



Published in Image Processing On Line on 2018-12-07.
Submitted on 2018-11-14, accepted on 2018-12-06.
ISSN 2105-1232 © 2018 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.2018.236>

An Analysis and Implementation of the Shape Preserving Local Histogram Modification Algorithm

Jose-Luis Lisani

Universitat Illes Balears, Spain
joseluis.lisani@uib.es

Communicated by Jean-Michel Morel and Miguel Colom

Demo edited by Jose-Luis Lisani

Abstract

In this paper we describe the implementation of the algorithm for local contrast enhancement published by Caselles et al. in 1999. This algorithm was the first designed explicitly to increase the contrast while preserving the so-called “shape structure” of the image, that is, its set of level sets. According to the mathematical morphology school, artifacts are created when this structure is modified. The original algorithm is described and also two alternative implementations are proposed, which limit the over-enhancement of noise.

Source Code

The reviewed source code and documentation for this algorithm are available from [the web page of this article](#)¹.

Supplementary Material

Most of the figures displayed in this article can be visualized at the following web page, provided as supplementary material: <http://www.ipol.im/pub/art/2018/236/supplementary/>. The display format of this page permits an easy comparison between the results of the different methods and parameters described in the paper.

Keywords: contrast enhancement; local histogram modification; mathematical morphology; level sets; shape; contrast-limited histogram equalization

¹<https://doi.org/10.5201/ipol.2018.236>

1 Introduction

In [6] was proposed the first local contrast enhancement method designed explicitly to prevent the introduction of ‘artifacts’ in the processed image. According to the mathematical morphology school [1, 35, 36], those artifacts are introduced whenever a processing operation modifies the **level sets** of the image. The family of level sets of an image $u : \Omega \rightarrow [0, 255]$ is defined as

$$\mathcal{X}_\lambda u = \{\mathbf{x} \in \Omega / u(\mathbf{x}) \leq \lambda\}, \quad (1)$$

for all values of λ in the range of u . The contours of the level sets are called level lines, and the set of all level lines of an image is called its **topographic map** [5].

Under fairly general conditions, an image can be reconstructed from its level-sets by the formula

$$u(\mathbf{x}) = \inf\{\lambda / \mathbf{x} \in \mathcal{X}_\lambda u\}. \quad (2)$$

If h is a strictly increasing function, the transformation $v = h(u)$ does not modify the family of level-sets of u , it only changes its index in the sense that

$$\mathcal{X}_{h(\lambda)} v = \mathcal{X}_\lambda u, \quad \forall \lambda. \quad (3)$$

Global histogram equalization, and in general any global tone mapping operation, involves an increasing mapping function, therefore preserving the image level sets. However, these operations fail to simultaneously increase the contrast in both dark and bright image regions; small bright regions, especially, are hardly visible after such a global operation. On the other hand, local histogram modification improves the contrast of small regions as well, but since the level-sets are not preserved, artificial objects are created. The method developed in [6] enjoyed the best of both words: the shape-preservation property of global techniques and the contrast improvement quality of local ones.

Figure 1 illustrates the shape preserving property of the method. In this figure an original image and the results of different processing techniques are presented. The level lines of each image are also displayed. The compared processing methods are: global histogram equalization (HE); local histogram equalization²; CLAHE [38], a popular local technique which shall be described in Section 4; and the proposed local method, which we will call Morpho-Local Histogram Equalization (MLHE) from now on. We observe that MLHE is the only local technique which increases the contrast without creating new level lines.

The authors of [6] gave a mathematical proof that the proposed method indeed preserved the topographic map of the original image. Implicit to the method is the assumption that all level sets of an image may be organized in a tree structure [2]. Since its publication in 1999, several image enhancement techniques have been proposed in the literature, we give a quick overview of them in Section 2. In the present paper we shall provide a detailed account of the MLHE algorithm (Section 3), we will present the control parameters used in the original implementation to limit the excessive enhancement of noise (Section 3.1), and we will propose two new alternatives to the classical histogram equalization technique, which improve the performance of the method (Section 4). Finally, we will compare the obtained results with the ones achieved using the popular CLAHE technique (Section 5). Some conclusions shall be presented in Section 6.

2 Overview of Image Enhancement Techniques

According to [31], “the principal objective of enhancement techniques is to process an image so that the result is more suitable than the original image for a specific application”. Depending on which

²The algorithm is described in [31]: each pixel is assigned a value computed after performing histogram equalization on a subimage around it.

