Ant Colony for Locality Foraging in Image Enhancement

Gabriele Simone, Davide Gadia, Ivar Farup and Alessandro Rizzi

Abstract This chapter presents a spatial color algorithm called Termite Retinex and the problem of filtering locality in this family of algorithms for image enhancement, inspired by the human vision system. The algorithm we present is a recent implementation of Retinex with a colony of agents, which uses swarm intelligence to explore the image, determining in this way the locality of its filtering. This results in an unsupervised detail enhancement, dynamic range stretching, color correction, and high dynamic range tone rendering. In the chapter we describe the characteristic of glocality (glocal = global + local) of image exploration, and after a description of the Retinex spatial color algorithm family, we present the Termite approach and discuss results.

Keywords Retinex · Spatial color algorithm · Artificial termites · Image enhancement

1 Introduction

The term "image enhancement" refers to a multitude of algorithms and approaches. They differ in the way they realize the enhancement, and as a consequence, different methods can be considered in order to measure their success. For example, one of the subgoals of image enhancement is the adjustment of visual contrast, but no agreed

G. Simone (✉) · D. Gadia · A. Rizzi
Department of Computer Science, Università degli Studi di Milano, Milano, Italy
e-mail: gabriele.simone@hig.no

D. Gadia
e-mail: davide.gadia@unimi.it

A. Rizzi
e-mail: alessandro.rizzi@unimi.it

G. Simone · I. Farup
The Norwegian Colour and Visual Computing Laboratory, Faculty of Computer Science and Media Technology, Gjøvik University College, Gjøvik, Norway
e-mail: ivar.farup@hig.no

S. Dehuri et al. (eds.), *Multi-objective Swarm Intelligence*,
Studies in Computational Intelligence 592, DOI 10.1007/978-3-662-46309-3_5

measure of visual contrast in digital images exists. In particular, an important part in image enhancement is played by the correction of color. Many image enhancement algorithms mimic the Human Visual System (HVS) since it accomplish powerful adjustment automatisms, like lightness constancy and color constancy. Lightness constancy enables a stable perception of the scene regardless of changes in the mean luminance intensity, while color constancy enables a stable perception of the scene regardless of changes in the color of the illuminant [1]. The difference among taking inspiration from HVS or trying to mimic it is relevant [2]. Any model of vision needs a careful calibration of input and output information up to single pixel level in order to be used for the simulation of human vision. To ease these constraints, HVS could be considered just as inspiration for image enhancement, with the goal to meet observers preferences rather than to match observer perceptual response [1, 3, 4]. This is the approach we have followed in the design and implementation of the Termite Retinex (TR) [5] method presented in this chapter, which uses swarm intelligence techniques to tune the locality of Retinex filtering. Termite Retinex belongs to the MI-Retinex [6] family of implementations of the Retinex theory, proposed by Land and McCann [7, 8] as a model of color sensation of HVS.

All these algorithms belong to a family of computational models inspired by the HVS, called Spatial Color Algorithms (SCAs) [2]. The visual evidence at the basis of SCAs is that color sensation does not depend on the point-wise value of the stimulus, but on the spatial arrangement of the stimuli in the scene. Same point-wise values of radiance can originate a completely different visual response according to the image content [9]. This mechanism is also at the basis of several visual illusions [10–12]. To implement this important principle, SCAs perform spatial comparisons among the pixels in the input image. SCAs share a common structure of two phases. In the first phase each pixel is computed according to the spatial distribution of the other pixels in the image (all, or some of them, as prescribed by each implementation). To form the output image, the matrix of values, spatially computed in the first phase, is scaled in the second phase onto the available output dynamic range according to the model goal. SCAs has been proven to be able to obtain high preferences scores in terms of user preferences/pleasantness [3, 4].

In this chapter we present a brief overview of Retinex theory and models in Sect. 2. In Sect. 3 we present Termite Retinex, and in Sect. 4 we analyze the effect of swarm intelligence techniques in the exploitation of image locality. Finally, in Sect. 5 we present the properties of Termite Retinex and afterwards in Sect. 6 conclusions are drawn.

2 Retinex Theory

The term Retinex has been coined by Land and McCann [7, 8, 13] referring to the roles that both retina and cortex play in human vision. Land and McCann report [8]:

> Sensations of color show a strong correlation with reflectance, even though the amount of visible light reaching the eye depends on the product of reflectance and illumination. The visual system must achieve this remarkable result by a scheme that does not measure flux. Such a scheme is described as the basis of retinex theory. This theory assumes that there are three independent cone systems, each starting with a set of receptors peaking, respectively, in the long-, middle-, and short-wavelength regions of the visible spectrum. Each system forms a separate image of the world in terms of lightness that shows a strong correlation with reflectance within its particular band of wavelengths. These images are not mixed, but rather are compared to generate color sensations. The problem then becomes how the lightness of areas in these separate images can be independent of flux.

The term "lightness", for Land and McCann, is associated to the brightness sensation in each channel. Edges among adjacent areas of an image, and lightness ratio between two areas, play a fundamental role in the final appearance at each point.

Formally, Retinex is based on computing the relative channel lightness (L) at a point i as the mean value of the relative channel lightnesses (l) computed along N random paths from point j to the point i (Fig. 1a):

$$\frac{L^i = \sum_{h=1}^{N} l_h^{i,j}}{N} \tag{1}$$

where,

$$l_h^{i,j} = \sum_{x=j}^{i} \delta log\left(\frac{I_{x+1\in path}}{I_{x\in path}}\right), \tag{2}$$

where I_x is the lightness intensity of the pixel x, and $I_x + 1$ is the lightness intensity of the pixel $x + 1$ and h is indicating the path. An example of Random paths is shown in Fig. 1. The reset mechanism δ forces the chain of ratios to restart from the unitary value, considering the lightness value found at the reset point a new local reference white [6]:

$$\delta = \begin{cases} 1 & \text{if } \left| log\left(\frac{I_{x+1\in path}}{I_{x\in path}}\right)\right| > T \\ 0 & \text{if } \left| log\left(\frac{I_{x+1\in path}}{I_{x\in path}}\right)\right| \leq T \end{cases} \tag{3}$$

where T is a defined threshold.

The three color channel R, G, and B are processed independently and thus the lightness is represented by the triplet (L_R, L_G, L_B) of lightness values in the three chromatic channels.

The original formulation of Retinex does not provide a description of how to generate the random paths. This is a critical point: in Retinex, appearance is calculated using ratios, and the ratios are applied to samples in the image. Therefore, changing the method to create the random paths (i.e., their structure), we change the way locality is considered and introduced in the Retinex elaboration.

There is a large literature available on the different approaches applied for the exploration of locality in the implementation of Retinex models. Here we present

a brief overview of the main approaches and implementations, for an exhaustive overview and analysis we suggest the reader to consult [6, 14].

We can identify two different approaches in Retinex implementations: sampling and integrating.

