Published in Image Processing On Line on YYYY–MM–DD.
Submitted on YYYY–MM–DD, accepted on YYYY–MM–DD.
ISSN 2105–1232
This article is available online with supplementary materials, software, datasets and online demo at
http://www.ipol.im/

Termite Retinex

Gabriele Simone [1], Mauro Fiorentini[1] Alessandro Rizzi [1]

[1] Universita' degli Studi di Milano, Dipartimento di Informatica, Milano, Italy (gabriele.simone@di.unimi.it)

PREPRINT March 14, 2024

Abstract

The original presentation of Retinex, a spatial color correction and image enhancement algorithm modeling the Human Vision System, uses paths to explore the image in search of a local reference white point.

Here we present a spatial color algorithm, called Termite Retinex, with an alternative way to explore local properties of Retinex, replacing random paths with a colony of agents, which uses swarm intelligence to explore the image, determining in this way the locality of its filtering.

We show the efficacy of Termite Retinex for unsupervised image enhancement, dynamic range stretching and color correction for digital images.

1 Introduction

Retinex is an algorithm that models how our visual system synthesizes the chromatic appearance of a scene. It is now scientifically established that the chromatic appearance of a point depends both on the value of the color signal in the point and on the relationship between this value and the other ones that make up the rest of the observed scene [24, 8, 1, 2]. In its first fifty years, and even a little longer, Retinex has generated a lot of interest and many different implementations [13, 23, 14, 19]. Making a complete comparison between the many implementations can be very difficult due to the fact that many of these lack the available code; while having the same code for all is essential to guarantee the correctness of the comparison.

A characteristic of the entire family of Retinex algorithms is that of estimating the chromatic appearance value of the point by considering not only the value of the point but above all its relationship with all the other pixels that make up the image. In doing this Retinex is characterized by a global effect of adjustment of the brightness and color of the image but above all by a local effect of adjustment of both the local contrast and the color at the point [26]. What distinguishes the many available implementations is the way in which spatial information is used to compute the pixel value. In other words, spatial exploration determines the weight of the local component and therefore different spatial explorations result in different filtering effects [21, 17, 18, 12].

A particular comment should be reserved for a variant of the retinex algorithm which derives from the latest works of Edwin Land, later developed by NASA [6]. In the original retinex the locality depends not only on the implementation of the algorithm but also on the content of the image to filter. In the NASA version of Retinex and its many derivatives, space exploration is made fixed, thus transforming Retinex computation into the execution of a convolutional filter. In this case, given the simplicity of the implementation, a possible publication of the algorithm benefits the reader little [20].

In this work we make available the code of an implementation of the original Retinex, called *Termite* [25], based on a colony of agents, an "ant colony" type. In this case, however, the agents implement an opposite exploration policy, not trying to optimize paths already visited but on the contrary trying to explore the image around the pixel in the most varied way possible.

2 Retinex Theory

Based on a set of experiments on human color vision, the Retinex algorithm was developed by Land and McCann aiming at estimating the human color sensation. The term Retinex coined by Land and McCann [9, 11, 10] refers to the roles that both retina and cortex play in human vision.

The core of the Retinex algorithm is the so-called "lightness" computation that 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:

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

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, I_{x+1} is the lightness intensity of the pixel $x+1$ and h is denoting the path. 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 [15]:

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

where T is a defined threshold. The threshold mechanism helps to discount slow varying gradients, however it has been proven that the algorithm maintains its fundamental properties even without it [16].

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. Thus, changing the method to create the random paths (i.e., their structure), we change the way locality is considered and introduced in the Retinex computation. There is a large literature available on the different approaches applied for the exploration of locality in the implementation of Retinex models [27].

The reset mechanism forces to restart the chain of ratios (sum of logs) from the point of reset, by considering the lightness value found at the reset point as new local reference white. The point i can be either along the path, like in the original Retinex, or always at the end of the path like in the MI–Retinex family as discussed in [15, 27]. Termite Retinex exploits the MI–Retinex approach, and thus does not use the logarithm of intensity values, but simply the intensity values themselves.

