



Published in Image Processing On Line on 2020-06-02.
Submitted on 2020-04-24, accepted on 2020-05-29.
ISSN 2105-1232 © 2020 IPOL & the authors CC-BY-NC-SA
This article is available online with supplementary materials,
software, datasets and online demo at
<https://doi.org/10.5201/ipol.2020.300>

Local Contrast Enhancement based on Adaptive Logarithmic Mappings

Jose-Luis Lisani

Universitat Illes Balears, IAC3-IEEC (Spain)
joseluis.lisani@uib.es

Abstract

This article presents an implementation of the local image enhancement technique proposed in [J.L. Lisani, Adaptive local image enhancement based on logarithmic mappings, IEEE International Conference on Image Processing (ICIP), 2018]. The technique is based on a logarithmic mapping function that is adapted to the luminance characteristics of the neighborhood of each pixel. The method permits to simultaneously increase the luminance in dark regions and decrease it in bright regions. Variants of the originally published method are discussed in terms of contrast enhancement and creation of halo artifacts.

Source Code

The C source codes implementing the algorithms described in the paper, and the online demo, are accessible at [the associated web page](#)¹.

Keywords: image enhancement; contrast enhancement; local method; logarithmic mapping; adaptive method

1 Introduction

Many image enhancement methods have been proposed in the literature to improve the quality of poorly illuminated images. These methods can be roughly classified as global or local. Global techniques apply the same tone-mapping function to all the image pixels, independently of their position in the image. Examples of global techniques are gamma correction, histogram equalization, Drago et al. [3] method and [9]. The main shortcoming of these methods is that an increase in contrast in the dark regions leads to a decrease in contrast in the bright regions and vice versa, since they are unable to deal simultaneously with both kinds of regions.

Local techniques adapt the tone-mapping to the characteristics of the spatial neighborhood of each pixel and obtain better results. Examples of such techniques are CLAHE [14], Multiscale Retinex (MSR) [7, 13], ACE [15, 4] and Local Color Correction (LCC) [11, 16]. Most of these techniques

¹<https://doi.org/10.5201/ipol.2020.300>

successfully improve the visibility of poorly illuminated regions of the image, but in bright regions the contrast is not significantly improved (or even it is reduced). An exception to this behavior is LCC, which simultaneously enhances the contrast in both dark and bright regions.

Figure 1 illustrates the previous comments. The same input image is processed using a global enhancement method (Drago et al. [3]) and two local methods (Multiscale Retinex (MSR) [13] and LCC [11, 16]). We observe that neither Drago nor MSR methods are able to improve the contrast in the bright region of the image (the lighthouse). Quite to the contrary, the contrast is reduced with respect to the original image. Meanwhile LCC improves the visibility both in dark and bright regions.

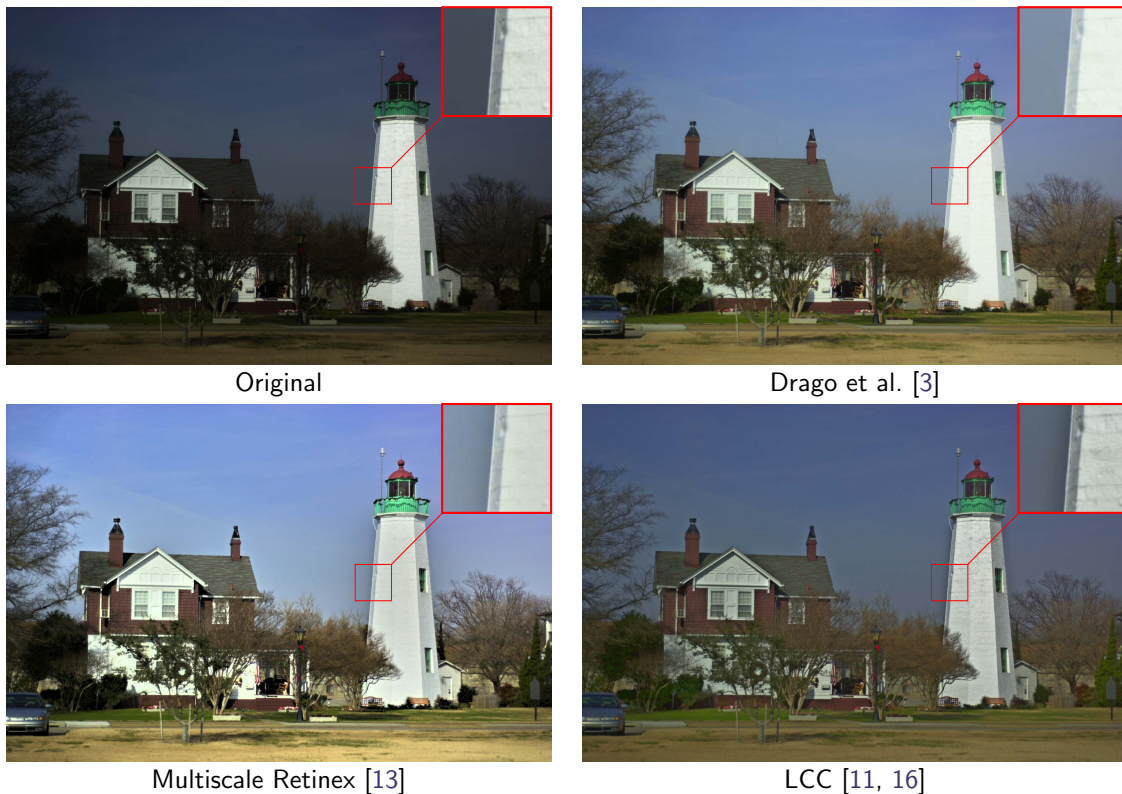


Figure 1: Original image and results of different color enhancement methods. The LCC method does not produce saturation in bright regions (in fact the contrast is improved with respect to the original), but creates halo artifacts.

Inspired by LCC, in [8] it was proposed a new local enhancement method which obtained better enhancement results than LCC in dark and bright image regions. The current paper describes this new method and analyzes its parameters and results.

The paper is organized as follows: Section 2 describes the method proposed in [8] and presents two variants to the original implementation. In Section 3 the behavior of the method in terms of contrast improvement and halo artifacts creation is analyzed for the three proposed implementations and for different values of their parameters. In Section 4 the results of the method are compared to the ones obtained with LCC and other techniques in the literature. Finally, some conclusions are exposed in the last section of the paper.

2 Description of the Method

The method presented in [8] is inspired by the work of L. Moroney in [11]. This author proposes to adapt the intensity of each pixel to the brightness characteristics of its neighborhood. In particular, pixels belonging to dark regions of the images are modified using a concave tone-mapping function

(see Figure 2-left) and therefore their values are increased. Conversely, a convex tone-mapping is applied to pixels in bright regions (see Figure 2-right), which produces a decrease of their values. The overall effect is an increase in visibility in both dark and bright regions. Moreney uses power functions (also called gamma correction functions) of the form I^α as tone mappings. Moreover, he assesses the brightness of the neighborhood of each pixel by convolving the intensity of the image with a Gaussian kernel. High (resp. low) values in the convolved image indicate that the pixel belongs to a bright (resp. dark) image region. The only parameter of the method is the scale of the Gaussian kernel. For color images the method is applied independently on each channel. A detailed description of Moroney's algorithm (called Local Color Correction, or LCC) can be found in [16] and Figure 1 shows an example of application.