In the sampling approach, Retinex elaboration in each pixel is applied using a subset of samples in its neighborhood, selected along paths starting from the considered pixel. The Retinex implementations based on sampling follow strictly the original model by Land and McCann, and they differ mainly in the method and geometry of the paths (e.g. spiral [15], or double spiral [16]). In other implementations [10, 15, 17], path exploration has been addressed by means of multilevel techniques, in which a multi-resolution pyramid of the input image is created, and Retinex ratio- reset operation, applied to the neighborhood of the pixel, is iterated from the coarsest to the finest level.

An important distinction among the sampling implementations regards the way pixel computing is scheduled across the path: the implementations closer to the original model update each pixel along the path, while the implementations in the MI-Retinex [6] family of algorithms apply the update of the pixel only at the end of each path. In Brownian Retinex [11], proposed by Marini and Rizzi, paths generation is based on an approximation of Brownian paths (Fig. 1b), inspired by the results of neuro-physiological research about human cortical vision areas, where the distribution of receptive fields centroids mimics a Brownian path, as demonstrated in many experiments by Zeki [18]. Brownian motion has been also investigated in [19] by Montagna and Finlayson, where it is proposed an efficient algorithm for the generation of pseudo-Brownian paths (Fig. 1c), applied to the original Retinex approach.

The sampling approach has been investigated in [20], where the authors have proved the redundancy of the MI-Retinex path-wise approach, demonstrating that, along a path, only the pixel with the highest value is relevant for the lightness calculation. As a consequence, they have proposed a simplified reformulation of MI-Retinex models and a novel implementation, called RSR (Random Spray Retinex) [21] where paths are replaced by two-dimensional point sampling (called sprays) across the image (Fig. 2), and the ratio-reset operation is replaced by a single ratio between the input pixel and the maximum value found in the spray. STRESS (Spatio-Temporal Retinex-inspired Envelope with Stochastic Sampling) [22] uses the same spray approach of RSR, but with an alternative technique based on the consideration for each pixel not only of the local maximum (reference white), but also of the local minimum (reference black), in each chromatic channel. RSR approach has been used also in the design of the RACE algorithm [23], which combines RSR with the ACE computational model [12], another member of the SCA family.

The integrating approach follows a modification proposed by Land [24] in one of his latest work, in which he has proposed to compute the lightness in a pixel using the average of a surround of the pixel, weighted using the square distance from the center. This approach is considerably different from the original Retinex formulation since it uses no more the reset mechanism. This absence, together with the average on a fixed region, makes it a Gray World type of algorithm, and not a

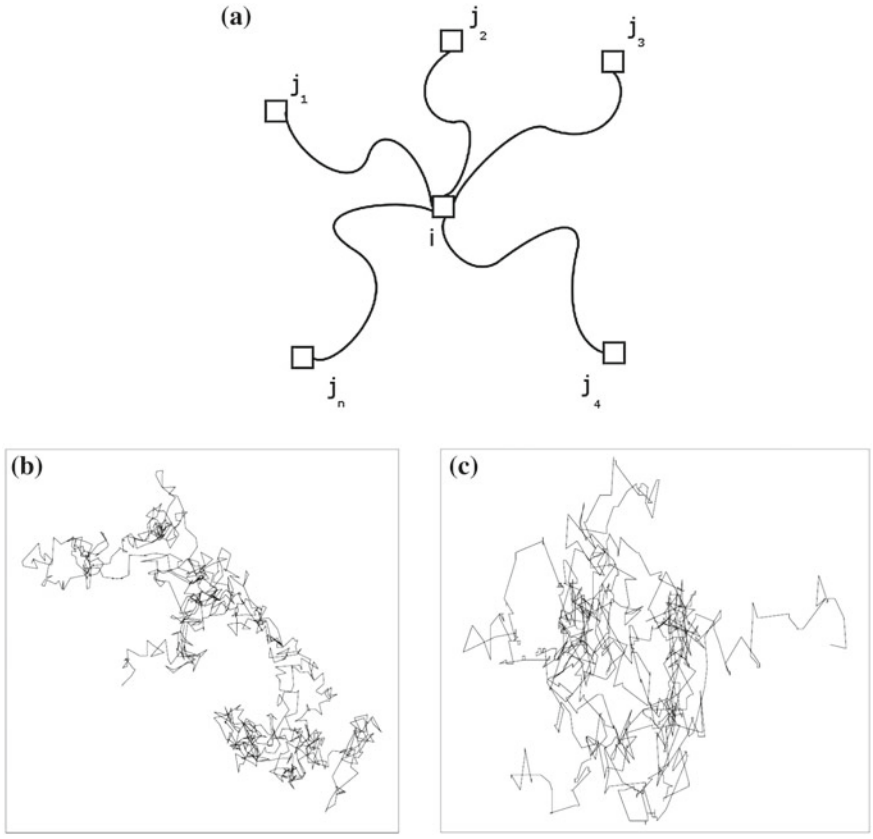

Fig. 1 Representation of N random paths. Traditional Brownian motion investigation proposed by Marini and Rizzi on the *bottom-left*, Pseudo-Brownian motion investigation proposed by Montagna and Finlayson. Figure from [19]. **a** Random paths. **b** Brownian paths. **c** Pseudo-Brownian paths

White Patch one like the original Retinex. In this model, locality is considered on the basis of the choice of the size of the surround, and of its sampling. Using a small surround leads to a significant increase in local contrast but induces halo artifacts along high contrast edges and an overall effect of desaturation in final color rendition. On the contrary, adopting a larger surround reduces the artifacts, but provides less increase in local contrast. Jobson et al. [25, 26] have taken inspiration from this center/surround approach to develop the Multi Scale Retinex (MSR), in which they have computed the weighted average of the surround by convolving the image with a normalized kernel function. To overcome halo problems and color desaturation, they have refined the algorithm by introducing a multilevel approach, and a color restoration stage, respectively. Different modifications to MSR have been proposed, like e.g. Ramponi et al. [27].

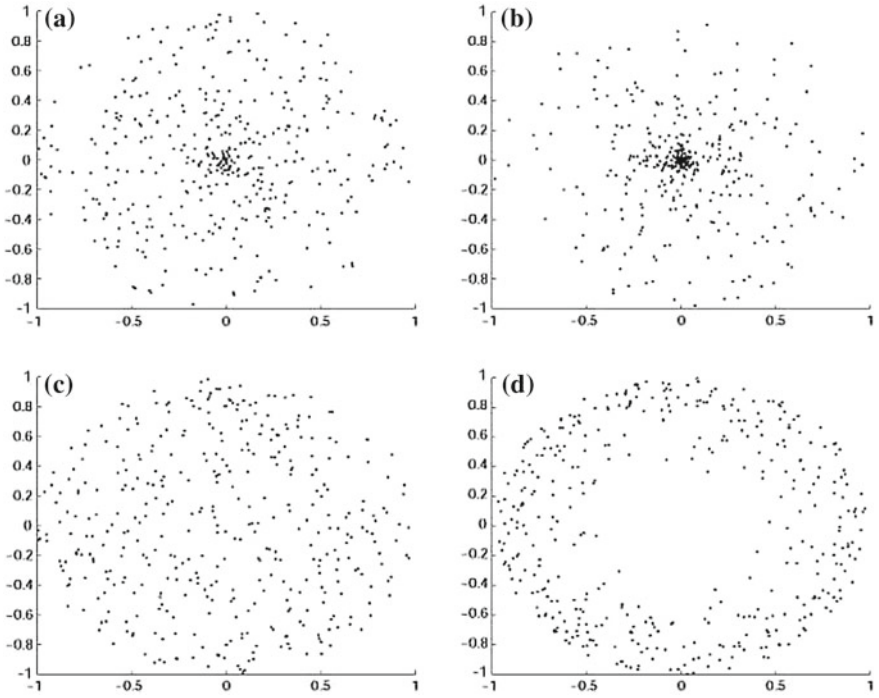

Fig. 2 Four examples of random sprays used by RSR. The density of a spray depends on three parameters: the radius of the spray, the radial density function, and the number of pixels per spray. Figure from [21]

The Retinex differential mathematical approaches belong to the second family, since their locality depends on the kernel they use to implement their variational approach, without the use of path sampling (e.g. [28, 29]).