3 Termite Retinex

3.1 The Ant Colony System

The Traveling Salesman Problem (TSP) is one of the most famous NP–hard problems in combinatorial optimization and theoretical computer science. Consider a salesman who must visit n cities. The salesman starts in his home city, and he wants to find an ordered tour, in which he can visit all the other cities only once and come back home, traveling the less distance as possible[7]. Thus:

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 [4] for solving the TSP, cities are placed 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 [3]:

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

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;
- α and β are parameters that allow a user to control the importance of the trail versus the visibility;
- M_k is the tabu list of the k^{th} ant, containing the cities already visited. City s is inserted in the list when the ant transits from city r to city s.

From the original model to its consecutive works [5], three ideas from the natural behavior of the ants are inherited 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.

3.2 From Ants to Termites

When attempting to model the human vision system, an important characteristic to consider is the locality of the visual sensation [22, 15]. The way the image is explored affects the influence that the surrounding pixels have on the final estimate of the visual sensation. The main objective to build a colony of termites an alternative mechanism by which the image content is explored, and thus the locality of perception. In the TSP problem, all the metaheuristics attempt to find the optimal solution, while in the field of image processing the optimal solution depends on the task of the algorithm. In this work, the goal is a qualitative emulation of the HVS for an unsupervised image enhancement [27]. Thus, in order to model Termite Retinex, assumptions and constraints of ACS need to be redefined. We come up for the followings:

1. Pixels are considered as cities: a termite can choose to move only to one of the 8–neighboring pixels, jumps are forbidden. 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, that we will refer to as "closeness". The use of the bilateral distance is driven by being known as an edge–preserving tool [28].

3. Defining a new quantity poison as the inverse amount of the pheromone. Poison acts as a repulsion "force", inverse of the attraction force, pheromone in ACS.

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 by applying the following probabilistic formula:

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

where:

- θ_u is the amount of poison on pixel u;

- $c_{r,u}$ is the bilateral distance between pixels r and u; α and β are parameters weighting the importance of the poison versus the closeness, which is directly related to the brightness of the pixel; M_k is the taboo list of the k^{th} termite, containing the coordinates of the pixels already visited. This list is updated inserting the coordinates of pixel s when the termite transits from pixel r to pixel s.

In the case all the surrounding pixels have the same probability, one pixel is drawn randomly with uniform probability.

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

$$c_{r,u} = \frac{d_e + d_v}{\sqrt{2}}, \tag{6a}$$

$$d_e = \sqrt{(x_r - x_u)^2 + (y_r - y_u)^2}, \tag{6b}$$

$$d_v = |I(x_r, y_r) - I(x_u, y_u)|, \tag{6c}$$

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.

Once a termite has transited on pixel u, the quantity of poison on pixel u is updated as follows:

$$\tau_u = \tau_u + Q \tag{7a}$$

$$\theta_u = \frac{1}{\tau_u} \tag{7b}$$

where Q is a chosen amount of poison with $0 < Q \leq 1$.

Differently from artificial ants, artificial termites are then essentially governed by three principles:

1. Higher preference for paths with a low *poison*, as we want divergence, in order to explore different areas of the image.
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.
3. The length of the path affecting the locality of the filtering. In particular, a termite should never travel across the whole set of pixels in the image, leading to the global white normalization of the image content.

