(\mathfrak{I}, i)$, Consider the codomain of $i(\cdot)$ and let $(i_k)_k^M$ the list of the distinct values of intensity

present in the image, listed in increasing order ($j > k \Leftrightarrow i_j > i_k$). One can partition the set $\mathfrak{I}$ of pixels into subsets $\tilde{\mathfrak{I}}_k \equiv \{x \in \mathfrak{I}: i(x) = i_k\}$, we call levels. The 3-D state space $\mathbb{S}$ consists of M subspaces of dimension 2, as many as the distinct intensity levels in the image

$$\mathbb{S} \equiv \bigcup_k^M S_k.$$

Each subspace S_k is defined as follows:

$$S_k \equiv \bigcup_{j=1}^k \mathfrak{I}_j, \tag{9}$$

i.e., the k'th subspace encompasses all the coordinates of those pixels having intensity lower or equal to the k'th level intensity. Each state $s \in \mathbb{S}$ is characterized by the coordinates: $s \in (x_u, x_v, i_k)$. Thus, the number of states in $\mathbb{S}$ is of the order of $N \times M$, where N is the number of pixels in the original image and M is the number of distinct levels.

Transition matrix T. For the, rather lengthy, formal description of the transition matrix T, we refer to Ref. 6; for the present discussion, it is sufficient to rely on the illustration provided by Fig. 1 and its caption.

Initial distribution $P^{(0)}$. The initial probability distribution depends on the source character of the states in $\mathbb{S}$. A state (x_u, x_v, i_j) is a source for a subspace S_j if and only if the corresponding pixel has intensity i_j, i.e., $x \in \mathfrak{I}_j$, it follows (see also Fig. 1) that in the lowest subspace all the states are sources, whereas in the other subspaces only a part of the states are sources. The actual amount of initial probability "mass" at a source, i.e., $P_j^{(0)}(s) = P^{(0)}(x_u, x_v, i_j)$, depends on the sampling scheme $U(x)$ [typically $U(x)$ is uniform over the image]

$$P_j^{(0)}(s) = P^{(0)}(x_u, x_v, i_j) = \begin{cases} U(x) & \text{when } i(x) = i_j \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

This initial probability distribution of the population of random walkers is made to evolve by the transition matrix T.

2.2.1 Algorithm

Given an input image $(\mathfrak{I}, i)$, the algorithm has to

1. identify the distinct intensity levels present in the image and create a list of M (unique) values of intensity $(i_1, \ldots, i_k, \ldots, i_M)$,

2. identify the set of geometric coordinates corresponding to each intensity value, the k'th intensity level is the set $\mathfrak{I}_k$ of geometric positions defined as $\mathfrak{I}_k \equiv \{x \in \mathfrak{I}: i(x) = i_k\}$,

3. for each level, $\mathfrak{I}_k$ creates a 2-D subspace $S_k = \bigcup_{j=1}^k \mathfrak{I}_j \subseteq \mathfrak{I}$,

4. for each subspace, S_k determines the absorbing states (reset points and target representatives) and build the transition matrix T_k,

5. for each subspace, S_k initializes the statistical weights $P_k^{(0)}$ on that level's sources to $U(x)$,

6. for each subspace, S_k starting from $k = 1$ up to $M - 1$:

    i. pass form the original basis to the canonical basis of the absorbing Markov Chain (e.g., absorbing states first),

    ii. find the canonical form $\hat{T}_k$ of T_k,

    iii. extract from $\hat{T}_k$ the transient-to-transient matrix $\mathcal{Q}_k$ and the transient-to-absorbing matrix $\mathcal{P}_k$,

    iv. $\mathcal{S}_k = (I - \mathcal{Q}_k)^{-1}$ and find the array $\hat{m}_k \equiv P_k^{(0)} \cdot (\mathcal{S}_k \mathcal{P}_k)$ of the "absorbed mass for states of level" k,

    v. return to the original state basis obtaining, from $\hat{m}_k$, the corresponding m_k,

    vi. for the "target absorbing state" set $P_k^{(\infty)}(\tau) = m_k(x)$,

    vii. transfer the mass absorbed across the "nontarget absorbing states" up to the level i_x, for which the pixel x is a source

    $$P^{(0)}(x_u, x_v, i_x) = P^{(0)}(x_u, x_v, i_x) + m_k(x),$$

7. for the top subspace, S_k with $k = M$, attribute all the mass to the target absorbing state, i.e., set $P_M^{(\infty)}(\tau) = \sum_{x \in \Im} P^{(0)}(x_u, x_v, i_M)$,