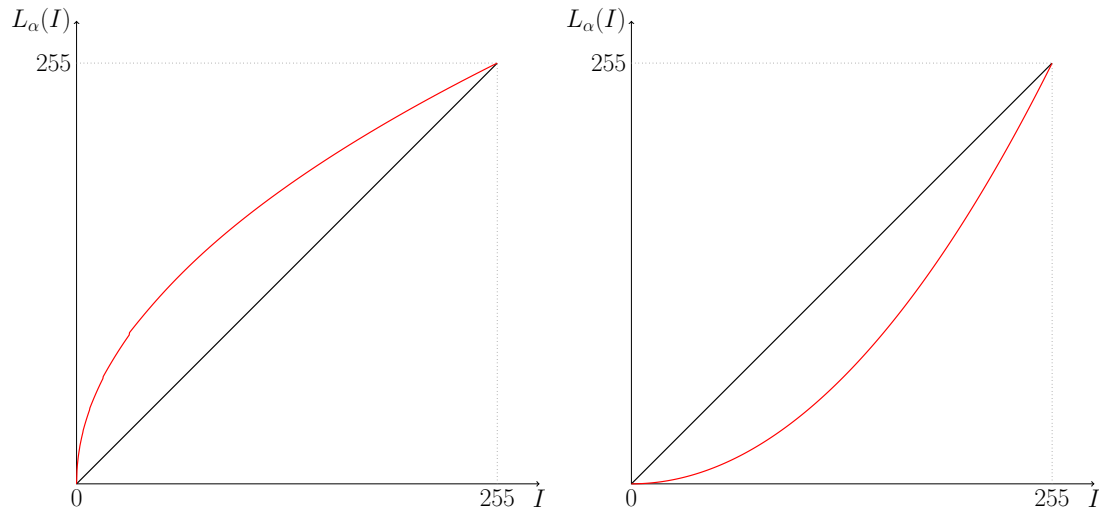


Figure 2: Left, example of concave mapping function (in this case $I^{0.5}$). Right, example of convex mapping function (in this case I^2). The black line represents the identity transform.

In [8] two modifications of LCC were proposed:

- Gamma correction functions were replaced by mappings based on logarithmic functions (see below for details), which produce a steepest increase (resp. decrease) for low (resp. high) values of intensity, thus leading to higher contrast enhancement in dark and bright image regions.
- The use of an anisotropic filter [1, 10], instead of the isotropic Gaussian kernel, for the assessment of the brightness in the pixel neighborhood. The goal here was the reduction of halo artifacts in the processed images, along the boundaries between dark and bright regions. The rationale behind this change is that halos are produced because the Gaussian filtering blurs the boundaries between these regions, and therefore by using a boundary-preserving anisotropic filter, halos should be reduced.

In particular, given an input intensity image² $I : \Omega \rightarrow [0, 255]$, the new output I' is computed as

$$I'(\mathbf{x}) = L_{\alpha(w(\mathbf{x}))}(I(\mathbf{x})). \quad (1)$$

This equation has three components: a mapping function $L_\alpha(I)$, a weight map $w(\mathbf{x})$, and a transition function $\alpha(t)$.

²In a pre-processing step the range of the input image is linearly mapped to the full range of an 8-bit image, i.e. $[0, 255]$

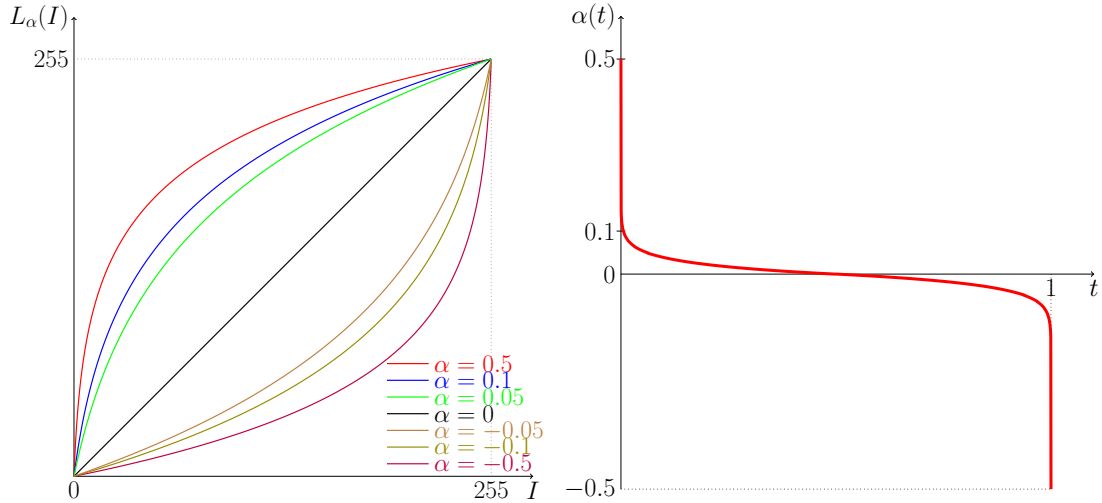


Figure 3: Left, tone mapping function $L_\alpha(I)$, as defined by Equation (2), for various values of α . Right, function $\alpha(t)$, as defined by Equation (6), for $\gamma = 0.05$.

The mapping function L_α represents in fact a family of tone mappings, based on the logarithmic function and parameterized by a single parameter α . It is defined as follows

$$L_\alpha(I) = \begin{cases} 255 \frac{\log(\alpha I + 1)}{\log(255\alpha + 1)}, & \text{if } \alpha > 0 \\ I, & \text{if } \alpha = 0 \\ 255(1 - \frac{\log(|\alpha|(255 - I) + 1)}{\log(255|\alpha| + 1)}), & \text{if } \alpha < 0 \end{cases} \quad (2)$$

Figure 3-left displays $L_\alpha(I)$ for different values of the parameter α . Remark that, for positive values of α , L_α is a classical logarithmic tone mapping that shall increase the contrast of dark regions of the image, as commented above. For $\alpha = 0$ we have the identity transform, and for $\alpha < 0$ we get an inverted logarithmic function that shall increase the contrast of bright regions.

The function $w(\mathbf{x})$ is called a *weight map*, and assesses the brightness in the neighborhood of pixel \mathbf{x} . In this paper we test three different alternatives for $w(\mathbf{x})$:

1. Gaussian weight map. As in [11, 16] the weight map is computed as the convolution of the intensity image with a 2D Gaussian kernel

$$w(\mathbf{x}) = I(\mathbf{x}) * G_\sigma(\mathbf{x}), \quad \text{with} \quad G_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\mathbf{x}\|_2^2}{2\sigma^2}\right), \quad (3)$$

where σ is the scale of the kernel, and $\|\cdot\|_2$ denotes the L^2 norm.

2. Mean Curvature Motion (MCM) weight map. This is the option used in [8]. It consists of an anisotropic filtering of the image intensity using the following PDE

$$w(\mathbf{x}) = \frac{\partial I(\mathbf{x})}{\partial t} = D^2 I \left(\frac{DI^\perp}{|DI|}, \frac{DI^\perp}{|DI|} \right), \quad (4)$$

where D denotes the gradient operator. This is an evolution equation that progressively modifies the initial image values depending on the curvature of their associated level lines. Details about this equation can be found in [10]. We use the implementation described in [16]. In principle, the only parameter of this filter is the *normalized scale* R (at normalized scale R a disk of radius R disappears). However, in the used implementation an additional parameter is added, namely a threshold T_g on the magnitude of the gradient. For pixels whose gradient

magnitude is below T_g , the heat equation³, which is equivalent to a Gaussian convolution, is applied instead of Equation (4).