4 Test Results and Discussion

The final filtering of TR is given by the number of termites k and the length of the path N_s, which leads to a computational complexity of:

$$O = k \cdot N_s \cdot n, \tag{8}$$

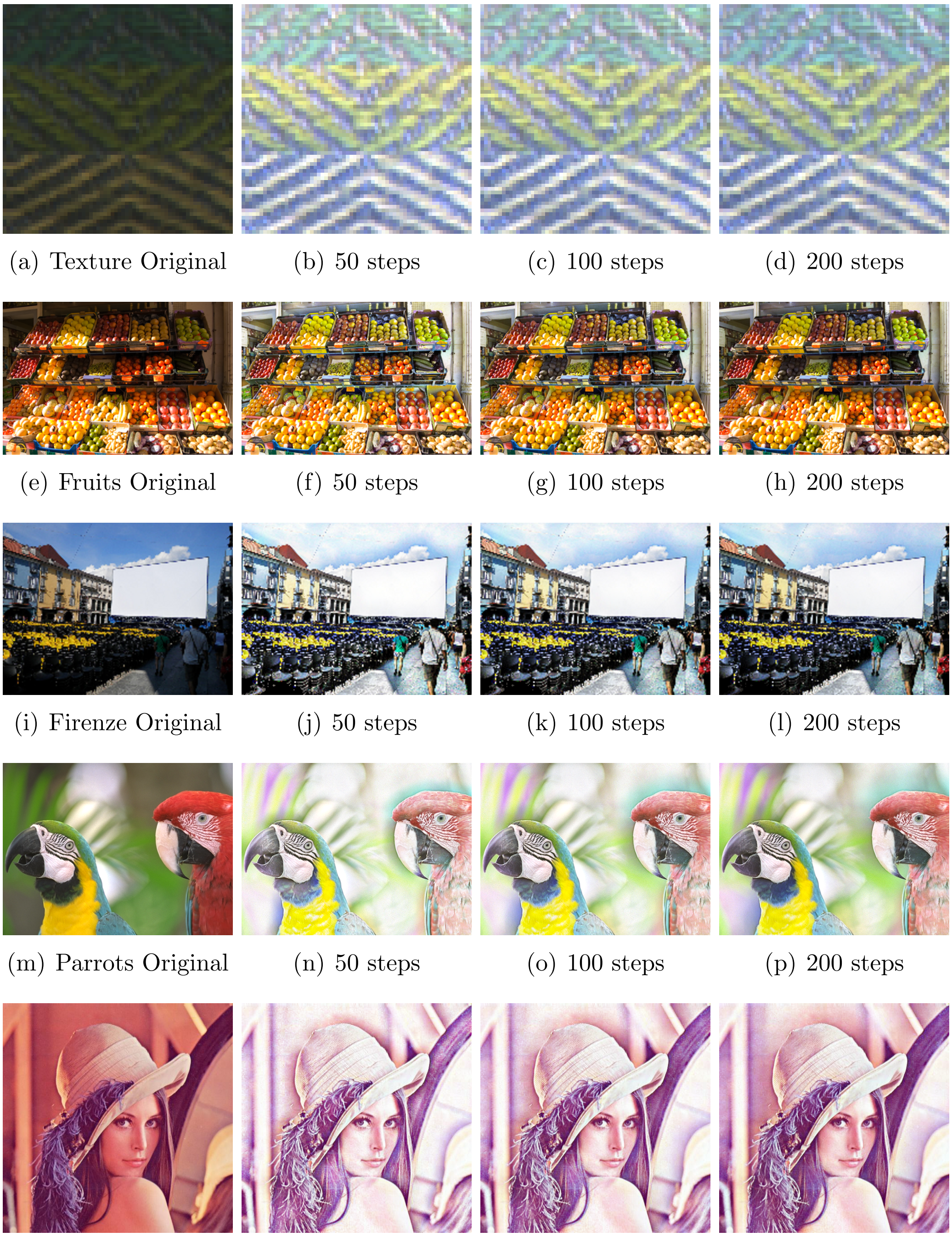
where n in this case is the size of the image (number of pixels).

An example with constant number of termites and variable length of path is show in Figure 1, while an example with constant length of the path and variable number of termites is shown in Figure 2 with: Texture (64×64), Fruits (385×256), Firenze (408×306) and Lena (512×512)

- 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.
- 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.
- Q (quantity of poison): it determines the strength of the divergence behavior of the swarm.