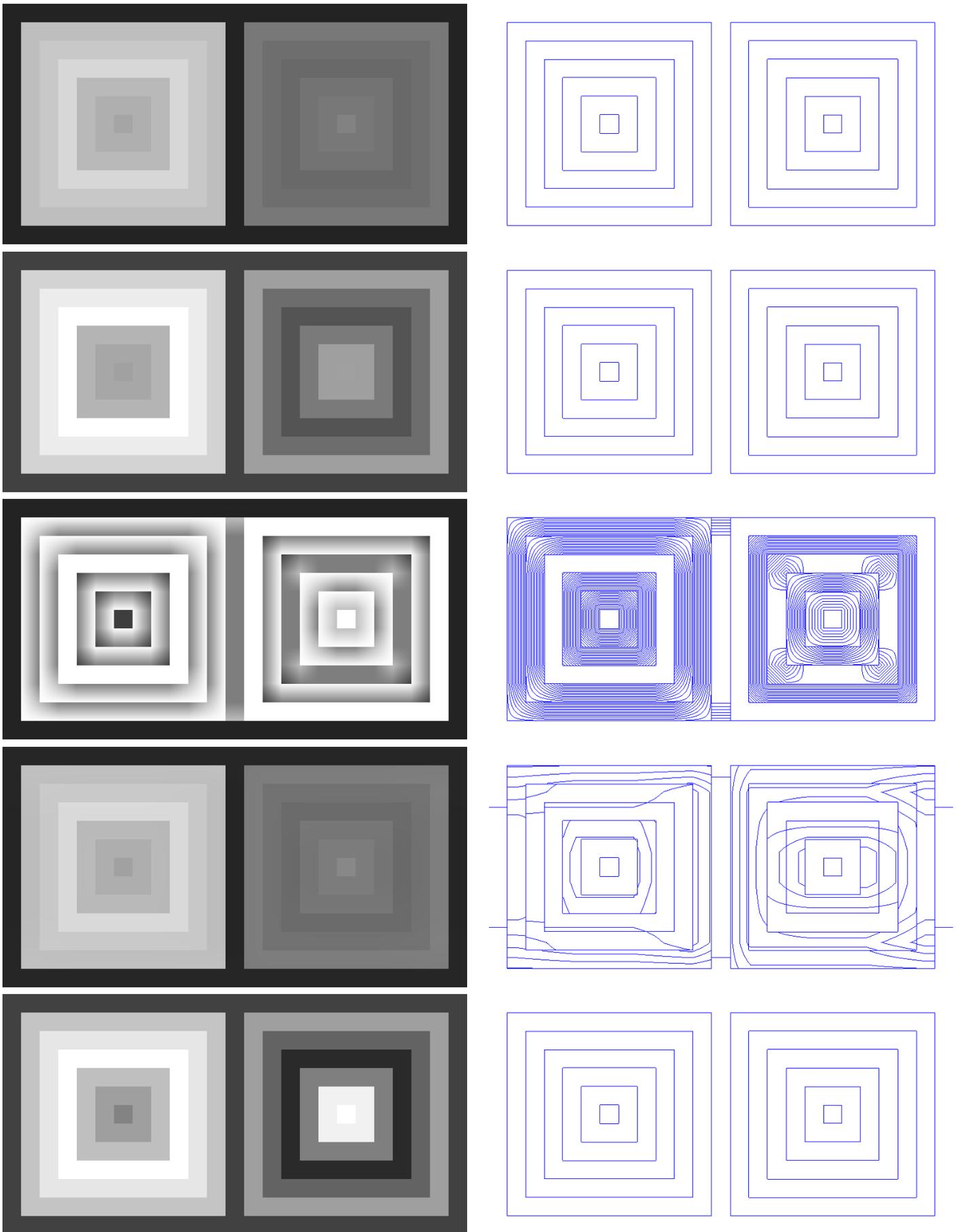


Figure 1: From top to bottom: original, HE, local HE, CLAHE, MLHE. The level lines of each image are displayed on the right. All the level lines are displayed for all the images, with the exception of local HE, for which only lines corresponding to levels multiple of 10 are displayed.

image characteristics need to be “enhanced” (brightness, contrast, small details, etc) several different enhancement methods have been proposed over the years. We give in this section a quick overview of the more relevant publications, which will serve to place our method in the landscape of image enhancement techniques.

The most popular methods for image enhancement can be classified into two categories:

1. Retinex-inspired methods. These methods have their origin in the Retinex theory of Land [16, 17], which tried to explain the adaptability of the human visual perception to local illumination conditions. Although the primary goal of Retinex is not to “enhance” the image but to mimic the behaviour of the human visual system (HVS), the Retinex algorithms have shown useful to improve the brightness and contrast of the images, and are therefore used for image enhancement.

Several different, and sometimes opposing, implementations of Retinex exist, which can be classified into three major classes. In all the cases the goal is to recover the reflectance of the scenes, discounting the effect of the illumination. The first class explores the ratios of reflectances using a variety of image paths or comparing the current pixel value to a set of random pixels [19, 26, 30]. This approach can be formalized using partial differential equations [12, 4, 27, 21]. The second class of methods compute the reflectance by subtracting to the input image a blurred version of itself [18, 14]. These are the so-called center/surround approaches, which include the popular Multiscale Retinex (MSR) algorithm [14, 13, 29]. Although not usually classified as a Retinex method, ACE [33, 11] may be also considered as a center/surround technique. A third category of methods introduce regularization on reflectance and illumination using a variational framework [15, 3, 24]. We refer to [37] for a recent review on Retinex methods.