3 Termite Retinex

"Termite Retinex", in short TR, is a path based MI-Retinex implementation that takes into account a swarm intelligence model. TR is directly derived from the Ant Colony Optimization (ACO) model proposed by Dorigo et al. [30, 31] for the Traveling Salesman Problem, which we briefly recall in order to be able to introduce TR.

The Traveling Salesman Problem (TSP) is probably the most famous NP-hard problem in combinatorial optimization and theoretical computer science. Consider a salesman who must visit n cities labeled $v_0, v_1, \ldots, v_n$. The salesman starts in city v_0, his home, and he wants to find an ordered tour, in which he can visit all

the other cities only once and come back home, traveling as little total distance as possible [32]. In other words:

Definition 1 Given a set of n cities and a pairwise distance function d(r, u), is there a tour of length $\leq D$?

In the original Ant Colony System [30], when cities are on a plane, and a path (edge) exists between each pair of cities (i.e., the TSP graph is completely connected), an artificial ant k in city r chooses the city s to move to among those which do not belong to its working memory M_k by applying the following probabilistic formula:

$$p_k(r,s) = \begin{cases} \frac{(\tau_{r,s})^{\alpha}(\eta_{r,s})^{\beta}}{\sum_{u \notin M_k}(\tau_{r,s})^{\alpha}(\eta_{r,s})^{\beta}} & \text{if } s \notin M_k \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where $\tau_{r,u}$ is the amount of pheromone trail on edge (r, u), $\eta_{r,u}$ is a heuristic function called visibility, which is the inverse of the distance between cities r and u and, α and β are parameters that allow a user to control the importance of the trail versus the visibility. The memory M_k is the taboo list of the kth ant, which contains the cities that it has already visited. City s is inserted in the list when the ant transits from city r to city s. The choice criteria of the parameters α and β can differ widely according to the problem for which the ACO is used. A guideline on how to choose the values of the different parameters for the TSP problem can be found in [31].

Essentially, from the origin of the ACO model to its consecutive works, three ideas from natural ant behavior are transferred to the artificial ant colony:

1. The preference for paths with a high pheromone level;
2. The higher rate of growth of the amount of pheromone on shorter paths;
3. The trail mediated communication among ants.

In TR a convex combination of Eq. 4 is derived with a different leading principle, where an artificial termite k in pixel r chooses to move to the pixel s among those that belong to the 8-neighborhood N_8 and that do not currently belong to its working memory M_k by applying the following probabilistic formula:

$$p_k(r,s) = \begin{cases} \frac{(\theta_s)^{\alpha}(c_{r,s})^{\beta}}{\sum_{u \notin M_k \text{ and } u \in N_8}(\theta_u)^{\alpha}(c_{r,s})^{\beta}} & \text{if } s \notin M_k \text{ and } s \in N_8 \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where θ_u is the amount of poison on pixel u, $c_{r,u}$ is the bilateral distance between pixels r and u and, α and β are parameters weighting the importance of the poison versus the closeness, which is directly related to the brightness of the pixel. In the case all the surrounding pixels have the same probability, one pixel is drawn randomly with uniform probability. In our model the memory M_k is the taboo list of the kth termite, which contains the coordinates of the pixels that the termite has already visited. This list is updated inserting the coordinates of pixel s when the termite transits from pixel r to pixel s. The poison is the inverse of the amount of pheromone, once a termite has transited on pixel u, the quantity of poison on pixel u is updated as follows:

$$\tau_u = \tau_u + Q \tag{6a}$$

$$\theta_u = \frac{1}{\tau_u} \tag{6b}$$

where Q is a chosen amount of poison with $0 < Q \leq 1$. As in Retinex model, each pixel is treated independently, after processing a pixel, the amount of poison at all pixels must be reset to the initial value of 1 before starting processing a new pixel.

Therefore, artificial termites are essentially governed by three principles:

1. The preference for pixels with a low poison level. Divergence is required, in order to explore different areas of the image: the poison acts as a repulsion "force", inverse of the attraction force pheromone in ACO;
2. The growth of the amount of poison on visited pixels. The higher the quantity of the poison added on a pixel, the stronger the divergence behavior of the termites. Furthermore the poison is a feature that prevents the paths from a complete randomness since it affects the glocality, it can be used as a tuning parameter in order to obtain different spatial effects;
3. The length of the path, which also affects the glocality of the filtering. In particular, a termite should never travel across the whole set of pixels in the image, because it would lead the filtering to the global white asymptote and thus a to a global white normalization of the image content.

Furthermore, the following constraints are required to be a novel tool for exploring glocality:

1. A termite can choose to move only to one of the 8-neighboring pixels, jumps forbidden. By exploring larger neighborhoods and allowing the termite to "jump" might lead to not discovering the proper local reference white;
2. The choice of a pixel is based on its distance and intensity value: the visibility η is substituted with the bilateral distance c as defined below, that we will refer to as "closeness". The use of the bilateral distance is known to be a suitable tool for edge-preserving [33] and in TR is justified by the fact that the presence of halos are reduced in the final image with respect to the simple distance based only in intensity values (Fig. 3).

The bilateral distance $c_{r,u}$ is defined as follows:

$$c_{r,u} = \frac{d_e + d_v}{\sqrt{2}} \tag{7a}$$

$$d_e = \sqrt{(x_r - x_u)^2 + (y_r - y_u)^2} \tag{7b}$$

$$d_v = |I(x_r, y_r) - I(x_u, y_u)| \tag{7c}$$

where d_e and d_v are the distance in coordinates and in intensity values respectively, I is the image channel and (x, y) are the coordinates of the pixels. When a termite has