A rule of thumb from previous studies suggests the following configuration in accordance with observers image quality preferences:

- $k \approx 70\%$ of the length of the diagonal.
- $N_s \approx 70\%$ of the length of the diagonal.
- $\alpha = 0.1, \beta = 0.9$.



(a) Texture Original (b) 50 steps (c) 100 steps (d) 200 steps

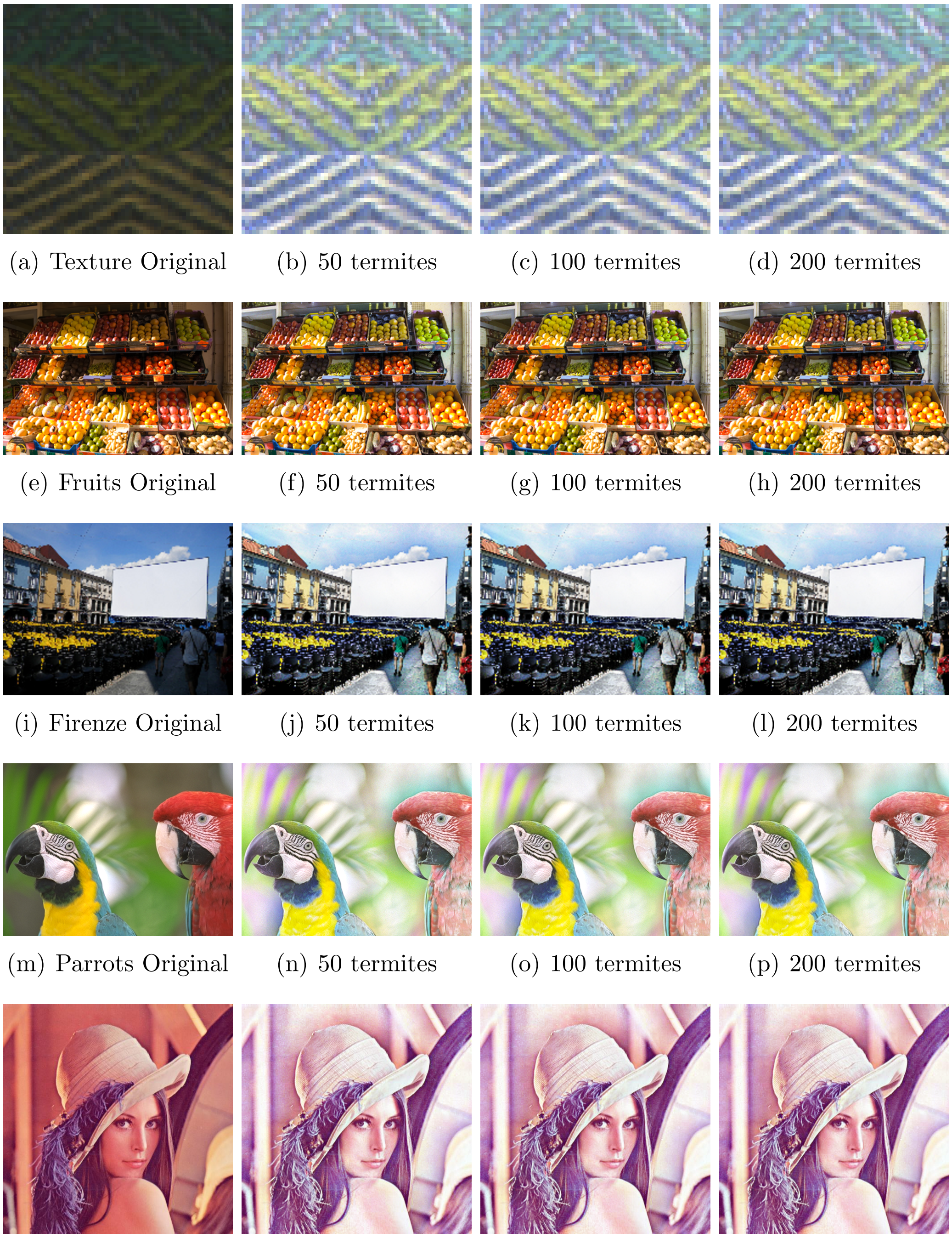
(e) Fruits Original (f) 50 steps (g) 100 steps (h) 200 steps