3. Bilateral weight map. As an alternative to the anisotropic filter proposed in [8] to reduce the halo effect, we propose in the current paper the use of a bilateral filter to compute the weight map

$$w(\mathbf{x}) = \frac{1}{W} \sum_{\mathbf{y} \in \Omega} I(\mathbf{y}) \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{2\sigma_S^2}\right) \exp\left(-\frac{(I(\mathbf{x}) - I(\mathbf{y}))^2}{2\sigma_I^2}\right), \quad (5)$$

with W a normalizing factor defined as

$$W = \sum_{\mathbf{y} \in \Omega} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{2\sigma_S^2}\right) \exp\left(-\frac{(I(\mathbf{x}) - I(\mathbf{y}))^2}{2\sigma_I^2}\right).$$

The filter depends on two parameters, a spatial scale parameter σ_S and a range parameter σ_I .

After computation with one of the above formulas, the values of the weight map are normalized in the range $[0, 1]$. Figure 4 shows how the weight map of an image looks like for the different filtering options. Remark that both the MCM and the bilateral filtering converge to the Gaussian blur when threshold T_g and the range parameter, respectively, are large enough. In the next section we will compare the results obtained with the different definitions of the weight function, and we will analyze the effect of the variation of its parameter(s).

Finally, a function $\alpha(t)$ is defined that assigns to each value t in the weight map a value of the parameter α

$$\alpha(t) = \begin{cases} 0.5(1 - (\frac{t}{0.5})^\gamma), & \text{if } t \leq 0.5 \\ -0.5(1 - (\frac{1-t}{0.5})^\gamma), & \text{if } t > 0.5 \end{cases}, \quad (6)$$

where γ is a parameter that we set to 0.05 to guarantee a smooth transition between the different values of α (see Figure 3-right). This function is applied on the values of the weight map. As a result, positive α values (i.e. logarithmic tone mappings) are assigned to dark regions of the image, while negative α 's (i.e. inverted logarithmic tone mappings) are assigned to bright regions. The final result of Equation (2) is an image with increased contrast in both dark and bright regions.

Color processing. For color images Equation (1) may be applied independently to each channel. However, this leads to chrominance changes with respect to the original image due to the modification of the original R/G/B ratios. One way to preserve these ratios is to apply Equation (1) to the luminance component of the image (i.e. the average of R, G and B values), and then to compute the new RGB values as

$$R' = R \left(\frac{I'}{I}\right), \quad G' = G \left(\frac{I'}{I}\right), \quad B' = B \left(\frac{I'}{I}\right) \quad (7)$$

where R, G and B (resp. R', G', B') are the color channels of the original (resp. output) image and I (resp. I') its luminance. Remark that by using this transformation the original R/G/B ratios are kept. Moreover, the luminance of the output color image is equal to I' .

The whole method, which we call LogLocal Color Correction (or LLCC for short), is summarized in Algorithm 1.

³ $\frac{\partial I(\mathbf{x})}{\partial t} = \Delta I$, where Δ is the Laplacian operator.

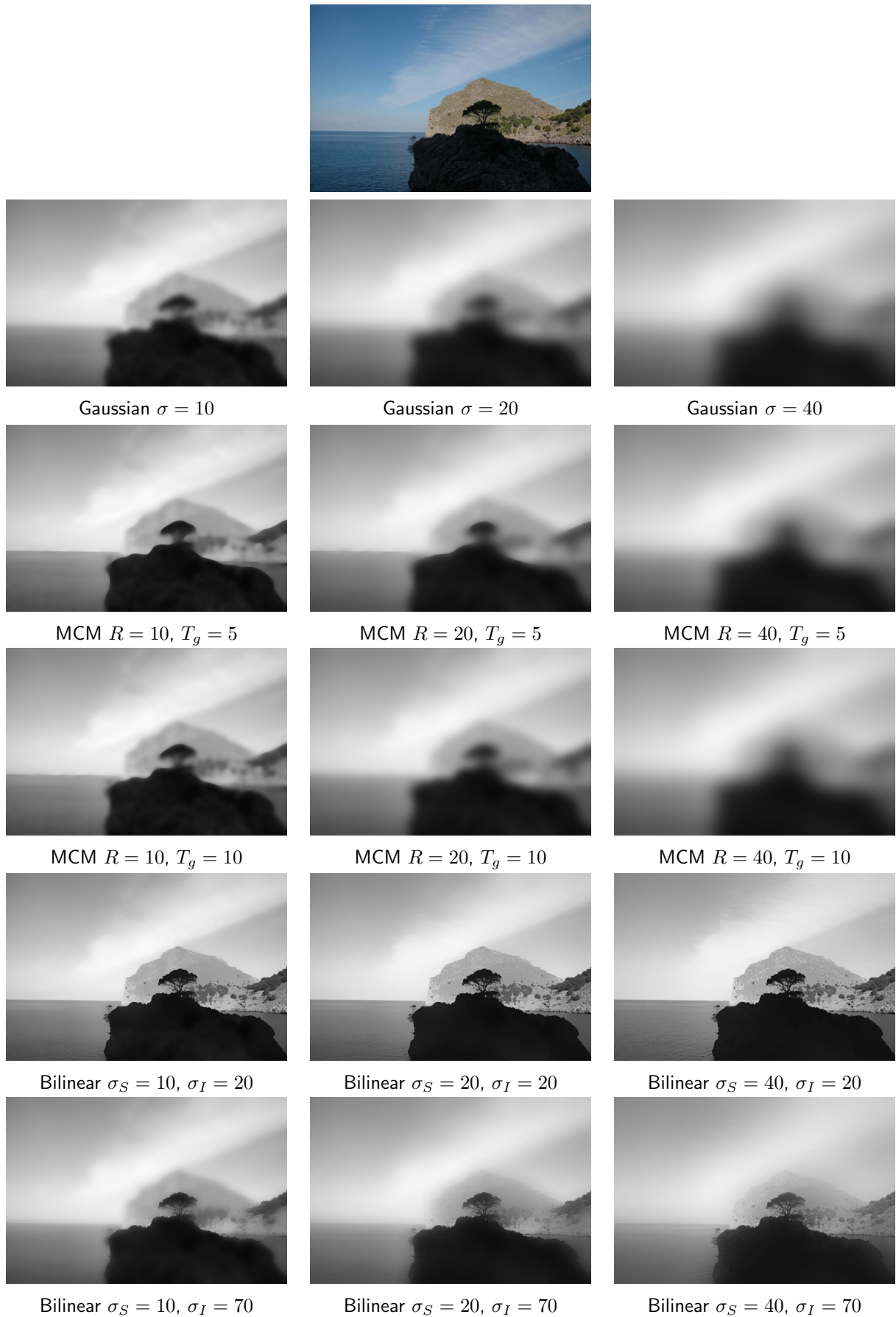


Figure 4: Top, original image. Second row, weight maps obtained by filtering the image intensity with a Gaussian kernel for increasing values of σ . Third and fourth rows, weight maps obtained by filtering the image intensity with a MCM filter for increasing values of R and two different values of threshold T_g . Last two rows, weight maps obtained by filtering the image intensity with a bilateral filter for increasing values of σ_S and two different values of σ_I .