2. Tone-mapping methods. These methods usually map the values of the image’ pixels from a given range of values to a new range much smaller than the original, as in the case of HDR (high dynamic range) input images that are mapped onto LDR (low dynamic range) output images. However, in some cases the mapping just implies a redistribution of the pixel values over the original input range, as in classical histogram equalization and its variants [23] and other basic processing methods (gamma correction, simplest color balance [20], etc). Moreover, many of the techniques developed for HDR images can be applied to map LDR images to new LDR images with improved brightness and contrast.

These techniques can be global, when the same tone mapping function is applied to the whole image, or local, where the mapping depends on the local neighborhood of each pixel. Among the most popular global techniques we may mention: Drago et al. method [7], which uses a logarithmic tone mapping adapted to the luminance characteristics of the scene; Reinhard et al. [32], which uses a variant of the Naka-Rushton equation that models photoreceptor responses to stimuli; in [25] Mai et al. use a tone curve that minimizes the distortions produced by the combined processes of tone-mapping and compression.

Examples of local techniques are CLAHE [38], LCC [28, 34] and the recently introduced LogLocal method [22]. CLAHE adapts the histogram equalization method to image subwindows, while LCC and LogLocal use, respectively, gamma corrections and logarithmic mappings adapted to local neighborhoods of each pixel. The technique described in this paper, MLHE, falls into the category of histogram-based local tone mapping method.

Some of the tone mapping methods are specially focused on recovering the image details. This is achieved by decomposing the image at different resolution scales, the coarser ones keeping the global illumination information and the finer ones the details. The former are attenuated and the later amplified. Examples of these techniques are Durand et al. [8] and Fattal et al. [10] methods.

In recent years, the focus of tone mapping techniques has shifted from static images to high dynamic range videos. A recent review is available at [9].

3 Description of the Algorithm

MLHE is a recursive algorithm that is applied on the connected components of the bi-level sets of an image. In order to describe the method we need to define first the notions of **bi-level set** and **connected component**.

The bi-level set³ $[\lambda, \mu]$ of an image $u : \Omega \rightarrow [0, 255]$ is composed of the pixels whose value is in the range $[\lambda, \mu]$.

$$\mathcal{X}_{[\lambda, \mu]}u = \{\mathbf{x} \in \Omega / \lambda \leq u(\mathbf{x}) \leq \mu\}. \quad (4)$$

Two pixels (x, y) and (x', y') in Ω are said to be ‘connected’ if $(|x - x'|, |y - y'|) \in \{(1, 0), (0, 1)\}$ (that is, one of them is the left, right, lower or upper neighbor of the other⁴). A level set (or a bi-level set) \mathcal{X} may be composed of one or several connected components, each of which is a subset of \mathcal{X} such that all of its pixels are connected. We refer to [6] for a formal definition of connected component. Algorithm 4 describes a method to compute the connected components of a set of pixels.

The main steps of the proposed algorithm are described in Algorithm 1. The method is applied on gray level images or, in the case of color images, on the intensity component, defined as the average of the three color channels.

The recursive algorithm (described in Algorithms 2 and 3) is applied on the connected components of bi-level sets of decreasing range. Given a connected component of a bi-level set $\mathcal{X}_{[\lambda, \mu]}$, the histogram of intensity values of the pixels in the component is equalized in the range $[\lambda, \mu]$. Then, the initial range is divided by 2 and the process is repeated for the connected components of the bi-level sets $[\lambda, \lfloor \frac{\lambda + \mu}{2} \rfloor]$ and $[\lfloor \frac{\lambda + \mu}{2} \rfloor + 1, \mu]$. The method is initially applied (level 0 of the recursion) on the whole image domain Ω , which is the only connected component of the set $\mathcal{X}_{[0, 255]}$. Next, the connected components of the bi-level sets $[0, 127]$ and $[128, 255]$ are processed (level 1 of the recursion). In the next step (level 2), the connected components of the bi-level sets $[0, 63]$, $[64, 127]$, $[128, 191]$ and $[192, 255]$, extracted from the connected components computed in the previous step, are processed. This process is repeated until the range of the bi-level sets is 2. The whole recursion has 8 levels (since the range of values in a 8-bit digital image is $256 = 2^8$), numbered from 0 to 7. Remark that the level 0 of the algorithm is a classical global histogram equalization.

Algorithm 5 describes the method for equalizing the histogram of intensity values of a connected component \mathcal{S} in a given range $[\lambda, \mu]$. The method applies the following tone mapping function

$$I'(\mathbf{x}) = \lambda + (\mu - \lambda)H(I(\mathbf{x})), \quad (5)$$

where I and I' are the original and processed intensity values and

$$H(\lambda) = \frac{|\{\mathbf{x} \in \mathcal{S} / I(\mathbf{x}) \leq \lambda\}|}{|\mathcal{S}|}, \quad (6)$$

is the cumulative distribution function of the intensity values. The symbol $|\cdot|$ denotes the number of pixels in the sets.