8. the output of the loop (step 6) and the step 7 over subspaces is the array of the weights $P_k^{(\infty)}(\tau)$ (with $k = 1, \ldots, M$),

9. compute the weighted sum of the contributions from the same level of the target k up to M

    $$\frac{1}{w(\tau)} = \sum_{j=k}^{M} \frac{1}{i_j} P_k^{(\infty)}(\tau), \tag{11}$$

10. get the retinex output as

    $$o^{\text{ReMark}}(\tau) = \frac{i(\tau)}{w(\tau)}. \tag{12}$$

The core computation in the algorithm consists in the inversion of a large matrix, the so-called fundamental matrix of the absorbing Markov chain, derived from the transition matrix. Since the matrix is sparse (each state is connected only to few neighbors), one can manage to process even large images. Still, the computation of this exact mapping of Brownian MI-retinex is rather expensive. Nonetheless, if one aims at producing noise-free filtered images, the algorithm, called ReMark, turns out to be more efficient than the strictly MRW of Brownian MI-retinex[3] (further details in Ref. 6).

3 Spray Samples and Populations

The RSR[4] algorithm builds on the modeling features formalized by the reset-only Brownian MI-retinex and achieves a higher efficiency, by trying to retain most of the qualitatively desirable features of the path-based algorithm. Its corresponding population-based version is called RSR-P. The key observation of RSR is that one could replace the expensive

process of random walk generation by a more efficient one. The Brownian MI-retinex process is inefficient in two respects:

- the local search of the target can take a very long time and

- while intermediate pixels can be visited several times, they do not bring extra information about the path maxima.

Provenzi et al.[4] propose to substitute each random path by a set of points—called "spray"—generated so as to loosely mimic the path sampling, they choose to define the spray as a set of points sampled from a "radially symmetric" probability distribution, called "sampling profile" (formally, the process corresponds to an isotropic nonhomogenous Poisson point-sampling process with intensity decreasing as the distance from the target increases), see Fig. 2. In RSR, the locality of the filtering is controlled by the speed of decrease of the sampling profile as a function of the distance from the target (for a recent variant of RSR, see also Ref. 16).

3.1 Random Spray Retinex

Given the input image, for each chromatic channel and for each target pixel τ, the algorithm proceeds as follows:
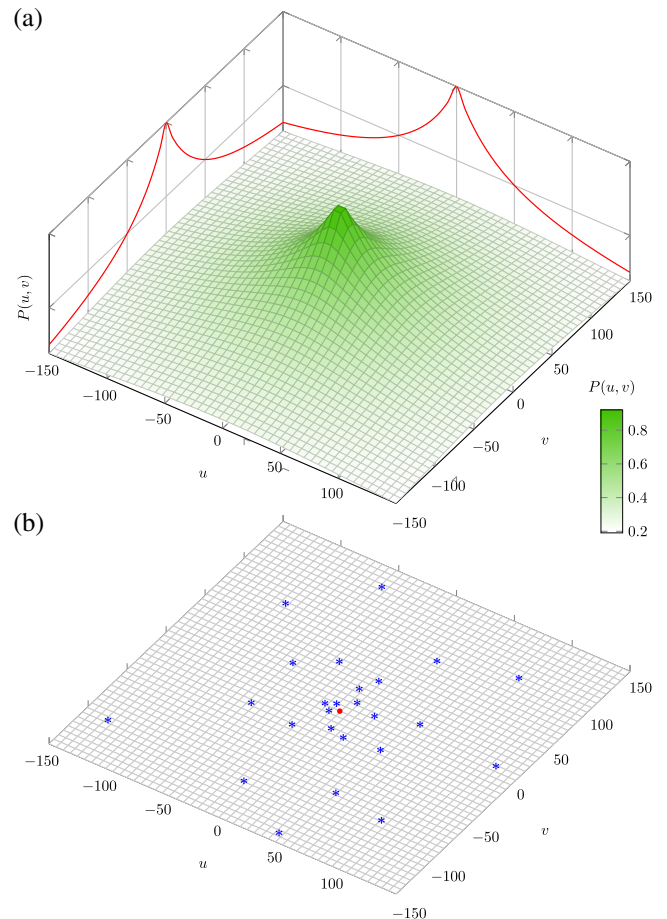