Algorithm 1: LogLocal Contrast Enhancement

```

Input      : input image  $u = (u_R, u_G, u_B)$  (image domain:  $\Omega$ )
Output    : processed image  $v$ 
Parameters: filtering option (Gaussian or mcm or bilateral), scale parameter  $\sigma$  (if
                filtering=Gaussian or mcm) or scale and range parameters,  $\sigma_S$  and  $\sigma_I$  (if
                filtering=bilateral), gradient magnitude threshold  $T_g$  (if filtering=mcm)

//Compute intensity
1  $I = \text{averageRGBchannels}(u)$ 
  //Stretch range: linear map from  $[\min(I), \max(I)]$  to  $[0, 255]$ 
2  $I = \text{linearStretch}(I)$ 
  //Compute weight map: normalize range of  $I$  and filter parameters to  $[0,1]$  and apply
  filter
3 if filtering option == Gaussian then
4    $w = \text{gaussian\_filter}(I/255, \sigma)$  //Equation (3)
5 if filtering option == mcm then
6    $w = \text{mcm\_filter}(I/255, T_g/255, \sigma)$  //Equation (4)
7 if filtering option == bilateral then
8    $w = \text{bilateral\_filter}(I/255, \sigma_S, \sigma_I/255)$  //Equation (5)
  //Locally apply logarithmic mappings
9 for  $\mathbf{x} \in \Omega$  do
10   $I'(\mathbf{x}) = L_{\alpha(w(\mathbf{x}))}(I(\mathbf{x}))$  //Use Equations (2) and (6)
  //Compute output color image
11 for  $\mathbf{x} \in \Omega$  do
12   $(v_R, v_G, v_B)(\mathbf{x}) = (u_R, u_G, u_B)(\mathbf{x}) \cdot \frac{I'(\mathbf{x})}{I(\mathbf{x})}$  //Values outside  $[0, 255]$  are clipped
    
```

2.1 Implementation Details

The algorithm complexity is dominated by the filtering operation used to compute the weight map. Once the weight map has been obtained, the rest of the algorithm runs in linear time. In order to reduce the computation time we have used fast implementations of the three proposed types of filtering. For the Gaussian convolution, we have used the SII (stacked integral images) approximation described in [5], which is the fastest of all the implementations analyzed in this article. Moreover, its computational cost does not depend on the scale of the Gaussian kernel. The MCM filter implementation has been obtained from [10]. Finally, a fast implementation of the bilateral filter due to Sylvain Paris and Frédo Durand [12] has been downloaded from <http://people.csail.mit.edu/sparis/bf/>. In particular, we use the truncated kernel implementation, which gives an approximate result but it is faster than the full kernel version.

The following table shows the computation times for the image in Figure 5-left (2000×1312 pixels) after running the algorithm for the three types of filtering. The scale of the Gaussian kernel was 20, the normalized scale of the MCM filter was 20 and the gradient threshold parameter 10, and the scale and range parameters of the bilateral filter were 20 and 70, respectively. The code was run on a single-thread of an Intel Core i5-4460 processor (4 cores, at 3.2GHz) with 16GB of RAM.

We observe that when using MCM filtering the algorithm is much slower than with the other types of filtering. In general we can conclude that, in terms of computation time and quality of the final result (see next section), it is better to use the bilateral filter for the computation of the weight map.

Gaussian	MCM	Bilateral
1.31	15.86	1.43

Table 1: Computation times (seconds) of the algorithm for image in Figure 5-left (2000×1312 pixels) and different types of filtering.

3 Experiments

In this section we compare the results of the method obtained with the different options for the computation of the weight map and with different values of the parameters. Figure 5 shows the two input images used in our tests.

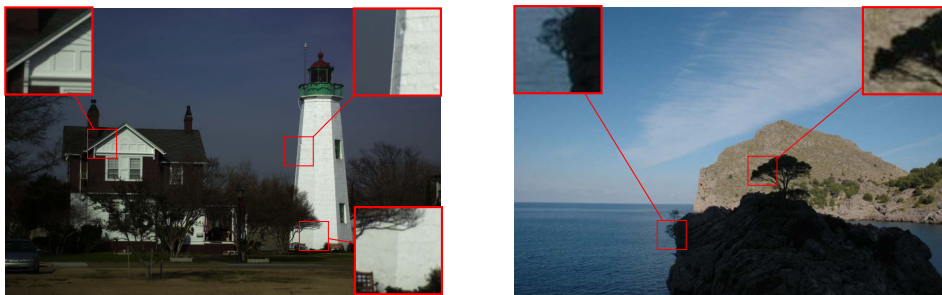


Figure 5: Original images used in the experiments of Section 3.

The results obtained with a Gaussian weight map and different values of the scale parameter are shown in Figures 7 and 8. We observe that a halo effect is visible where very bright or very dark regions meet with a uniform background. We can see it for example on the side of the lighthouse in Figure 7. In this case the halo is dark. On the other hand we observe a bright halo on the side of the rocky outcrop, and also around the tree, in Figure 8. This halo is more conspicuous for large values of the scale parameter. Moreover, remark that the image is less contrasted when σ is small. When the Gaussian weight map is used the only difference between LLCC and LCC is the local mapping function (a logarithmic function vs. a power function). It must be noted that in this case, and for the same scale parameter, the halo effect is more visible for the LLCC method (see Figure 6). The reason is that the same property of the logarithmic mappings (i.e. steepest increase/decrease for low/high intensity values) that permits a higher enhancement of the image contrast also makes more visible the halo.



Figure 6: Comparison between LCC and LLCC when using the same weight map (Gaussian) and scale parameter ($\sigma = 20$). The LLCC result is more contrasted, but the halo effect is more conspicuous.



Figure 7: Results of LLCC when using a weight map obtained by convolution with a Gaussian kernel of increasing scale σ .

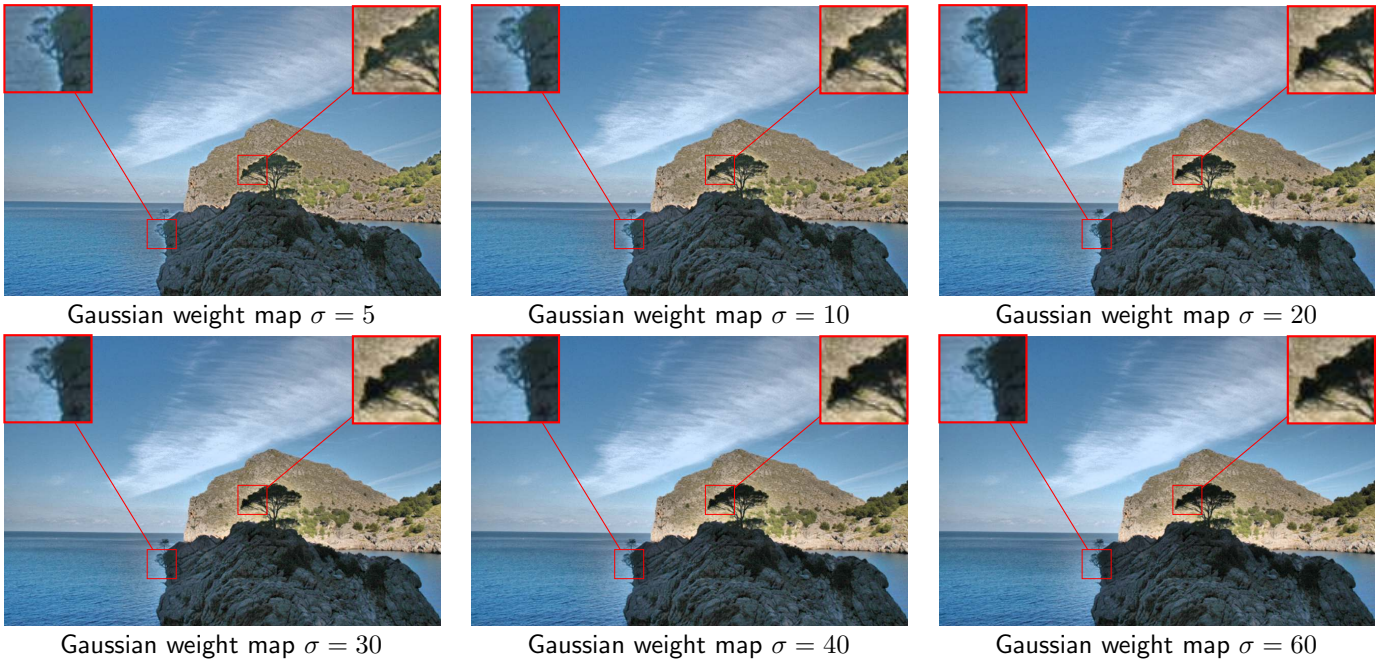


Figure 8: Results of LLCC when using a weight map obtained by convolution with a Gaussian kernel of increasing scale σ .