(i) Firenze Original (j) 50 steps (k) 100 steps (l) 200 steps

(m) Parrots Original (n) 50 steps (o) 100 steps (p) 200 steps

(q) Lena Original (r) 50 steps (s) 100 steps (t) 200 steps

Figure 1: An example of filtering with number of termites constant ($k = 200$) and increasing number of steps ($N_s = 50, 100, 200$). On the top Texture, followed by Fruits, Parrots and Lena on the bottom.



(a) Texture Original (b) 50 termites (c) 100 termites (d) 200 termites

(e) Fruits Original (f) 50 termites (g) 100 termites (h) 200 termites

(i) Firenze Original (j) 50 termites (k) 100 termites (l) 200 termites

(m) Parrots Original (n) 50 termites (o) 100 termites (p) 200 termites

(q) Lena Original (r) 50 termites (s) 100 termites (t) 200 termites

Figure 2: An example of filtering with number of steps constant ($N_s = 200$) and increasing number of termites ($k = 50, 100, 200$). On the top Texture, followed by Fruits, Parrots and Lena on the bottom.

(a) Parrots Original (b) Parrots TR

Figure 3: An example of filtering following the rule of thumb. Parrots with $k = 450, N_s = 450, \alpha = 0.1, \beta = 0.9, Q = 1$.

- $Q = 1$.

An example is shown in Figure 3.

As each pixel in each color channel can be recomputed independently, here we deliver multi-thread implementation, where the pixels of the image can be processed in parallel. We report here the time of processing for Lena image (512×512) exploiting a different number of threads with an increasing number of steps and constant termites with an increasing number of termites and constant steps, Figures 4 and 5 respectively. A gain of $\approx 6\times$ can be achieved in both cases. [1] We remind to the reader that choosing a number of threads higher than the core available will increase the context-switching resulting in a loss of performance, e.g., running Lena with $k = 200$, $N_s = 200$ and 32 threads will downgrade to $\approx 5\times$ in gain of speed.

Another approach to parallelism of TR could have been to process each termite of the swarm independently. After investigation, having a thread for each termite would lead to a strong overhead as each termite fundamentally execute a small amount of work that would get worse with a high usage of context-switching and lock mechanisms. Preliminary tests with Lena $k = 200, N_s = 200$ show that pixel-wise parallelism is $300\times$ faster than termite-wise parallelism and thus, the authors have considered this last approach inconvenient.

5 Conclusions

We have presented an implementation of Retinex based on of swarm intelligence for color and contrast enhancement of digital images.

Reconsidering the idea of the paths (the basis of the sampling approach of the original Retinex model) 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.

This algorithm named Termite Retinex has not the purpose of optimization of some constraints but an exploration of the image content tuned a small set of parameters, controlling the overall behavior of the swarm and therefore the final effect of the Retinex filtering.

[1] Tests performed on HP ZBook Power 15.6 inch G8 Mobile Workstation PC 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, 32.0 GB

Figure 4: Time with number of termites constant ($k = 200$) and increasing number of steps ($N_s = 50, 100, 200$) for 1, 2, 4, 8 threads. The computational time reported is the average of ten processing, and thus it may vary of $\pm 8\%$ at each execution.