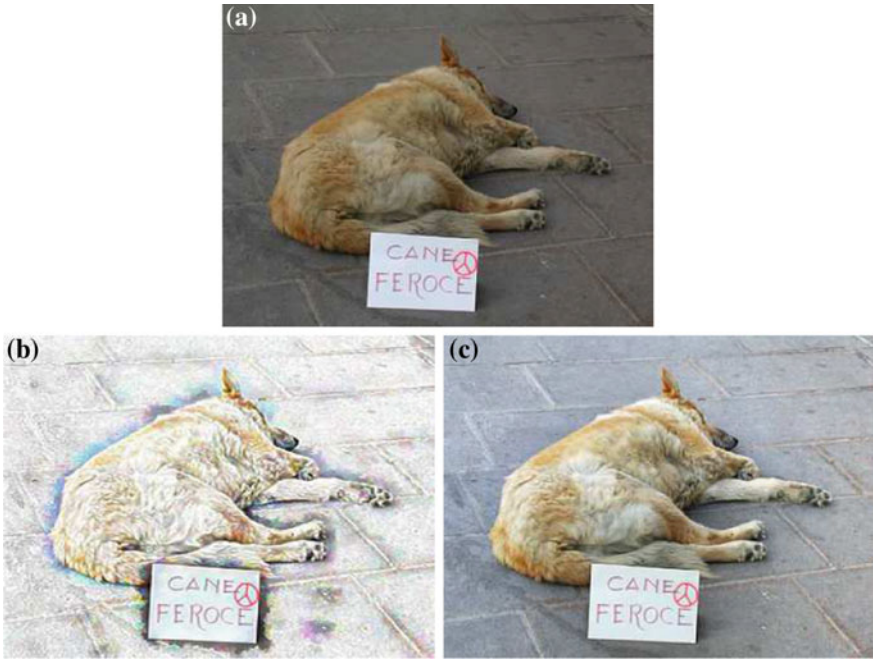

Fig. 3 Halos appearing using only the simple distance on intensity values on the *bottom-left*, using bilateral distance on the *bottom-right*. **a** Original. **b** Only intensity values. **c** With bilateral distance

to choose among eight neighboring pixels with same intensity values, the preference goes to one of the four diagonal pixels with respect to the vertical and horizontal ones. This phenomenon of preference for the diagonal pixel gradually disappears with a high number of termites and with the participation of the poison.

At the current stage, the reader may wonder why the proposed method is not called "Ant Retinex". Two reasons motivate the choice of "Termite Retinex". Our artificial termites attempt an eager exploration in search of the local reference white, in analogy with biological worker termites, also known as "white ants", which undertake the labors of foraging. As well as the behavior of biological termites change with the evolving of the nest structure, our artificial termites change their behavior with the exploring of the image content.

4 Termite Retinex Locality

TR glocality is controlled by the following five parameters:

- α and β: they determine the trade off the quantity of poison found on the pixel and between the brightness of a pixel to choose; In the case of high values of β,

a pixel is chosen, even if already visited by another termite. On the contrary high values of α force a termite to choose another direction;
- k (number of termites): it determines the size of the swarm, specifically higher is the number of termites, higher is the chance to find the proper local reference white;
- N_s (length of the path): it determines how far a termite should travel, specifically how wide is the area of the image to be explored by the swarm;
- Q (quantity of poison): it determines the strength of the divergence behavior of the swarm.

In TR, the values of the parameters, do not change during pixel recomputation, they are kept constant for the whole processing of the image.

Essentially, α and β determine the behavior of a single termite, k, N_s, and Q influence the glocality and the final visual sensation of the image.

Figure 4 shows the behavior of the termites with two different configurations of α and β, specifically $\alpha = 0.9$ and $\beta = 0.1$ in Fig. 4a, and $\alpha = 0.1$ and $\beta = 0.9$ in Fig. 4b. The higher the value of α and the lower the value of β, the wider the area explored by the termites. Viceversa, the lower the value of α and the higher the value of β, the smaller the area of the image explored, making the termites to forage the

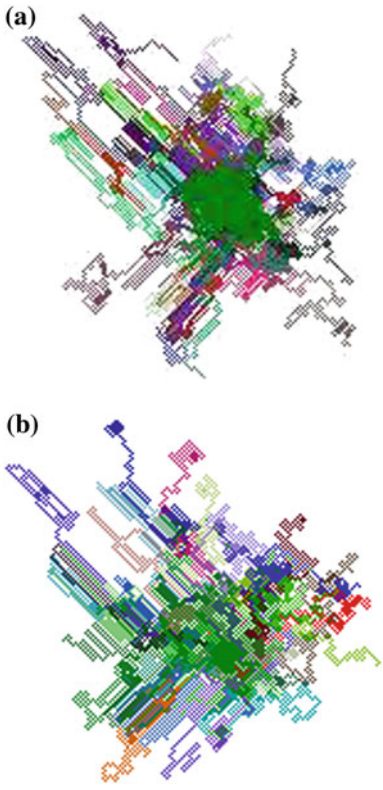

Fig. 4 Termite investigation with two different configurations of α and β and quantity of poison Q = 1. The path of each termite is distinguished by a different color. A high value of α and a low value of β make the termite swarm to explore a wide area of the image as shown on the *top*, while a low value of α and a high value of β make the termite swarm to explore a smaller area of the image as shown on the *bottom*. **a** $\alpha = 0.1$, $\beta = 0.9$. **b** $\alpha = 0.9$, $\beta = 0.1$