For the MCM weight map we show the results for two different values of the gradient magnitude threshold in Figures 9 and 10. For larger values of this threshold the results are very similar to the ones obtained with the Gaussian weight map. As with the Gaussian weight map, the halo effect and the contrast increase with the scale parameter, for a fixed value of the gradient threshold. Moreover, if the scale is fixed and the gradient threshold is increased, the halo becomes more visible and the contrast between bright and dark regions also increases. When the threshold parameter is small some artifacts may appear in the image due to the curved shapes of the weight map typical of the curvature filtering (see images in the third row of Figure 4, for $T_g = 5$). This effect can be observed in the top-left detail of the image computed with parameters $R = 10$ and $T_g = 5$ in Figure 9. The

use of a larger threshold value permits to eliminate this artifact. Although the scale parameters of the Gaussian and MCM filter have a different interpretation, when using the same value of the parameter, the halo is less visible in the case of an MCM weight map. In order to better appreciate the differences between these results we recommend the reader to run the online demo associated to this article.

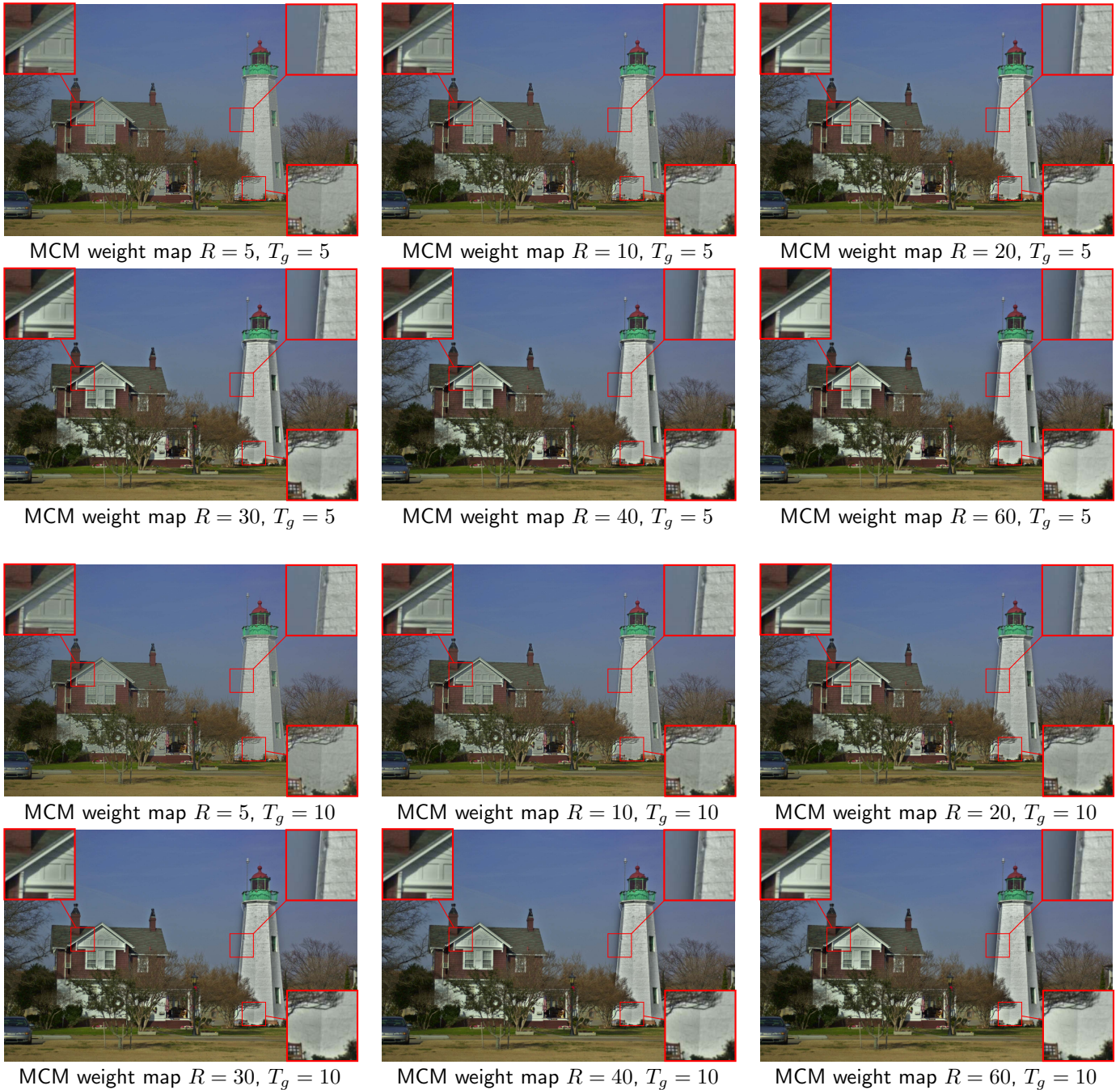


Figure 9: Results of LLCC when using a weight map obtained by filtering with MCM of increasing scale R and two different values of the threshold T_g .