Fig. 2 Schematic examples of (a) a sampling profile and of (b) spray. In (b), the (blue) asterisks represent a bare spray, the (red) bullet in (0, 0) represents the target position; altogether the markers represent an augmented spray.

a. Repeat N times the spray generation and processing cycle (we use the index s to refer to a spray).

1. Sample n points from a neighborhood Ω_τ of the target according to a radially symmetric sampling profile (see for example Fig. 2), obtaining an n-point set.

2. Get the corresponding sample of n "intensities" $\iota_s^* \equiv \{i_k\}_{k=1}^n$. We refer to this sample as "bare spray."

3. Add the target τ to the point set and obtain an $(n+1)$-point intensity set $\iota_s \equiv \{\{i_\tau\} \cup \iota_s^*\}$, the "augmented spray."

4. Compute the maximum intensity y_s of the augmented spray ι_s, i.e., $y_s \equiv \max\{i_\tau, \max_k\{i_k\}_{k=1}^n\}$.

b. After repeating N times, the steps 1 through 4 of point a, one obtains a set of maxima $\{y_1, y_2, \ldots, y_s, \ldots, y_N\}$. Compute the harmonic average of the maxima

$$\frac{1}{w_\tau} \equiv \frac{1}{N}\sum_{s=1}^N \frac{1}{y_s}. \tag{13}$$

The quantity w_τ plays the role of "white reference value" for τ.

c. Set the output value o_τ^{RSR} for the target pixel, rescaling the input i_τ by the white reference value

$$o_\tau^{\text{RSR}} = \frac{i_\tau}{w_\tau}. \tag{14}$$

The sampling profile—a radially symmetric spatial probability density—can be assigned by providing a radial function, $\lambda(r)$, typically a decreasing function of the argument r, a standardized distance defined as $r(\tau, p) = d(\tau, p)/D$, i.e., as the ratio between the Euclidean distance $d(\tau, p)$ of a point p from the target τ and the diagonal of the image, $D = \max_{p', p'' \in \Im} d(p', p'')$.

3.2 Random Spray Retinex-P

Given a target pixel, in principle the algorithm RSR-P computes, for each pixel of the image, the probability that it becomes the maximum of a spray that includes the target. In practice, the algorithm proceeds by grouping first the pixels according to their intensity levels, then by computing the intensity distribution of the bare spray maxima and finally the distribution of the augmented spray maxima. For convenience, the computation passes through the cumulatives of the distributions. The detailed motivations of the steps of this procedure can be found in Ref. 7.

A key element of the computation is the probability of drawing a point of the sample from a specific level. This can be obtained as follows. Given a sampling profile in terms of $\lambda(r)$ and a target τ, the probability that a specific pixel p is drawn at random can be computed through unnormalized weight $\Lambda_\tau(p)$ of $p : \Lambda_\tau(p) = \lambda[r(\tau, p)]/2\pi r(\tau, p)$; the normalization value is $\Lambda_\tau^{\text{tot}} = \sum_p \Lambda_\tau(p)$ and the probability $P(p)$ of a pixel p is $P(p) = \Lambda_\tau(p)/\Lambda_\tau^{\text{tot}}$. The probability of an intensity level i can be obtained as follows: the unnormalized weights $\Lambda_\tau(p)$ can be cumulated into a

histogram, where each intensity level corresponds to a channel; the total unnormalized weight of the intensity i is then

$$h_\tau(i) \equiv \sum_{\{p \in \{1, \ldots, n_{\text{pix}}\} | i_p = i\}} \Lambda(p). \tag{15}$$