For color images, the RGB components of the final output are computed as

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

³In [6] the bi-level sets of u are called *sections of the topographic map* of u

⁴We use here the notion of ‘4-connectivity’.

where R , G and B are the original RGB values, I is the original intensity and I' the processed intensity. Remark that by using Equation (7) the original R/G/B ratios are kept, thus preserving in this sense the original chrominance of the input image. Moreover, the intensity of the processed color image is I' . For some pixels, it is possible that the processed color values exceed the limit permitted for 8-bit images (i.e. 255). In these cases, the multiplicative factor $\frac{I'}{I}$ is replaced by the maximum factor that permits to avoid out-of-range results. Algorithm 6 describes the color processing step.

Algorithm 1: MLHE main

Input : input image, $u(\mathbf{x}) = (R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}))$, $\mathbf{x} \in \Omega$ (image domain)
Output : output image, u'

// Get image Intensity component as average of the three color channels
 1 $I(\mathbf{x}) = \left\lfloor \frac{R(\mathbf{x})+G(\mathbf{x})+B(\mathbf{x})}{3} \right\rfloor$, $\forall \mathbf{x} \in \Omega$ // $\lfloor \cdot \rfloor$ denotes rounding to the closest integer
 //Initialize output Intensity
 2 $I' = I$
 //Apply MLHE Algorithm
 3 RecursiveHistogramEqualization(I' , Ω , 0, 255)
 //Apply changes in Intensity to R, G, B components
 4 $u' = \text{GetColorImage}(u, I, I')$

Algorithm 2: RecursiveHistogramEqualization (MLHE Algorithm)

Input : input image (one channel), $I(\mathbf{x}) \in \{0, 1, \dots, 255\}$, $\mathbf{x} \in \Omega$ (Ω : image domain)
Input : set of pixels on which histogram equalization is performed, $\mathcal{S} \subseteq \Omega$
Input : (min, max) , limits of the range of values of the pixels in \mathcal{S} , $I(\mathcal{S}) \in [min, max]$
Output : output image: update values in I

1 **Parameters:** maximum allowed level of recursion, L_{max} (optional)

//Equalize Histogram
 2 $I' = \text{HistogramEqualization}(I, \mathcal{S}, min, max)$
 //Update values in I
 3 $I(\mathbf{x}) = I'(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{S}$
 4 //Recursion level, $\frac{256}{2^{level}} = max - min + 1$
 5 $level = \log_2 \frac{256}{max-min+1}$
 6 **if** $level + 1 > L_{max}$ **then**
 7 | //Attained maximum level of recursion
 8 | **Return**
 9 **if** $max - min \leq 2$ **then**
 10 | //not enough values in range for dyadic split
 10 | **Return**
 //Mid-value of initial range, rounded to the nearest lower integer
 11 $med = \lfloor \frac{min+max}{2} \rfloor$
 // Process (Bi-)Level Set $[min, \lfloor \frac{min+max}{2} \rfloor]$
 12 ProcessLevelSet(I, \mathcal{S}, min, med)
 // Process (Bi-)Level Set $[\lfloor \frac{min+max}{2} \rfloor + 1, max]$
 13 ProcessLevelSet($I, \mathcal{S}, med+1, max$)

Algorithm 3: ProcessLevelSet

```

Input      : input image (one channel),  $I(\mathbf{x}) \in \{0, 1, \dots, 255\}$ ,  $\mathbf{x} \in \Omega$  ( $\Omega$  : image domain)
Input      : set of pixels to be processed,  $\mathcal{S} \subseteq \Omega$ 
Input      :  $(min, max)$ , range of the (bi-)level set
1 Parameters: minimum required number of pixels in connected component,  $A_{min}$  (optional)
   //Extract (bi-)level set
2  $\mathcal{A} = \{\mathbf{x} \in \mathcal{S} / I(\mathbf{x}) \in [min, max]\}$ 
   //Connected components (4-connectivity) of the level set
3  $\{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_K\} = \text{GetConnectedComponents}(\mathcal{A})$ 
   //Process connected components
4 for  $i \in \{1, \dots, K\}$  do
5   if  $|\mathcal{A}_i| \geq A_{min}$  then
6     RecursiveHistogramEqualization( $I, \mathcal{A}_i, min, max$ )