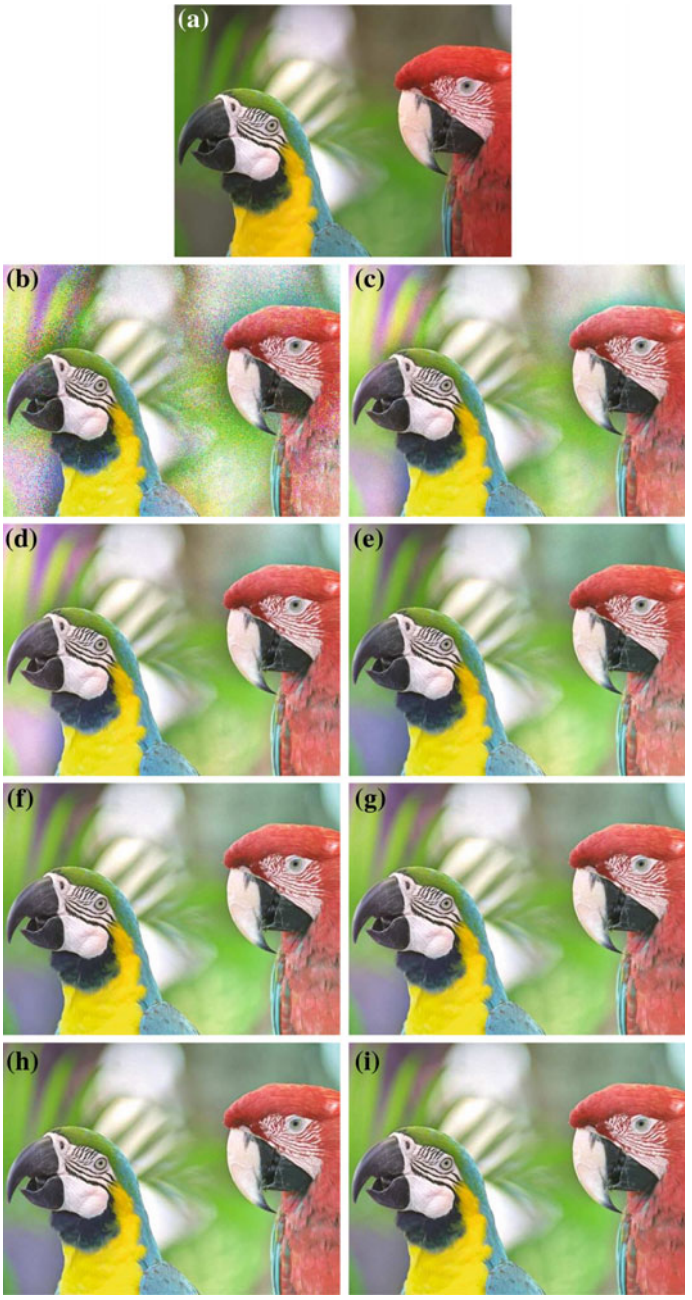

Fig. 5 Termite Retinex filtering with $N_s = 500$, $\alpha = 0.1$, $\beta = 0.9$, and $Q = 1$, with increasing number of termites k. A few number of termites may lead to chromatic noise while a too large number of termites may lead to a low contrast enhancement and loss of details. **a** Original. **b** $k = 1$. **c** $k = 10$. **d** $k = 50$. **e** $k = 100$. **f** $k = 250$. **g** $k = 500$. **h** $k = 750$. **i** $k = 1{,}000$

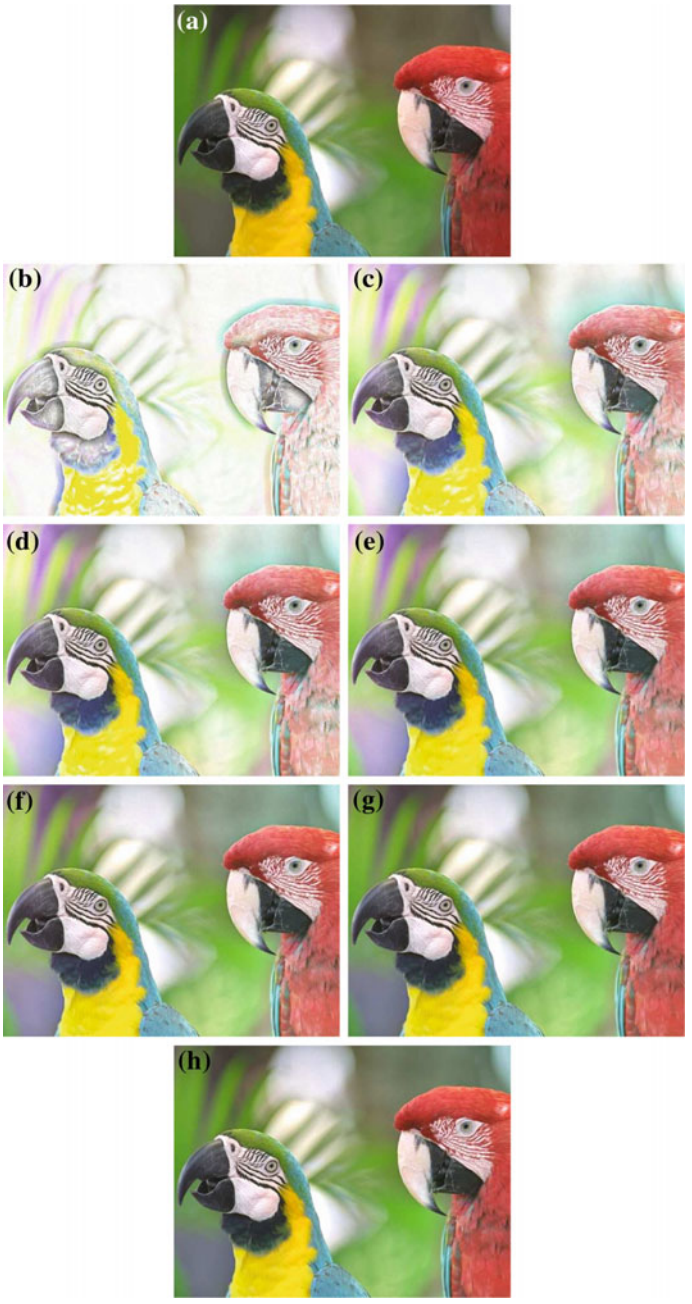

Fig. 6 Termite Retinex filtering with $k = 500$, $\alpha = 0.1$, $\beta = 0.9$, and $Q = 1$, with increasing length of the path N_s. A too short path may lead to edge overenhancing while a too long path may lead in a loss of the local effect. **a** Original. **b** $N_s = 10$. **c** $N_s = 50$. **d** $N_s = 100$. **e** $N_s = 250$. **f** $N_s = 500$. **g** $N_s = 750$. **h** $N_s = 1{,}000$

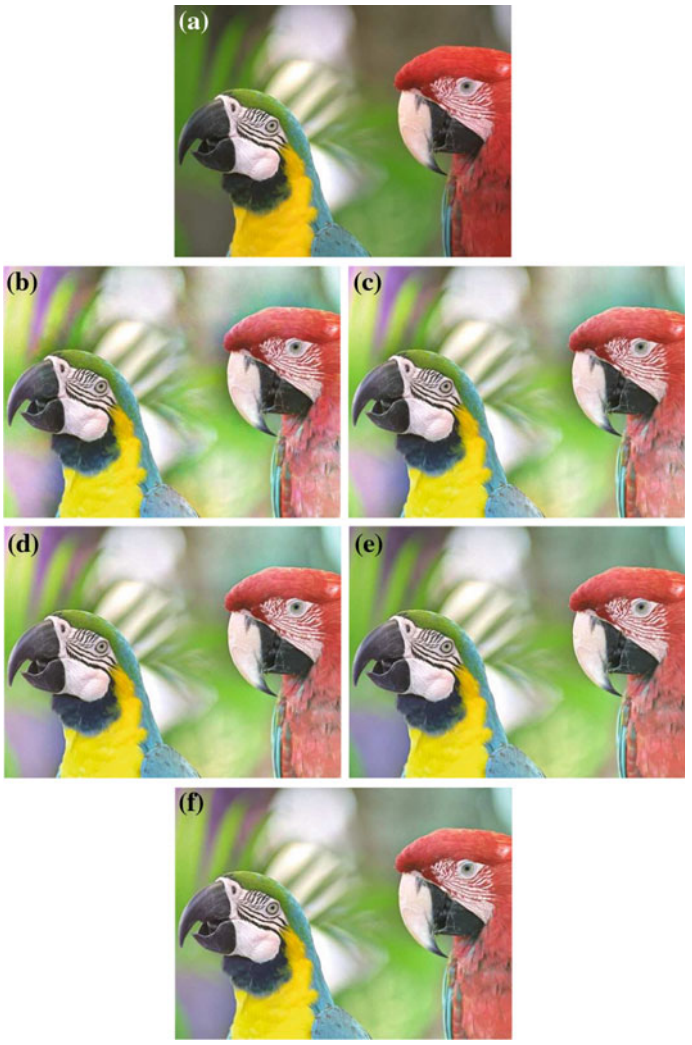

Fig. 7 Termite Retinex filtering with $k = 500$, $\alpha = 0.1$, $\beta = 0.9$, and $N_s = 500$, with increasing quantity of poison Q. Very low quantities of poison may lead to the presence of halos. **a** Original. **b** $Q = 10^{-4}$. **c** $Q = 10^{-3}$. **d** $Q = 10^{-2}$. **e** $Q = 10^{-1}$. **f** $Q = 1$

same local reference white. The particular case, with $\alpha = 0$ and $\beta = 0$, yields to complete random paths.