Normalization yields the sought probability distribution

$$f_\tau(i) = h_\tau(i)/\sum_j h_\tau(j). \tag{16}$$

3.2.1 Algorithm

For each individual target pixel τ, the computation has to undergo the following phases, passing through the computation, for each level i, of the probability $\beta_\tau(i)$ that the level is the maximum of an n-point spray, then of the probability $\alpha_\tau(i)$ that the level is the maximum of an augmented spray centered in τ.

1. Computation of the reference pixels' weight $\Lambda_\tau(p)$ and histogram of the intensities h_τ.
   Let D be the size of the image diagonal. Set $h_\tau(i) = 0 \, \forall i$, then for each pixel $p \in \Im$:

   • compute its Euclidean distance $d(\tau, p)$ from the target,

   • compute the standardized distance $r(\tau, p) = d(\tau, p)/D$,

   • set $\Lambda_\tau(p) = \lambda[r(\tau, p)]/2\pi r(\tau, p)$,

   • set $h_\tau(i = i_p) = h_\tau(i = i_p) + \Lambda_\tau(p)$.

2. Computation of distribution $f_\tau(i)$ and cumulative $F_\tau(i)$:

   • compute $H_\tau \equiv \sum_i h(i)$,

   • $\forall i$ compute $f_\tau(i) \equiv h_\tau(i)/H_\tau$,

   • compute the cumulative $F_\tau(i)$ of $f_\tau(i)$.

3. Computation of the bare spray maxima distribution $\beta_\tau(i)$ and cumulative $B_\tau(i)$

   • Compute the cumulative as $B(i) = [F(i)]^n$.

   • Compute the distribution for each intensity $i > i_\tau$ as

   $$\beta_\tau(i) = \beta_\tau(i_l) = [F_\tau(i_l)]^n - [F_\tau(i_{l-1})]^n. \tag{17}$$

4. Compute the target weight as $\alpha_\tau = B(i_\tau) = [F(i_\tau)]^n$.

5. Compute $1/w_\tau$ as

   $$\frac{1}{w_\tau} = \frac{1}{i_\tau}\alpha_\tau + \sum_{i>i_\tau}\frac{1}{i}\beta_\tau(i). \tag{18}$$

6. Compute the output value $o_\tau^{\text{RSR-P}}$ for the target pixel as

   $$o_\tau^{\text{RSR-P}} = \frac{i_\tau}{w_\tau}. \tag{19}$$

4 Comparison of the Two Models Brownian MI-retinex and Random Spray Retinex

4.1 Maxima of Paths and of Sprays

Consider an input image consisting in three concentric regions C, A, and B each with a distinct intensity levels $i_A = a$, $i_B = b$, and $i_C = c$ as in Fig. 3. Let the level intensities be ordered as follows: $0 < c < b < a$. Notice that in the input image, the lowest intensity—i.e., c—is associated to the central disk C, the highest—i.e., a—is associated to the middle ring A, whereas the intermediate—i.e., b—is associated to the outer region B. Let us indicate by $|A|$, $|B|$, and $|C|$, respectively, the areas of regions A, B, and C, normalized to 1, i.e., such that $|A| + |B| + |C| = 1$. We consider the problem of computing the intensity of the target pixel τ situated at the center of the image, by means of RSR (through RSR-P) and Brownian MI-retinex (through ReMark). We will show that not only the two outputs $o^{\text{MIRetinex}}(\tau)$ and $o^{\text{RSR}}(\tau)$ are, in general, different, but also that they depend in a qualitatively different way from the parameters of the problem. The most remarkable difference is that in MI-retinex the output is independent of the actual value of the level b (provided that, as stipulated, $a > b$).