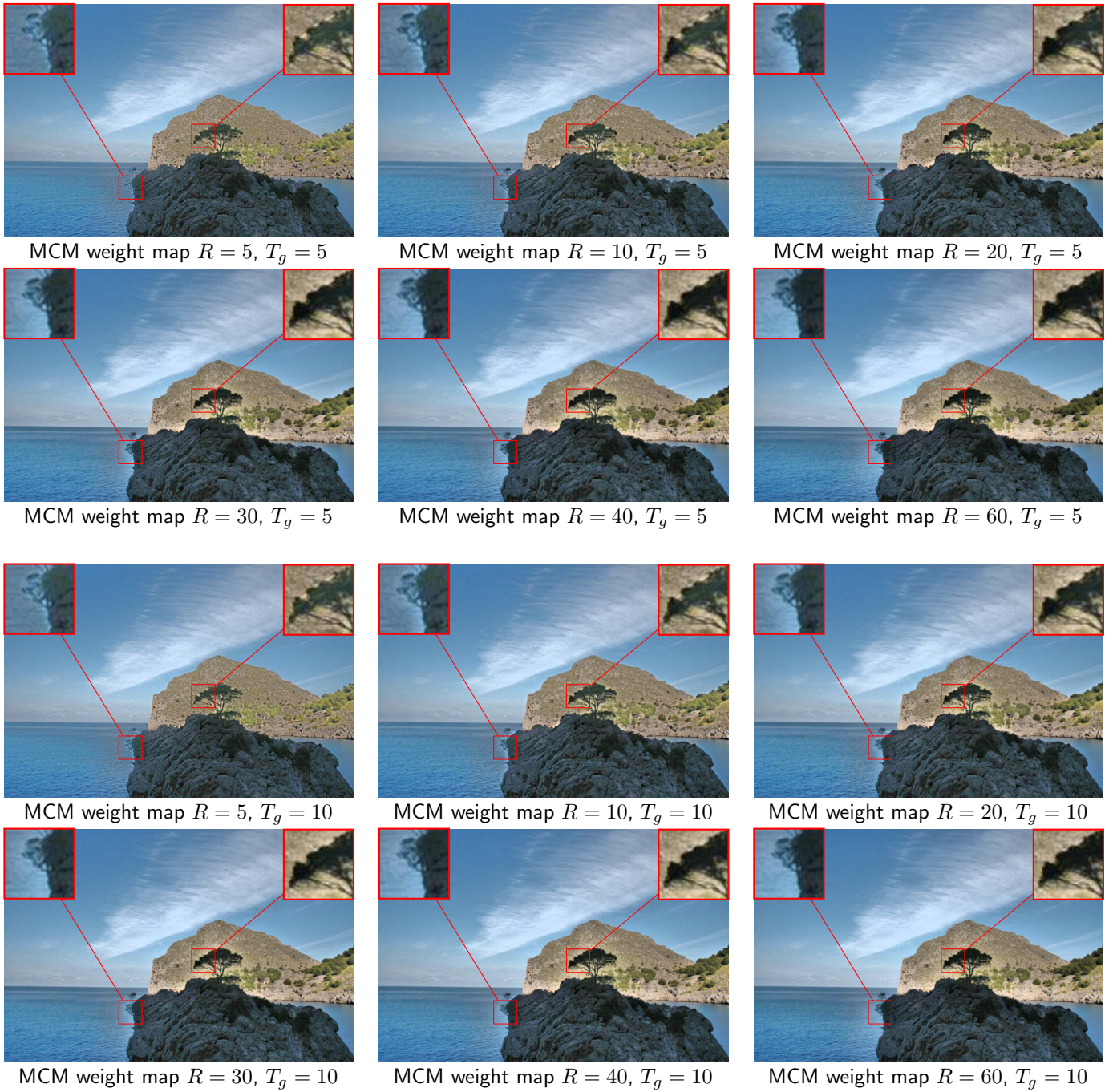


Figure 10: Results of LLCC when using a weight map obtained by filtering with MCM of increasing scale R and two different values of the threshold T_g .

For the bilateral weight map Figures 11 and 12 display the results obtained for increasing values of the scale parameter and two different values of the range parameter. For larger values of the range parameter the obtained results are similar to the ones obtained with the Gaussian weight map. For small values of this parameter no halos are created. However some details of the images are lost, such as the branches of the trees in the bottom-right detail of Figure 11 and in the top-left detail of Figure 12 (in both cases for $\sigma_I = 20$).



Figure 11: Results of LLCC when using a weight map obtained by bilateral filtering of increasing scale σ_S and two different values of the range scale σ_I .

This artifact occurs when small dark (resp. bright) objects are in front of a slightly brighter (resp. darker) background. In this case, when the range parameter is small, the weight map takes a low value at the dark pixels (which are therefore brightened) and a high value at the bright ones (which are darkened). The overall effect is that both types of pixels may end up with similar values, as

in the displayed examples. By using a larger value of the range parameter a more uniform weight map is obtained and approximately the same mapping is applied to all of them, thus preserving (or increasing) the original differences in luminance. In this case, only when the differences in luminance are large the values of the weight map shall be different and the contrast in both dark and bright regions will be enhanced. This can be seen in the images obtained with $\sigma_I = 70$. In these images we also observe that when the scale parameter increases a slight halo effect is introduced and the image contrast increases.

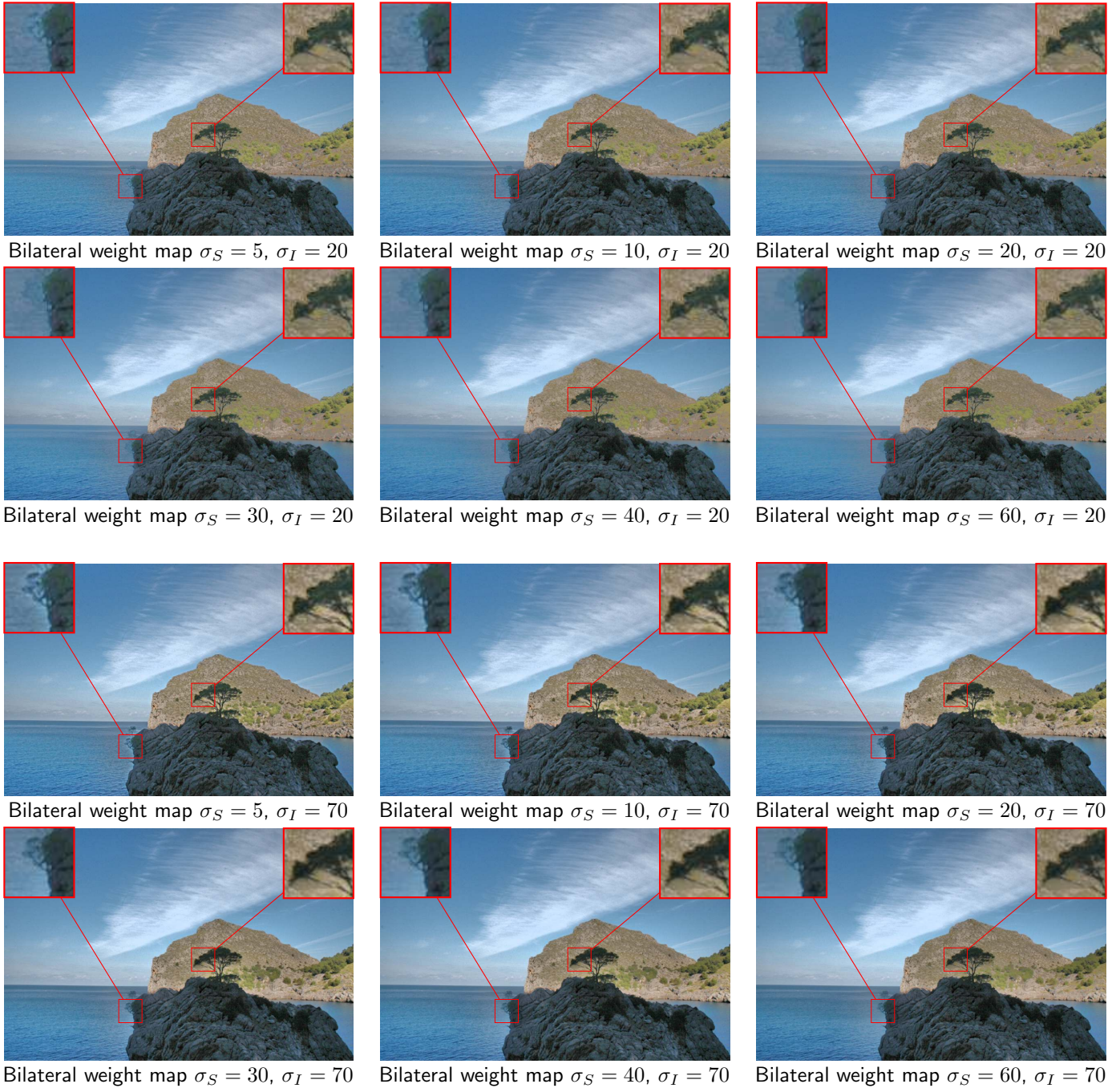


Figure 12: Results of LLCC when using a weight map obtained by bilateral filtering of increasing scale σ_S and two different values of the range scale σ_I .