```

Algorithm 4: GetConnectedComponents

```

Input      : set of pixels,  $\mathcal{A} \subseteq \Omega$ 
Output     : set of connected components of  $\mathcal{A}$ ,  $\{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_K\}$  (use 4-connectivity)
1  $K = 0$                                      //K = number of connected components
2 while  $\mathcal{A} \neq \emptyset$  do
3    $K = K + 1$ 
4    $\mathcal{A}_K = \emptyset$                          //Initilialize connected component
   //First pixel of the connected component
5    $\mathbf{x} = (x, y) \leftarrow \mathcal{A}$                  //Get pixel from  $\mathcal{A}$ 
6    $\mathcal{A} = \mathcal{A} \setminus \{\mathbf{x}\}$                  //Remove pixel from  $\mathcal{A}$ , so it is not used again
   //Organize pixels using a FIFO queue data structure
7    $\mathcal{F} = \emptyset$ 
8    $\mathbf{x} \rightarrow \mathcal{F}$                          //Add at the beginning of the queue
9   while  $\mathcal{F} \neq \emptyset$  do
10     $\mathbf{x} \leftarrow \mathcal{F}$                        //Extract last element of the queue
11     $\mathcal{A}_K = \mathcal{A}_K \cup \{\mathbf{x}\}$                  //Add pixel to  $\mathcal{A}_K$ 
    //Get 4-connected pixels
12    for  $\mathbf{x}_c \in \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$  do
13      if  $\mathbf{x}_c \in \mathcal{A}$  then
14         $\mathbf{x}_c \rightarrow \mathcal{F}$                  //Add at the beginning of the queue
15         $\mathcal{A} = \mathcal{A} \setminus \{\mathbf{x}_c\}$          //Remove pixel from  $\mathcal{A}$ , so it is not used again

```

Algorithm 5: HistogramEqualization

Input : input image (one channel), $I(\mathbf{x}) \in \{0, 1, \dots, 255\}$, $\mathbf{x} \in \Omega$ (Ω : image domain)
Input : set of pixels on which histogram equalization is performed, $\mathcal{S} \subseteq \Omega$
Input : (min, max) , limits of the range of values of the pixels in \mathcal{S} , $I(\mathcal{S}) \in [min, max]$
Output : output image, I'

- 1 **Parameters:** maximum allowed ratio between initial and final range of values, r_{max} (optional)
- 2 **Parameters:** minimum allowed ratio between initial and final range of values, r_{min} (optional)

```

//Discrete Cumulative Distribution Function, for pixels in S
3  $H(\lambda) = \frac{|\{\mathbf{x} \in \mathcal{S} / I(\mathbf{x}) \leq \lambda\}|}{|\mathcal{S}|}$ ,  $\lambda \in [min, max]$  //| $\cdot$ | denotes number of pixels in set
//Get output values uniformly distributed in range [min, max]
4  $I'(\mathbf{x}) = [min + (max - min) \cdot H(I(\mathbf{x}))]$  //[] denotes rounding to the closest integer
5  $range_I = \max_{\mathbf{x} \in \mathcal{S}}(I(\mathbf{x})) - \min_{\mathbf{x} \in \mathcal{S}}(I(\mathbf{x}))$ 
6  $range_{I'} = \max_{\mathbf{x} \in \mathcal{S}}(I'(\mathbf{x})) - \min_{\mathbf{x} \in \mathcal{S}}(I'(\mathbf{x}))$ 
7 if  $range_I = 0$  or  $\frac{range_{I'}}{range_I} > r_{max}$  or  $\frac{range_{I'}}{range_I} < r_{min}$  then
    | //Excessive increase (or decrease) in range: use original values
8 |  $I' = I$ 
    
```

Algorithm 6: GetColorImage

Input : input image, $u(\mathbf{x}) = (R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}))$, $\mathbf{x} \in \Omega$ (image domain)
Input : intensity of input image, I
Input : processed intensity, I'
Output : output image, u'

- 1 **for** $\mathbf{x} \in \Omega$ **do**
- 2 | **if** $I(\mathbf{x}) \neq 0$ **then**
 - 3 | | $\alpha = \min\{\frac{I'(\mathbf{x})}{I(\mathbf{x})}, \frac{255}{\max\{R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x})\}}\}$
 - 4 | | $R' = \alpha \cdot R(\mathbf{x})$
 - 5 | | $G' = \alpha \cdot G(\mathbf{x})$
 - 6 | | $B' = \alpha \cdot B(\mathbf{x})$
 - 7 | | $u'(\mathbf{x}) = ([R'], [G'], [B'])$ //[] denotes rounding to the closest integer
- 8 | **else**
- 9 | | $u'(\mathbf{x}) = (0, 0, 0)$

3.1 Adding Parameters to the Method

Classical histogram equalization is a parameter-less algorithm, and its recursive application as described above does not require any parameters either. However, it is interesting to add a few control parameters that help to analyze the obtained results, and that permit to limit the over-enhancement of noise typical of histogram equalization methods.

Four control parameters may be added to MLHE. The lines of the algorithms concerning them have been marked in blue (Algorithms 2, 3, 5), to highlight the fact that they are optional parameters.