4.1.1 Brownian MI-retinex output

Indeed, in Brownian MI-retinex, no information about the level of B can reach τ: every path starting from a pixel of B undergoes a reset in region A before meeting τ. Thus, any path from B will eventually carry the value a as a path maximum. Taking into account, the fact that any path starting from A will carry to τ the value a because this is the global maximum of the image (paths from A will undergo no reset), one has from Eqs. (11) and (12)

$$o^{\text{MIRetinex}}(\tau) = i_\tau \frac{1}{w_\tau} = c\left(|A|\frac{1}{a} + |B|\frac{1}{a}\right.$$
$$\left. + |C|\left\{[1 - Q(\tau)]\frac{1}{a} + Q(\tau)\frac{1}{c}\right\}\right)$$

In the term with coefficient $|C|$, which represents the contribution by the region C, the factor $Q(\tau)$ represents the ratio of
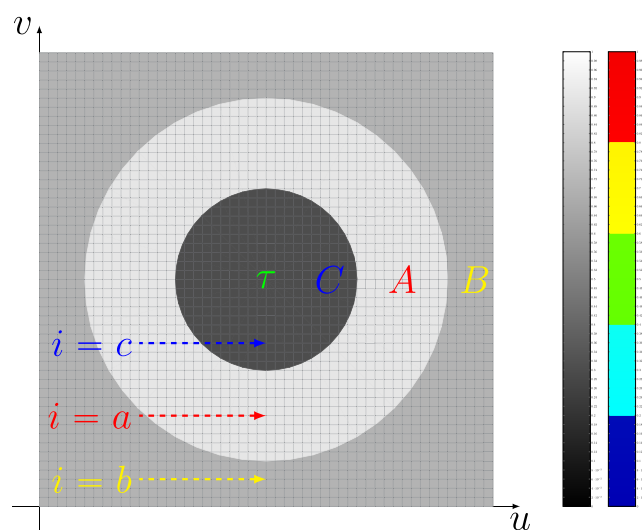

Fig. 3 A test input image for illustrating the qualitative difference between the Brownian MI-retinex and the RSR filtering (see text).

the paths starting from a point within C that meet τ before meeting A (in the former case, it contributes $1/c$ to the target, in the latter, it contributes $1/a$). We postpone the computation of $Q(\tau)$, the salient feature of the above expression is that the value b "does not appear in the computation."

4.1.2 Random spray retinex output

In RSR, the probability of a contribution from B depends on the nonzero probability that in a spray of n points (centered on the target τ) no point touches the region A and at least one point touches region B, thus level b can contribute to the output.

Let us compute the probabilistic weights $\alpha(a)$, $\alpha(b)$, and $\alpha(c)$ (from where we dropped, for the sake of clarity, the index of the target) by which the regions A, B, and C contribute the quantities $1/a$, $1/b$, and $1/c$ to the factor $1/w_\tau$.

Notice first that the bare spray and augmented spray have the same maximum, since the target is at the absolute minimum of the image, thus $\alpha(i) = \beta(i) \forall i$.

Consider, furthermore, the probabilities $f(a)$, $f(b)$, and $f(c)$, that an individual point drawn at random using the radial sampling profile λ, falls in the areas A, B, and C, respectively. Recall the cumulative function $F(c) = f(c)$, $F(b) = f(b) + f(c)$, and $F(a) = f(a) + f(b) + f(c) = 1$. Recall Eq. (17) for the probability of a maximum at an intensity i for a set of n points drawn independently from f. The sought probabilistic weights are

$$\alpha(a) = \beta(a) = F^n(a) - F^n(b) = 1 - [1 - f(a)]^n \quad (20)$$

$$\alpha(b) = \beta(b) = F^n(b) - F^n(c) = [1 - f(a)]^n - f^n(c) \quad (21)$$

$$\alpha(c) = \beta(c) = F^n(c) = f^n(c). \quad (22)$$

And finally

$$o^{\text{RSR}}(\tau) = i_\tau \frac{1}{w_\tau} = c\left[\frac{1}{a}\beta(a) + \frac{1}{b}\beta(b) + \frac{1}{c}\beta(c)\right].$$

4.1.3 Comparisons

We summarize the most relevant differences between Brownian MI-retinex and RSR, emerging from the comparison of the above analytic results.