Figure 5 shows an example of filtering with $\alpha = 0.1$ and $\beta = 0.9$, constant path length and quantity of poison, but varying number of termites. As it can be observed in Fig. 5c, d, too few termites, due to insufficient image sampling, can lead to a high chromatic noise. On the other side a large number of termites leads to low contrast enhancement and loss of details. In the proposed example this can be particularly

seen in the breast of the red parrot, where the details of the feathers are lost in Fig. 5h, i with respect to Fig. 5e–g.

Figure 6 shows the previous example of filtering, but this time, varying the length of the path N_s and keeping constant all the other parameters. Too short paths lead to edge overenhancing and the presence of halos as it can be observed in Fig. 6b–d. On the contrary, too long paths result in a loss of the local effect and in not very bright images as it can be seen in Fig. 6g, h with respect to Fig. 6e, f. The specific case with the longest path $N_s = n$ with $n = width \times height$, or in other words making the termites visiting all pixels, yields to global white normalization. This normalization, known also as "White Patch" [34], performs a global enhancement of the three color channel separately dividing each pixel for the maximum value of the channel. It is important to recall that differently from Von Kries classic White Patch, Retinex normalization to the channel maxima is local for each pixel.

As the last parameter to discuss, we present in Fig. 7 an example of filtering varying only the quantity of poison Q. Very low quantities of poison lead to the presence of halos as it can be observed in Fig. 7b–d with respect to Fig. 7e, f. This is due by the fact that lowering the quantity of poison, reduces the divergence of the swarm making the termites foraging the local reference white in a smaller area. This may lead also the termites to discover the same local maxima as reference white.

Previous studies has shown that $\alpha = 0.1$ and $\beta = 0.9$, $k \approx \min(height, width)$, $N_s \approx 70\,\%$ of the length of the diagonal, and $Q = 1$ are in line with observer preference [5].

5 Termite Retinex Properties

5.1 Computational Complexity

The computational complexity of TR is given by:

$$O(k \cdot N_s \cdot n) \tag{8}$$

where k is the number of termites, N_s is the length of the path, and n is the number of pixels in the image. TR has the same computational complexity of other SCAs, such as RSR [21] or STRESS [22] which have a computational complexity of $O(NMn)$, where N is the number of iterations, M is the number of samples and n is the number of pixels in the image.

In the last years, Retinex algorithms have been implemented using different programming languages and environments (like e.g. Matlab [17, 19] and C++ [21, 22]), and on different architectures and devices (like e.g. on GPU [35] and in digital cameras [36]). In all these cases, the way the algorithm is implemented effects relevantly its overall performance.

We give some remarks for an efficient TR implementation:

- Each color channel is treated independently like in the original model, thus each channel can be processed in parallel.
- As well, each pixel in each color channel is recomputed independently, thus all pixels of the image can be processed in parallel.
- Each termite is an autonomous worker, so each termite foraging can be parallelized.
- Each termite allocates its taboo list. The higher the number of termites and the length of the path, higher the memory consumption.
- The higher the length of the path, the higher becomes the computational time.

5.2 Dynamic Range Stretching

An important property of TR in line with other SCAs is the ability of performing dynamic range stretching. TR, as many other Retinex implementations and SCAs, performs dynamic range stretching processing all the pixel-value population but also taking into account the local spatial relationships of the image content, which control appearances and thus the final visual sensation of the image [6].

Figure 8 presents an example of dynamic range stretching on a grey scale image. As it can be observed, the original image presents a reduced dynamic range, with

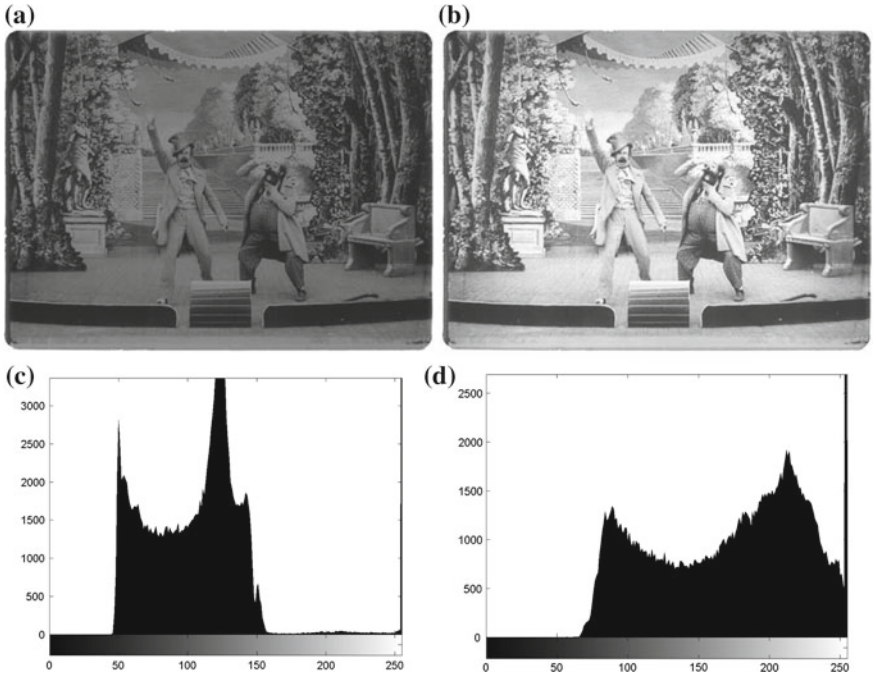

Fig. 8 An example of unsupervised dynamic range stretching on grey scale image. **a** Original. **b** Termite Retinex. **c** Original histogram. **d** Termite Retinex histogram

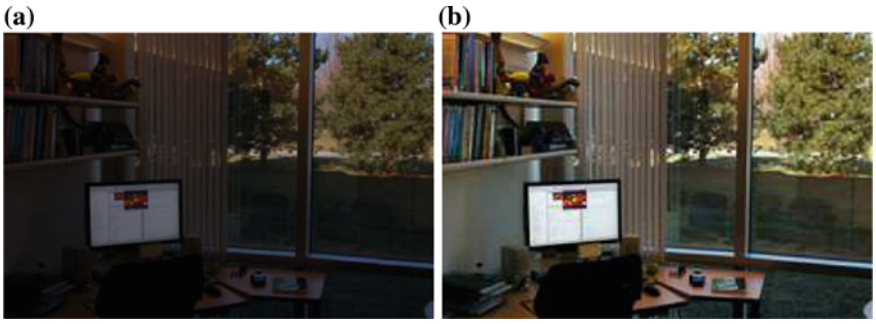

Fig. 9 An example of Termite Retinex unsupervised dynamic range stretching on color image. **a** Original. **b** Termite Retinex