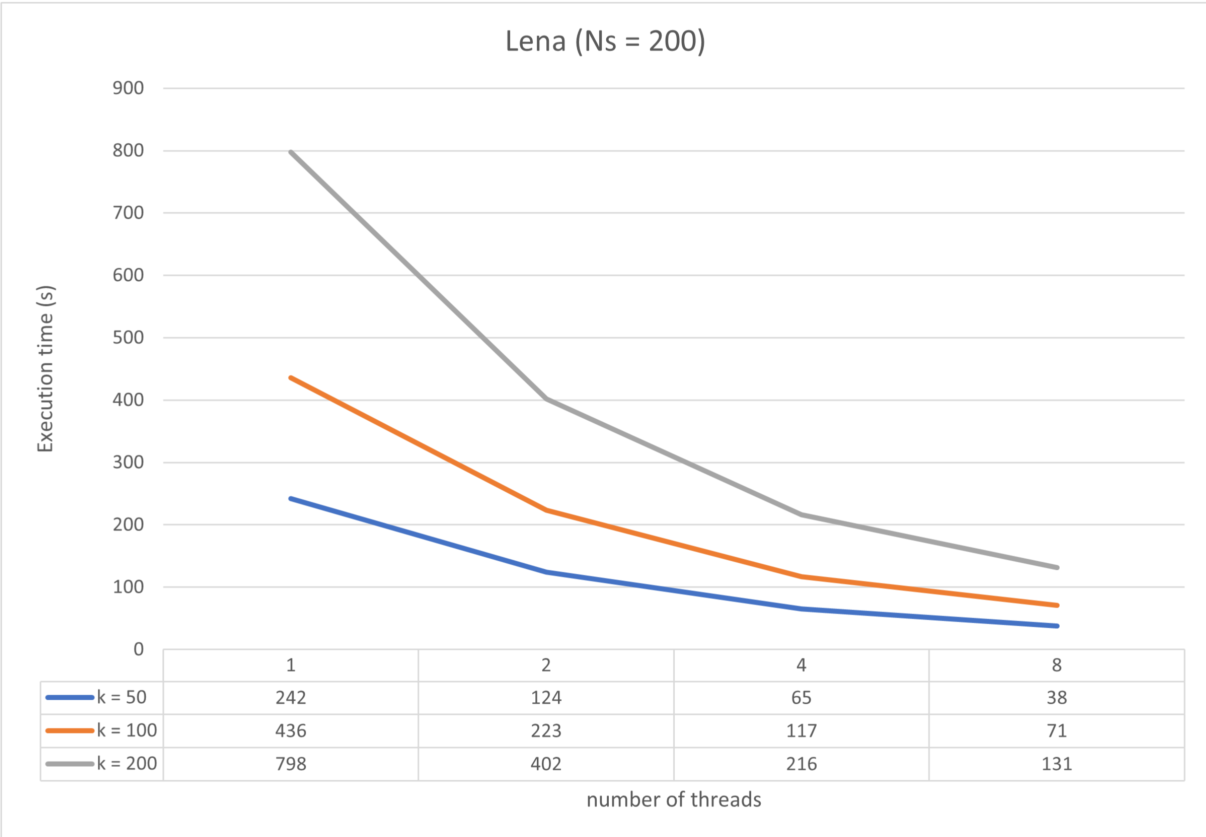


Figure 5: Time with number of steps constant ($N_s = 200$) and increasing number of termites ($k = 50, 100, 200$) for 1, 2, 4, 8 threads. The computational time reported is the average of ten processing, and thus it may vary of $\pm 8\%$ at each execution.

We have presented a detailed explanation of the swarm intelligence model used in Termite Retinex, and a discussion of its parameters in order to demonstrate the capability of unsupervised image enhancement, dynamic range stretching and color correction.

References

[1] J. Albers. *Interaction of Color: Unabridged Text and Selected Plates.* Yale University Press, 1975.

[2] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26, 1980.

[3] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: An autocatalytic optimizing process. Technical Report 91–016, Dipartimento di Elettronica e Informazione-Politecnico di Milano, Italy, 1991.

[4] M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1):29–41, 1996.

[5] M. Dorigo and T. Stützle. Ant Colony Optimization. MIT Press, Cambridge, MA, USA, 2004.