- L_{max} , the maximum level of recursion. By default, the recursive algorithm will run until the range of the bi-level sets is smaller than 2. By using this parameter we can analyze the evolution

of the result as a function of the recursion level. Figure 2 displays an example. The result for $L_{\max} = 0$ corresponds to a classical histogram equalization. We observe that, as the recursion level increases, smaller details are visible in the image. Results for $L_{\max} = 6$ and $L_{\max} = 7$ are not displayed since they are indistinguishable from those for $L_{\max} = 5$. In fact, the result practically stabilizes at $L = 3$. The bigger differences are observable from $L_{\max} = 0$ to $L_{\max} = 1$ and a few more changes are noticeable from $L_{\max} = 1$ to $L_{\max} = 3$. We recommend the reader to visualize the images in the provided supplementary HTML page to fully appreciate the differences between them.

- A_{\min} , minimum number of pixels in a connected component required to process it. Its value is typically set to 20 pixels. The use of this parameter has a double effect: first, it permits to accelerate the computation since very small connected components are not processed; second, noisy spots, which are usually small, are not over-enhanced by the method, leading to a better visual result. Figure 3 compares the results obtained with (image (b)) and without (image (a)) the use of this parameter for the image displayed in Figure 2. We observe that a few small bright spots have been removed from the result in (b) (better viewed in the supplementary HTML page).
- r_{\min} and r_{\max} , minimum and maximum allowed ratio between the range of values of the pixels of a connected component after and before histogram equalization. The goal is to prevent both excessive contrast reductions (and eventually the loss of level lines due to rounding errors), and excessive enhancements that increase the visibility of noise in the final result. Figure 3 shows results obtained with different values of the parameters. We again suggest the reader to use the supplementary HTML page to fully appreciate the differences between the results. Results in images (c), (d) and (e) shall be compared to those in image (b). In all the cases the parameters L_{\max} and A_{\min} are set to 7 and 20, respectively. The differences between the images stem from the use of different settings for r_{\min} and r_{\max} . In image (c) r_{\min} was set to 0.8 and no limit was used for r_{\max} . We observe that some small regions are less enhanced when using this setting. Although the goal of this setting is indeed to prevent reductions of the contrast, somewhat surprisingly, when used in the recursive algorithm, it may lead to less contrasted small spots in the image. The reason is that without the use of the parameter two originally different levels may become quite close to each other and become part of the same bi-level set, which may be significantly enhanced (and become quite visible) at a further level of the recursion. Again, a reduction in the number of small enhanced regions is observed when $r_{\max} = 3$ and no limit is used for r_{\min} (image (d)). The use of both parameters ($r_{\max} = 3$ and $r_{\min} = 0.8$) permits a further reduction of these small contrasted regions, leading to a better visual result.

In practice, we set the parameters to $L_{\max} = 7$, $A_{\min} = 20$, $r_{\min} = 0.8$ and $r_{\max} = 3$. The results displayed in Figures 4 and 5 were obtained using these parameters. The result in Figure 1 was obtained without limiting the value of r_{\max} to permit the maximum enhancement of the synthetic image. For natural images this setting is excessive. We observe in Figures 4 and 5 that the contrast is increased with respect to the result of the global algorithm. However, in some parts of the images an excessive enhancement increases the visibility of noise (e.g. in the sky in Figure 4) and produces the appearance of white spots (in Figure 5).