Segregation. The most relevant observation about the above expression for $o^{\text{RSR}}(\tau)$ is that b does participate into the determination of $o^{\text{RSR}}(\tau)$, whereas it does not have any explicit role in $o^{\text{MIRetinex}}(\tau)$, that can be rewritten as

$$o^{\text{MIRetinex}}(\tau) = c\left(\frac{1}{a}\{|A| + |B| + |C|[1 - Q(\tau)]\} + \frac{1}{c}Q(\tau)\right).$$

In short, $o^{\text{MIRetinex}}(\tau)$ is independent of b, $o^{\text{RSR}}(\tau)$ depends on b. This finding can be easily generalized, one could easily demonstrate that when an image region is surrounded by a topologically closed stretch A, of chromatic intensity a, according to Brownian MI-retinex, it becomes insensitive to any pixel of the image with intensity less than a.

Independence of Brownian MI-retinex from the geometry of the top level. Let r_C be the radius of the inner disk (region C) and let r_A be the radius of the outer border of

region A. Let us use a typical RSR sampling profile $\lambda(r) \propto 1/(1 + r)^{\eta}$ with $\eta \geq 1$. In RSR, the statistical weight associated to the area A, from Eq. (20), is $\beta(a) = 1 - [1 - f(a)]^{n}$, where the probability f is

$$f(a) \propto \int_{r_C}^{r_A} \frac{2\pi r}{(1 + r)^{\eta}} \, dr.$$

It is clear that, in RSR, the direct contribution of the ring A to the weight of $1/a$, i.e., $\beta(a)$, depends on r_A and r_B.

Let two distinct instances of the test image—whose structure is illustrated Fig. 3—be given, differing from one another in the thickness of the ring A (of top intensity a):

- Let the first image have a ring A thicker and closer to the center, enclosed by the circles of radii r_C' and r_A'.
- Let the second image have a ring A thinner and farther from the center, enclosed by the circles of radii r_C'' and r_A'', e.g., with $r_C'' > r_A'$
- Let, however, the area of the two rings be the same, i.e.,

$$\pi r_A'^2 - \pi r_C'^2 = \pi r_A''^2 - \pi r_C''^2 = |A|.$$

In Brownian MI-retinex, the direct contribution of the ring A to the weight of $1/a$ equals $|A|$, thus it is equal in the two images and independent of the geometry of the area A (by referring to the direct contribution, we mean that we are not considering the contribution to the term $1/a$ coming from B and the fact that A precludes to the blocks from B the access to τ). This contrasts with typical assumptions about the dependence of the sensation on the geometry of the brightest patches.[17]

On the contrary, in RSR, the two different rings will determine two distinct contributions from A: it is clear from the above expression that $f(a)$ depends not directly on the area of the ring region A, but also on its geometry. It is worth pointing out, in passing, that still, in RSR, the geometry of the brightest patches is accounted for only in terms of distribution of the radial distances w.r.t. the target. More sophisticated approaches can be devised to take into account more complex dependencies,[17–19] better accounting for the experimental findings.

In short, Brownian MI-retinex and RSR treat differently the set of pixels with intensity corresponding to the absolute maximum; in Brownian MI-retinex, only its total surface counts, whereas in RSR counts also its geometry.

5 Conclusions

Sampling-based retinex models and population-based retinex models are complementary to one another. Original insights into the modeled phenomenon can be provided by each type of model individually and by their mutual comparison. In this paper, we compared the population-based models of Brownian MI-retinex and RSR by deriving the two corresponding analytic expressions of the output based on a specific test image. The two expressions show that the two outputs depend in a qualitatively different way from the features of the image. With a specific test image, we highlighted two phenomena built-in into Brownian MI-retinex, the segregation of regions surrounded by high chromatic

intensity closed areas, and the insensitivity to the geometry of the top chromatic intensity level. Although RSR was born as an efficient approximation of Brownian MI-retinex, the two mentioned features highlight that the two algorithms behave in qualitatively different ways. Nonetheless, the mentioned effects can go unnoticed, when processing natural images.