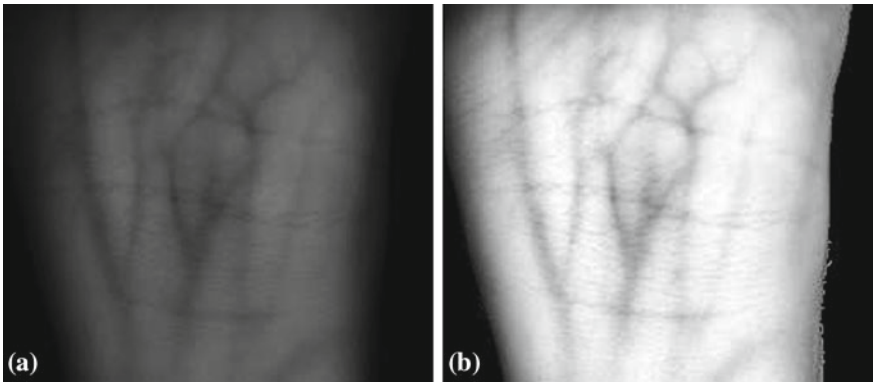

Fig. 10 An example of Termite Retinex unsupervised enhancement of a medical image. Original image provided with courtesy by Daniel Hartung [37]. **a** Original. **b** Termite Retinex

the most of the pixel-value population in the range of neutral grey and lower. TR filtering stretches and enhances the used dynamic range, and the resulting pixel-value population is centered in higher ranges.

Likewise, TR is able to perform dynamic range stretching processing on color images (Fig. 9) and on medical images (Fig. 10).

5.3 Color Constancy and Color Correction

Color Constancy is a complex issue largely debated in the imaging community [1]. In digital imaging literature, there are two different approaches to color constancy: Computer Vision CC (in some works referred as Computational CC, but we prefer to refer it as CV CC since both CCs have computational aspects), in short CV CC and Human Vision CC, in short HV CC. They have distinct goals in modeling this

Fig. 11 An example of Termite Retinex unsupervised color correction on Lena portrait. **a** Original. **b** Termite Retinex

phenomenon and thus different kind of outcomes are expected, and different measures of performance are required.

Computer Vision CC has the goal of separating illuminant from reflectance or alternatively estimating the physical reflectance of objects in different illuminants, or alternatively estimating the illuminant spectral or colorimetric component. This is a well-known ill-posed problem [38], thus these algorithms need constraints or assumptions on the scene content, illumination or geometry. In any case, CV CC aims to cancel totally the interaction between reflectance and illumination.

In Human Vision CC, the illuminant component is not totally canceled, it generates appearances that are close to reflectance, but with significant departures. These departures serve as important signatures of the underlying visual mechanisms [6]. For many years these two variables, reflectance and appearance, have been treated as an unique correlated feature. This has been proven to be an incorrect assumption [1, 39]. HV CC aims at computing appearance and algorithms belonging to the Retinex family, and thus TR, share this approach attempting to mimic the response of human vision.

Figure 11 shows the TR effect of unsupervised color cast reduction on the classic Lena portrait.

5.4 HDR Tone Rendering

As every SCA algorithm, TR can be used as a tone renderer for High Dynamic Range (HDR) images [6]. HDR images are formed by values that span over a range much wider than the range allowed by the visualization devices. For these images a tone rendering operation is necessary to display all the values preserving as much as

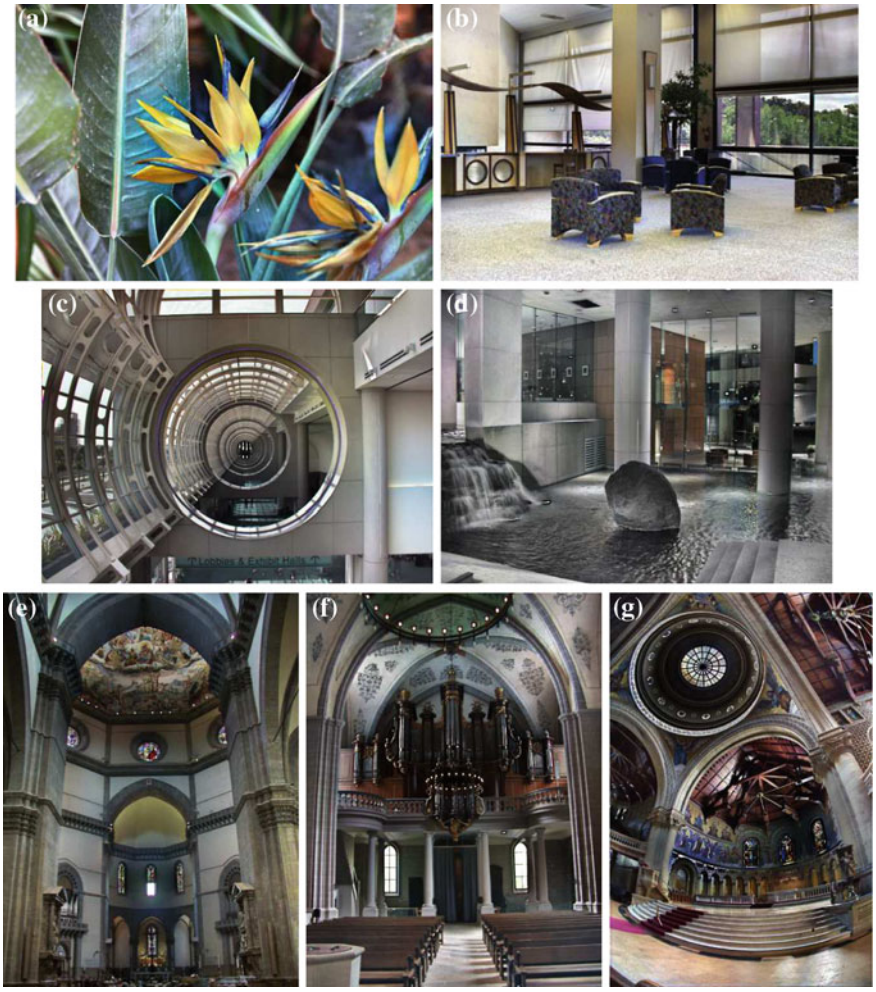

Fig. 12 Examples of unsupervised tone rendering of HDR images. **a** Paradise flower. **b** Chairs. **c** Convention center. **d** Vancouver building. **e** Duomo. **f** StFrancis. **g** Memorial. *Source*: web

possible their original appearance. Figure 12 shows some examples of the application of TR as a tone renderer for HDR images from the web.

6 Conclusions

In this chapter, we have presented an example of swarm intelligence applied for color and contrast enhancement of digital images, inspired by the Human Vision

System. In particular, we have presented a Retinex implementation, called Termite Retinex, in which the glocality (glocal = global + local) of image exploration is exploited through a colony of agents. In Termite Retinex, the idea of paths (the basis of the sampling approach of the original Retinex model) is reconsidered from the point of view of the Ant Colony Optimization model: the local filtering of Retinex is investigated through a set of "termites" moving inside the image. By tuning a small set of parameters, we can control the overall behavior of the swarm and therefore the final effect of the Retinex filtering. We have presented a detailed explanation of the swarm intelligence model used in Termite Retinex, and a discussion of its parameters. To demonstrate the capabilities of Termite Retinex, we have shown different examples and applications like unsupervised image enhancement, dynamic range stretching, color correction, and high dynamic range tone rendering.