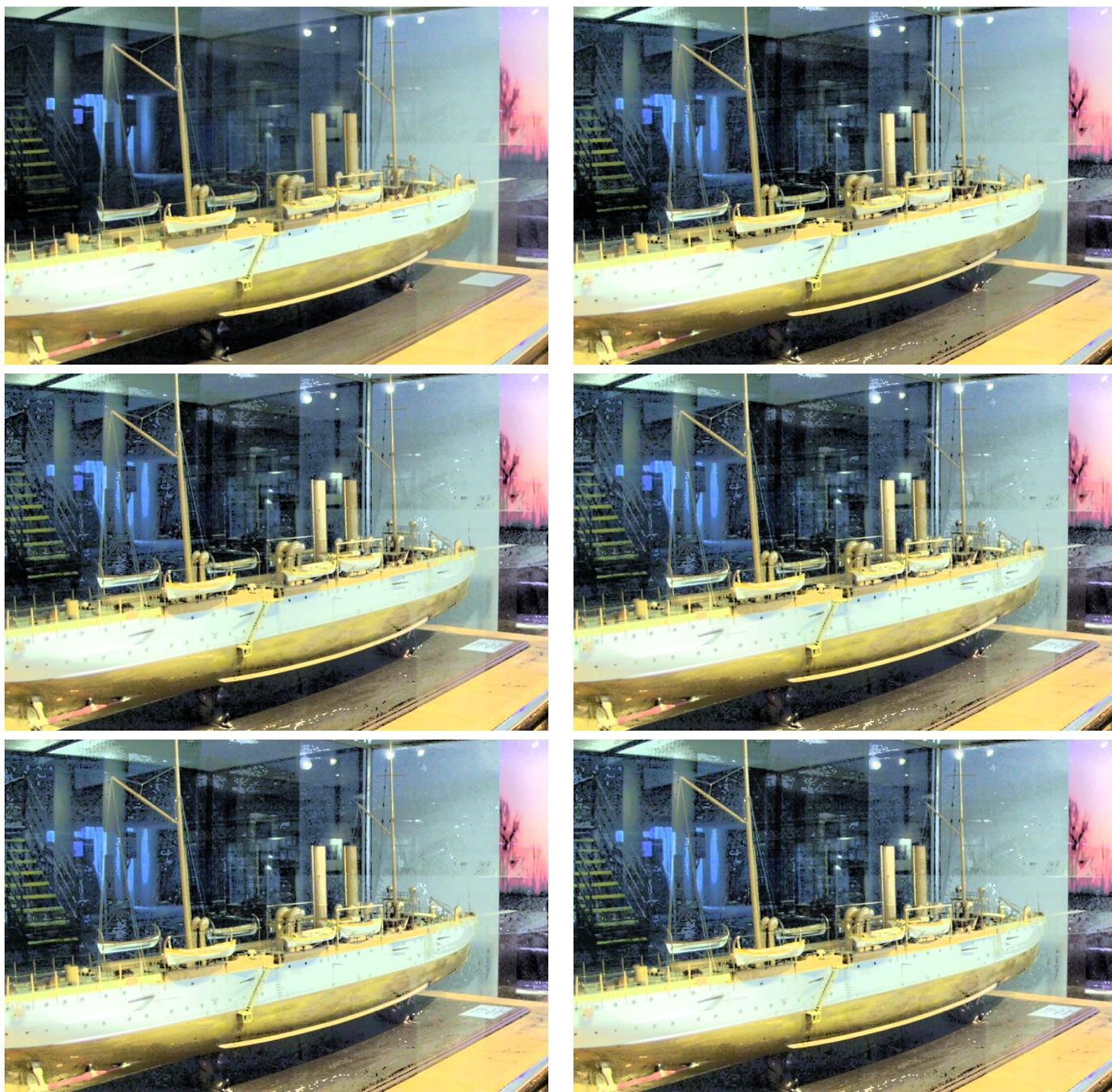
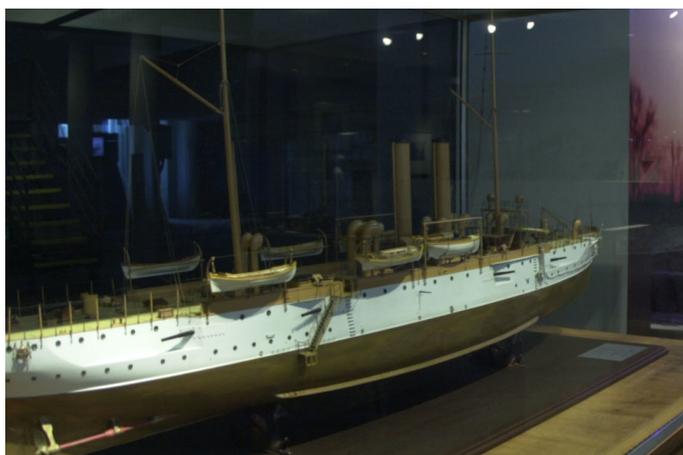
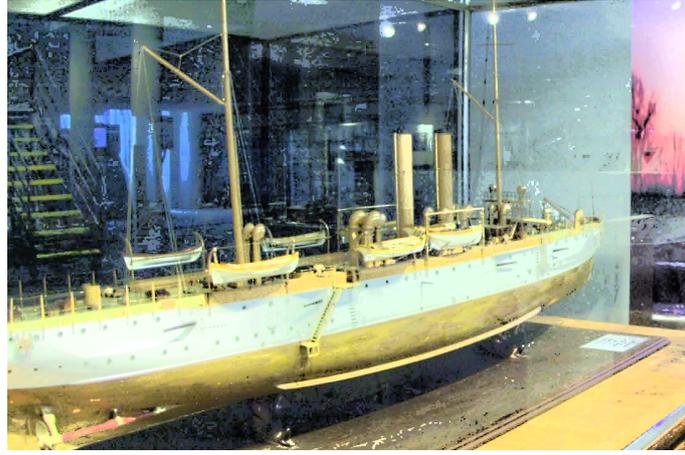
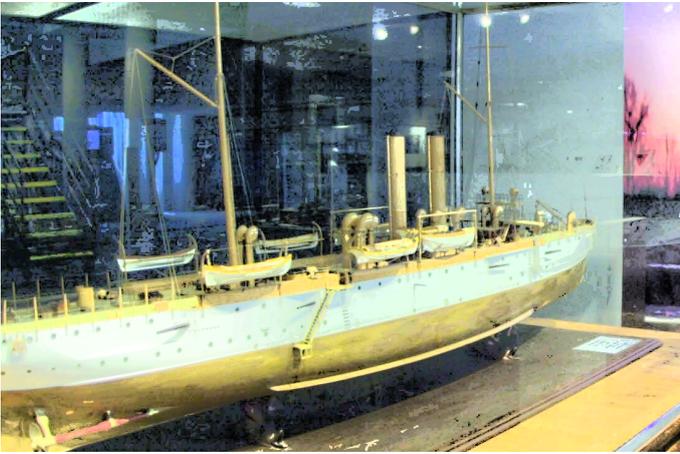