[6] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. *IEEE transactions on image processing*, 6(3):451–462, 1997.

[7] J. Kleinberg and E. Tardos. Algorithm design. Boston, MA, USA, 2005. Addison–Wesley Longman Publishing.

[8] D. Land. Visual perception by tom n. cornsweet. *Leonardo*, 5(4):360–361, 1972.

[9] E. H. Land. The Retinex. *American Scientist*, 52:247–64, 1964.

[10] E. H. Land. The Retinex theory of color vision. *Scientific American*, 237:108–128, 1977.

[11] E. H. Land and J. J. McCann. Lightness and Retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, jan 1971.

[12] M. Lecca and A. Rizzi. Tuning the locality of filtering with a spatially weighted implementation of random spray retinex. *JOSA A*, 32(10):1876–1887, 2015.

[13] J. J. McCann. Retinex at 40. *Journal of Electronic Imaging*, 13(1):6–7, 2004.

[14] J. J. McCann. Retinex at 50: color theory and spatial algorithms, a review. *Journal of Electronic Imaging*, 26(3):031204–031204, 2017.

[15] J. J. McCann and A. Rizzi. The art and science of HDR imaging. John Wiley, 2012.

[16] E. Provenzi, L. D. Carli, and A. Rizzi. Mathematical definition and analysis of the Retinex algorithm. *Journal of the Optical Society of America A*, 22(12):2613–2621, dec 2005.

[17] E. Provenzi, M. Fierro, A. Rizzi, L. De Carli, D. Gadia, and D. Marini. Random spray retinex: a new retinex implementation to investigate the local properties of the model. *IEEE Transactions on Image Processing*, 16(1):162–171, 2006.

[18] E. Provenzi, C. Gatta, M. Fierro, and A. Rizzi. A spatially variant white-patch and gray-world method for color image enhancement driven by local contrast. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1757–1770, 2008.

[19] A. Rizzi and C. Bonanomi. Milano retinex family. *Journal of Electronic Imaging*, 26(3):031207–031207, 2017.

[20] A. Rizzi et al. Designator retinex, milano retinex and the locality issue. In *Proceedings of IS&T International Symposium on Electronic Imaging 2016 Retinex at 50*, pages 1–5. Ingenta connect, 2016.

[21] A. Rizzi, C. Gatta, and D. Marini. A new algorithm for unsupervised global and local color correction. *Pattern Recognition Letters*, 24(11):1663–1677, 2003.

[22] A. Rizzi and J. J. McCann. On the behavior of spatial models of color. In R. Eschbach and G. G. Marcu, editors, *Color Imaging XII: Processing, Hardcopy, and Applications*, volume 6493 of *Proceedings of SPIE*, page 649302, San Jose, California, USA, Jan 2007.

[23] A. Rizzi, J. J. McCann, M. Bertalmío, and G. Gianini. Retinex at 50. *Special Issue on J. Electron. Imaging26*, 2017.

[24] J. G. Robson. Spatial and temporal contrast-sensitivity functions of the visual system. *Josa*, 56(8):1141–1142, 1966.

[25] G. Simone, G. Audino, I. Farup, F. Albregtsen, and A. Rizzi. Termite retinex: a new implementation based on a colony of intelligent agents. *Journal of electronic imaging*, 23(1):013006–013006, 2014.

[26] G. Simone, M. Lecca, G. Gianini, and A. Rizzi. Survey of methods and evaluation of retinex-inspired image enhancers. *Journal of Electronic Imaging*, 31(6):063055–063055, 2022.

[27] G. Simone, M. Lecca, G. Gianini, and A. Rizzi. Survey of methods and evaluation of Retinex-inspired image enhancers. *Journal of Electronic Imaging*, 31(6):063055, 2022.

[28] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the 6th International Conference on Computer Vision*, pages 839–846, Bombay, India, Jan 1998. IEEE Computer Society.