References

1. McCann, J.J., Parraman, C., Rizzi, A.: Reflectance, illumination, and appearance in colorconstancy. Front. Psychol. **5**(5), (2014)
2. Rizzi, A., McCann, J.J.: On the behavior of spatial models of color. In: IS&T/SPIE Electronic Imaging, vol. 6493, p. 649303. San Jose, California, USA, January 2007
3. Parraman, C., Rizzi, A.: Searching user preferences in printing: a proposal for an automatic-solution. In: Printing Technology SpB06, St. Petersburg, Russia, June 2006
4. Parraman, C., Rizzi, A.: User preferences in color enhancement for unsupervised printing-methods. In: SPIE, vol. 6493, pp. 64930U–64930U-11 (2007)
5. Simone, G., Audino, G., Farup, I., Albregtsen, F., Rizzi, A.: Termite Retinex: a new implementation based on a colony of intelligent agents. J. Electron. Imaging **23**(1), 013006-1-13 (2014)
6. McCann, J.J., Rizzi, A.: The Art and Science of HDR Imaging. Wiley, New York (2011). ISBN: 978-0-470-66622-7
7. Land, E.H.: The Retinex. Am. Sci. **52**, 64–247 (1964)
8. Land, E.H., McCann, J.J.: Lightness and Retinex theory. J. Opt. Soc. Am. **61**(1), 1–11 (1971)
9. McCann, J., McKee, S., Taylor, T.: Quantitative studies in Retinex theory. A comparison between theoretical predictions and observer responses to the color Mondrian experiments. Vis. Res. **16**(5), 445–458 (1976)
10. McCann, J.J.: Lessons learned from Mondrians applied to real images and color gamuts. IS&T Rep. **14**(6), 1–7 (1999)
11. Marini, D., Rizzi, A.: A computational approach to color adaptation effects. Image Vis. Comput. **18**, 1005–1014 (2000)
12. Rizzi, A., Gatta, C., Marini, D.: A new algorithm for unsupervised global and local colorcorrection. Pattern Recognit. Lett. **24**, 1663–1677 (2003)
13. Land, E.H.: The Retinex theory of color vision. Sci. Am. **237**, 108–128 (1977)
14. McCann, J.J.: Retinex at 40. J. Electron. Imaging **13**(1), 6–7 (2004)
15. Frankle, J.J., McCann, J.J.: Method and apparatus for lightness imaging (1983)
16. Cooper, T.J., Baqai, F.A.: Analysis and extensions of the Frankle-Mccann Retinex algorithm. J. Electron. Imaging **13**(1), 85–92 (2004)
17. Funt, B., Ciurea, F., McCann, J.J.: Retinex in MATLAB. J. Electron. Imaging **13**(1), 48–57 (2004)
18. Zeki, S.: A Vision of the Brain. Blackwell Scientific Publications, Oxford (1993)
19. Montagna, R., Finlayson, G.D.: Constrained pseudo-Brownian motion and its application to image enhancement. J. Opt. Soc. Am. A **28**(8), 1677–1688 (2011)

20. Provenzi, E., Carli, L.D., Rizzi, A.: Mathematical definition and analysis of the Retinexalgorithm. J. Opt. Soc. Am. A **22**(12), 2613–2621 (2005)
21. Provenzi, E., Fierro, M., Rizzi, A., Carli, L.D., Gadia, D., Marini, D.: Random spray Retinex: a new Retinex implementation to investigate the local properties of the model. IEEE Trans. Image Process. **16**(1), 162–171 (2007)
22. Kolas, ø., Farup, I., Rizzi, A.: Spatio-temporal Retinex-inspired envelope with stochastic sampling: a framework for spatial color algorithms. J. Imaging Sci. Technol. **55**(4):1–10 (2011)
23. Provenzi, E., Gatta, C., Fierro, M., Rizzi, A.: A spatially variant white-patch and gray-world method for color image enhancement driven by local contrast. IEEE Trans. Pattern Anal. Mach. Intell. **30**(10), 1757–1770 (2008)
24. Land, E.H.: An alternative technique for the computation of the designator in the Retinex theory of color vision. PNAS **83**(10), 3078–3080 (1986)
25. Jobson, D.J., Rahman, Z., Woodell, G.A.: Properties and performance of a center/surround Retinex. IEEE Trans. Image Process. **6**(3), 451–462 (1997)
26. Jobson, D.J., Rahman, Z.U., Woodell, G.A.: A multiscale Retinex for bridging the gap between color images and the human observation of scenes. IEEE Trans. Image Process. **6**(7), 965–976 (1997)
27. Ramponi, G., Tenze, L., Carrato, S., Marsi, S.: Nonlinear contrast enhancement based on the Retinex approach. In: Proceeding of the Image Processing: Algorithms and Systems II, Santa Clara, CA, USA January 2003, vol. 5014, pp. 169–177 (2003)
28. Kimmel, R., Elad, M., Shaked, D., Keshet, R., Sobel, I.: A variational framework for Retinex. Int. J. Comput. Vis. **52**(1), 7–23 (2003)
29. Bertalmo, M., Cowan, J.D.: Implementing the Retinex algorithm with Wilsoncowan equations. J. Physiol.-Paris **103**(1–2), 69–72 (2009) (Neuromathematics of Vision)
30. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern.-Part B **26**(1), 29–41 (1996)
31. Dorigo, M., Gambardella, L.: Ant colonies for the traveling salesman problem. BioSystems **43**, 73–81 (1997)
32. Kleinberg, J., Tardos, E.: Algorithm Design. Addison-Wesley Longman Publishing Co. Inc., Boston (2005)
33. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the Sixth International Conference on Computer Vision, ICCV 98, Bombay, January 1998, pp. 839–846. IEEE Computer Society
34. von Kries, J.: Sources of color science. Chromatic Adaptation, pp. 109–119. MIT Press, Cambridge (1970)
35. Wang, Y.-K., Huang, W.-B.: Acceleration of the Retinex algorithm for image restoration by GPGPU/CUDA. In: Parallel Processing for Imaging Applications, number SPIE 7872, San Francisco, CA, USA January 2011
36. Sobol, R.: Improving the Retinex algorithm for rendering wide dynamic range photographs. J. Electron. Imaging **13**(1), 65–74 (2004)
37. Hartung, D.: Vascular pattern recognition: and its application in privacy-preserving biometric online-banking systems. Ph.D. thesis, Gjϕvik University College (2012)
38. Barnard, K., Cardei, V., Funt, B.: A comparison of computational color constancy algorithms - part i: methodology and experiments with synthesized data. IEEE Trans. Image Process. **11**(9), 985–996 (2002)
39. Albers, J.: Interaction of Color. Yale University Press, New Haven (1975)