From the previous experiments we can conclude that, in order to get rid of halo artifacts, the best option is to use the bilateral filter for the computation of the weight map. Moreover, to avoid the introduction of new artifacts using this filter, the range parameter cannot be too small and the scale parameter cannot be too large. A safe option is to use the default parameters $\sigma_S = 5$ and $\sigma_I = 70$, which shall be used in the experiments of the next section. Remark however that halo artifacts are generally only noticeable if the image presents large uniform areas (e.g. the sky or the sea) that meet some very dark or very bright image object, as in the examples shown in the previous images. If this is not the case, the use of a larger scale parameter may be advisable in order to obtain a more contrasted result. This is illustrated in Figure 13 where the results obtained with the three proposed filtering techniques are shown and no artifacts are noticeable. In this example, the use of a Gaussian weight map permits to obtain a more contrasted image than with the other filtering methods.

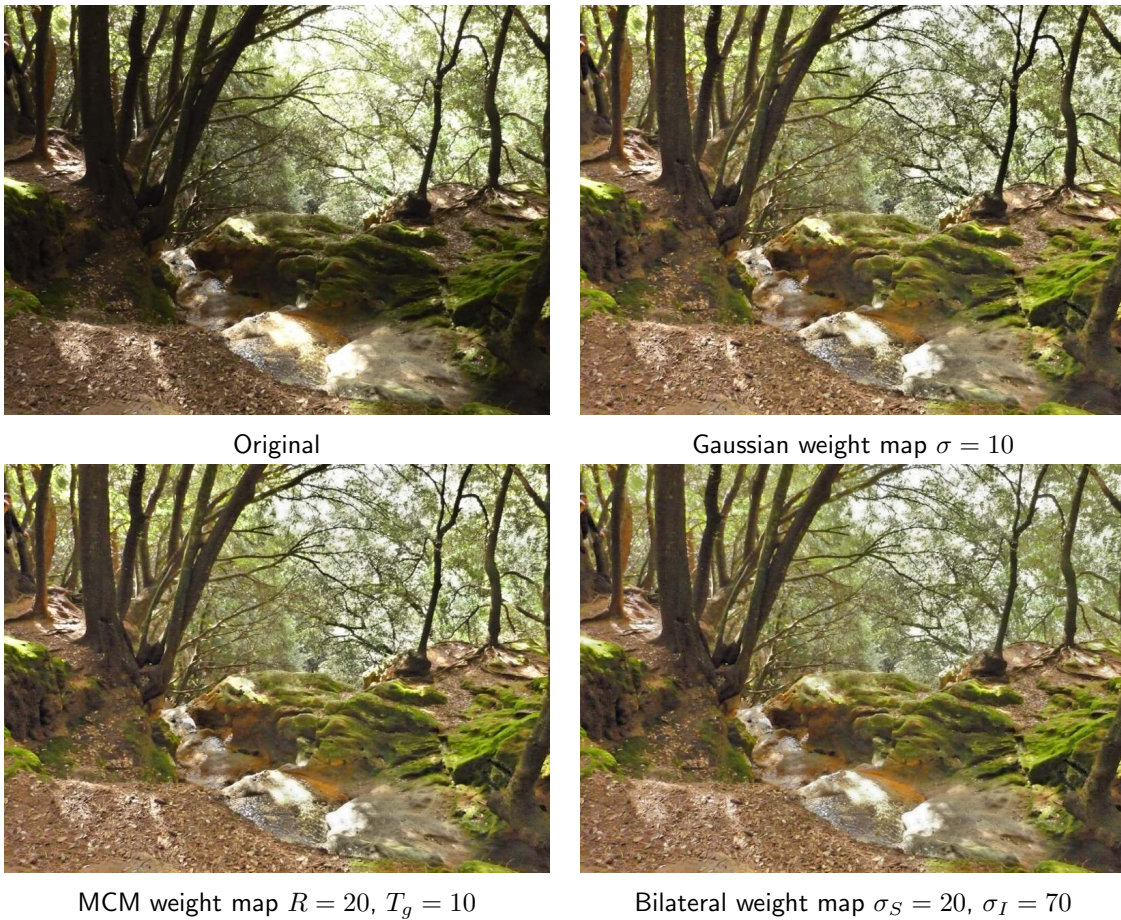


Figure 13: Example of enhancement using a large value of the scale parameter. For this image no noticeable artifacts can be observed.

4 Comparison with other Methods

In this section the results of LLCC are compared to those obtained with other image enhancement techniques. First of all, the method is compared to LCC, to show that it indeed improves its results, which was the primary goal of its implementation. Moreover, two recently published techniques are included in the comparison, namely BIE [2] and SEF [6]. Both techniques are aimed at improving simultaneously dark and bright regions of the image and follow a similar approach: several enhanced versions of the original image are combined using a fusion technique. For the comparison, and according to the conclusions of the previous section, we use the version of LLCC that computes the weight map with a bilateral filter and parameters $\sigma_I = 70$ and $\sigma_S = 5$. LCC, BIE and SEF results have been computed using their associated online demos⁴.

For LCC we use the default scale parameter of the demo (10% of the smallest dimension of the image). For BIE and SEF we use the default parameters of the demo, except for the parameters of the robust normalization of SEF. By default, this normalization implies clipping 1% of the most saturated pixels of the image (both dark and bright). In order to do a fairer comparison with the other methods we have disabled the clipping by setting the parameter of the normalization to 0%, which simply means that a linear mapping is applied so that the minimum and maximum values of the output image are set to 0 and 255, respectively.

Figure 14 shows the results of the comparison on five images. We can observe that all the methods are able to enhance the visibility in both dark and bright areas. In dark areas the four methods give similar results, although BIE and SEF offer a slightly better performance. However, in bright areas LLCC is the method that obtains better results, improving the contrast while avoiding oversaturation, as can be seen in the details shown in Figure 15. On the negative side, LLCC produces a halo artifact in the image displayed in the second row of both figures. This halo can be observed around the tree on the right side of the image in Figure 15.

5 Conclusions

This paper describes an image enhancement technique (LLCC) based on a family of logarithmic tone mappings which are locally applied to each image pixel depending on the luminance characteristics of its neighborhood. Three different alternative implementations of the method, and their parameters, are analyzed and the results are compared with those of state of the art techniques. The experiments show that LLCC is able to improve the visibility in both dark and bright image regions and, in particular, is the method that produces less saturated results in bright regions, at the expense of introducing halo artifacts in some images.

⁴LCC: http://demo.ipol.im/demo/gl_localcolorcorrection/,
 BIE: <http://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000027>,
 SEF: <http://ipolcore.ipol.im/demo/clientApp/demo.html?id=279>



Figure 14: Comparison between LLCC and other image enhancement methods. From top to bottom, images Iris, Es Canonge, Swan, Cathedral and Girl.