References

1. E. H. Land and J. J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Am.* **61**, 1–11 (1971).
2. E. H. Land et al., *The Retinex Theory of Color Vision*, Vol. **237**, No. 6, pp. 108–128, Scientific American, San Francisco, California (1977).
3. E. Provenzi et al., "Mathematical definition and analysis of the retinex algorithm," *J. Opt. Soc. Am. A* **22**(12), 2613–2621 (2005).
4. E. Provenzi et al., "Random spray retinex: a new retinex implementation to investigate the local properties of the model," *IEEE Trans. Image Process.* **16**(1), 162–171 (2007).
5. G. Gianini, A. Manenti, and A. Rizzi, "Qbrix: a quantile-based approach to retinex," *J. Opt. Soc. Am. A* **31**(12), 2663–2673 (2014).
6. G. Gianini, A. Rizzi, and E. Damiani, "A retinex model based on absorbing Markov chains," *Inf. Sci.* **327**, 149–174 (2016).
7. G. Gianini, M. Lecca, and A. Rizzi, "A population-based approach to point-sampling spatial color algorithms," *J. Opt. Soc. Am. A* **33**(12), 2396–2413 (2016).
8. Ø. Kolås, I. Farup, and A. Rizzi, "Spatio-temporal retinex-inspired envelope with stochastic sampling: a framework for spatial color algorithms," *J. Imaging Sci. Technol.* **55**(4), 40503 (2011).
9. G. Gianini, "Statistical aspects of space sampling in retinex models," *Electron. Imaging* **2016**(6), 1–6 (2016).
10. M. Bertalmo, V. Caselles, and E. Provenzi, "Issues about retinex theory and contrast enhancement," *Int. J. Comput. Vision* **83**(1), 101–119 (2009).
11. R. Palma-Amestoy et al., "A perceptually inspired variational framework for color enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(3), 458–474 (2009).
12. J. Albers, *Interaction of Color*, Yale University Press, New Haven (1975).
13. G. Simone et al., "Termite retinex: a new implementation based on a colony of intelligent agents," *J. Electron. Imaging* **23**(1), 013006 (2014).
14. M. Lecca, A. Rizzi, and G. Gianini, "Energy-driven path search for termite retinex," *J. Opt. Soc. Am. A* **33**(1), 31–39 (2016).
15. R. Montagna and G. D. Finlayson, "Constrained pseudo-Brownian motion and its application to image enhancement," *J. Opt. Soc. Am. A* **28**(8), 1677–1688 (2011).
16. M. Lecca and A. Rizzi, "Tuning the locality of filtering with a spatially weighted implementation of random spray retinex," *J. Opt. Soc. Am. A* **32**(10), 1876–1887 (2015).
17. J. McCann and A. Rizzi, *The Art and Science of HDR Imaging*, John Wiley (2011).
18. J. L. Lisani et al., "A generalized white-patch model for fast color cast detection in natural images," *Electron. Imaging* **2016**(6), 1–4 (2016).
19. G. Gianini and A. Rizzi, "A fuzzy set approach to retinex spray sampling," *Multimedia Tools Appl.* (2017) (in press).

Gabriele Gianini received PhD in physics from University of Pavia, Italy, in 1996. Since 2005 he has been an assistant professor at the Computer Science Department of the Università degli Studi di Milano, Italy. He has held visiting positions at a number of international institutions, including INSA de Lyon (France), University of Passau, Germany, EBTIC/Khalifa University, Abu Dhabi (UAE), CERN, Geneva (Switzerland), Fermilab, Chicago (USA), and CBPF (Brazil). His research interests include probabilistic modeling, statistical and soft computing techniques.