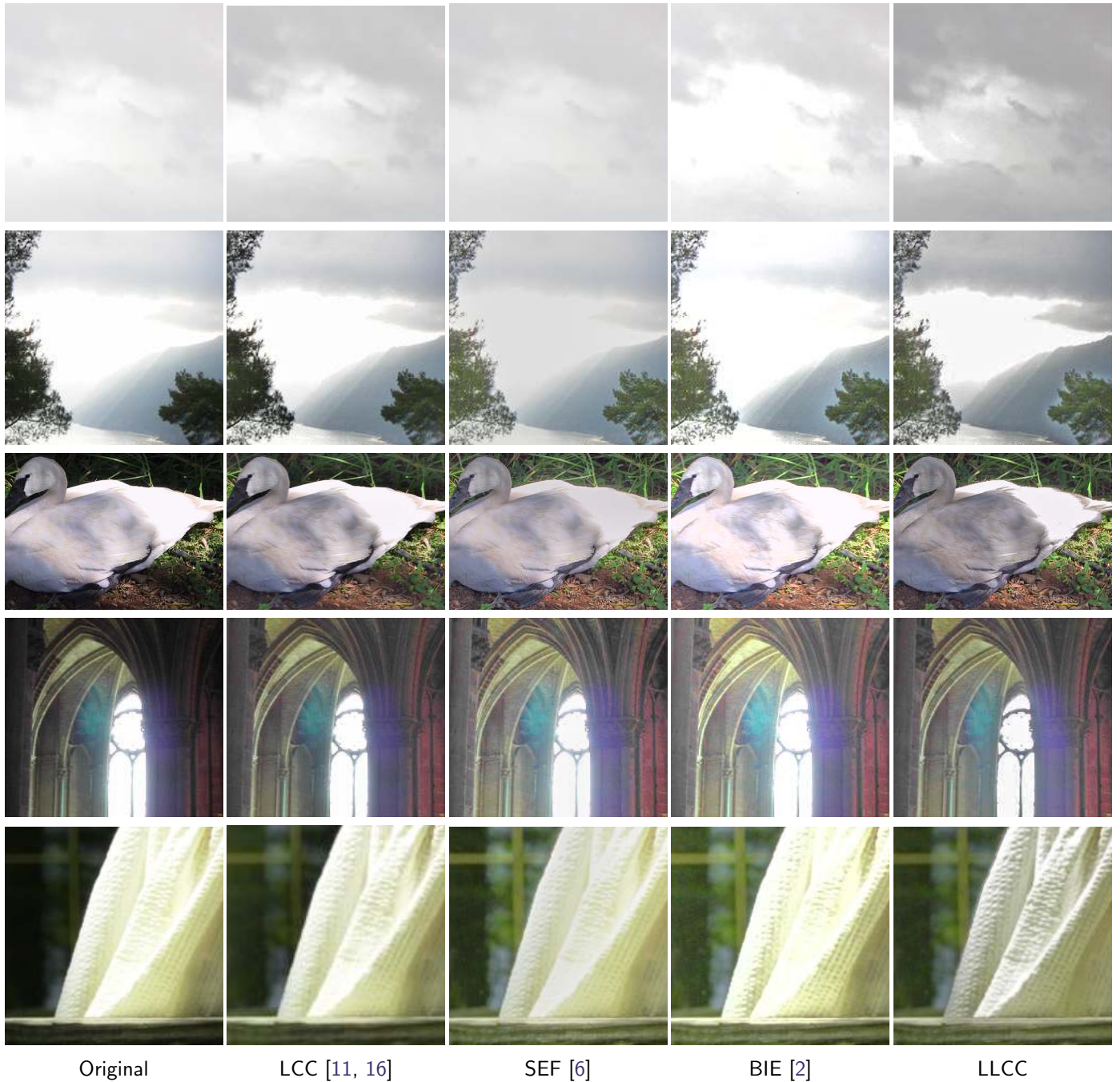


Figure 15: Details of the comparison between LLCC and other image enhancement methods. From top to bottom, images Iris, Es Canonge, Swan, Cathedral and Girl.

Acknowledgment

The author was partially supported by MINECO/AEI/FEDER, UE projects TIN2017-85572-P and DPI2017-86372-C3-3-R.

Image Credits



NASA <https://dragon.larc.nasa.gov/retinex/pao/news/>



CC-BY Juan Gabriel Gomila



Jean-Michel Morel



CC-BY Ana Belén Petro



Unknown author

References

- [1] L. ALVAREZ, F. GUICHARD, P-L. LIONS, AND J-M. MOREL, *Axioms and fundamental equations of image processing*, Archive for Rational Mechanics and Analysis, 123 (1993), pp. 199–257.
- [2] A. BUADES, J.L. LISANI, A.B. PETRO, AND C. SBERT, *Backlit images enhancement using global tone mappings and image fusion*, IET Image Processing, 14 (2019), pp. 211–219. <https://doi.org/10.1049/iet-ipr.2019.0814>.
- [3] F. DRAGO, K. MYSZKOWSKI, T. ANNEN, AND N. CHIBA, *Adaptive logarithmic mapping for displaying high contrast scenes*, in Proceedings of EUROGRAPHICS, vol. 22, 2003. <https://doi.org/10.1111/1467-8659.00689>.
- [4] P. GETREUER, *Automatic Color Enhancement (ACE) and its Fast Implementation*, Image Processing On Line, 2 (2012), pp. 266–277. <https://doi.org/10.5201/ipol.2012.g-ace>.
- [5] —, *A Survey of Gaussian Convolution Algorithms*, Image Processing On Line, 3 (2013), pp. 286–310. <https://doi.org/10.5201/ipol.2013.87>.
- [6] C. HESSEL, *Simulated Exposure Fusion*, Image Processing On Line, 9 (2019), pp. 469–482. <https://doi.org/10.5201/ipol.2019.279>.
- [7] D. J. JOBSON, Z. RAHMAN, AND G. A. WOODSELL, *A multiscale retinex for bridging the gap between color images and the human observation of scenes*, IEEE Transactions on Image Processing, 6 (1997), pp. 965–976. <https://doi.org/10.1109/83.597272>.
- [8] J.L. LISANI, *Adaptive local image enhancement based on logarithmic mappings*, in IEEE International Conference on Image Processing (ICIP), 2018. <https://doi.org/10.1109/ICIP.2018.8451655>.
- [9] J.L. LISANI, A.B. PETRO, AND C. SBERT, *Color and Contrast Enhancement by Controlled Piecewise Affine Histogram Equalization*, Image Processing On Line, 2 (2012), pp. 243–265. <https://doi.org/10.5201/ipol.2012.lps-pae>.

- [10] M. MONDELLI AND A. CIOMAGA, *Finite Difference Schemes for MCM and AMSS*, Image Processing On Line, 1 (2011), pp. 127–177. https://doi.org/10.5201/ipol.2011.cm_fds.
- [11] N. MORONEY, *Local color correction using non-linear masking*, in IS&T/SID Eight Color Imaging Conference, 2000, pp. 108–111.
- [12] S. PARIS AND F. DURAND, *A fast approximation of the bilateral filter using a signal processing approach*, in European Conference on Computer Vision (ECCV), 2006. https://doi.org/10.1007/11744085_44.
- [13] A.B. PETRO, C. SBERT, AND J-M. MOREL, *Multiscale Retinex*, Image Processing On Line, (2014), pp. 71–88. <https://doi.org/10.5201/ipol.2014.107>.
- [14] S.M. PIZER, E.P. AMBURN, J.D. AUSTIN, R. CROMARTIE, A. GESELOWITZ, T. GREER, B. TER HAAR ROMENY, J.B. ZIMMERMAN, AND K. ZUIDERVELD, *Adaptive histogram equalization and its variations*, Computer Vision, Graphics, and Image Processing, 39 (1987), pp. 355–368. [https://doi.org/10.1016/S0734-189X\(87\)80186-X](https://doi.org/10.1016/S0734-189X(87)80186-X).
- [15] A. RIZZI, C. GATTA, AND D. MARINI, *A new algorithm for unsupervised global and local color correction*, Pattern Recognition Letters, 24 (2003), pp. 1663 – 1677. [https://doi.org/10.1016/S0167-8655\(02\)00323-9](https://doi.org/10.1016/S0167-8655(02)00323-9).
- [16] J.G. GOMILA SALAS AND J.L. LISANI, *Local Color Correction*, Image Processing On Line, 1 (2011), pp. 260–280. https://doi.org/10.5201/ipol.2011.gl_lcc.