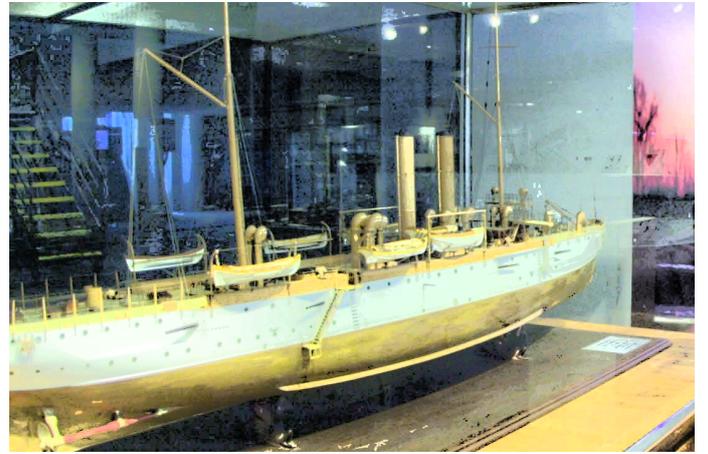
Figure 2: Top: original image. Bottom, from left to right and from top to bottom: MLHE results for $L_{\max} = 0, 1, 2, 3, 4, 5$. Results for $L_{\max} = 6$ and $L_{\max} = 7$ are indistinguishable from those for $L_{\max} = 5$.



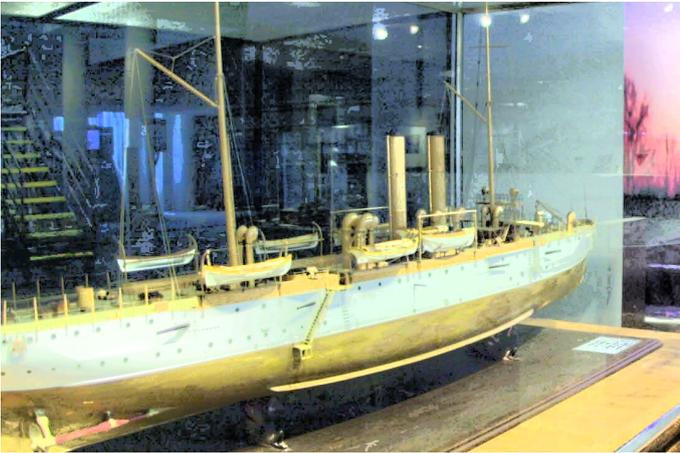
(a)



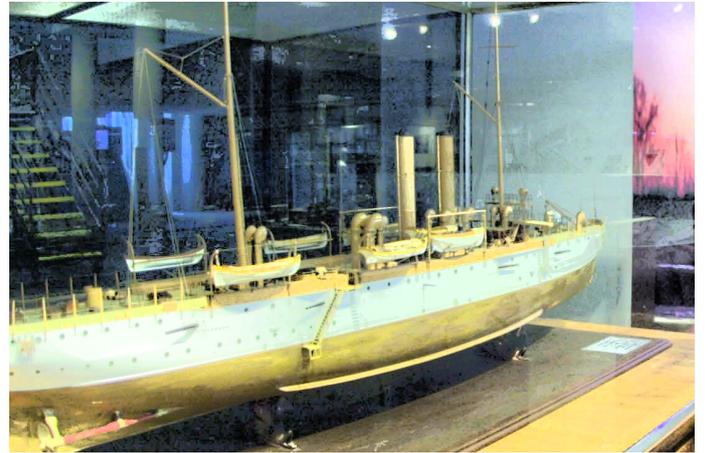
(b)



(c)



(d)



(e)

Figure 3: Results for different parameter settings, without recursivity limit ($L_{\max} = 7$). Top (a): result without control parameters (i.e. $A_{\min} = 0$, $r_{\max} = \infty$, $r_{\min} = 0$). Bottom: (b) limitation of connected components size ($A_{\min} = 20$, $r_{\max} = \infty$, $r_{\min} = 0$), (c) limitation of connected components size and minimum variation of range ($A_{\min} = 20$, $r_{\max} = \infty$, $r_{\min} = 0.8$), (d) limitation of connected components size and maximum variation of range ($A_{\min} = 20$, $r_{\max} = 3$, $r_{\min} = 0$), (e) limitation of connected components size and maximum and minimum variation of range ($A_{\min} = 20$, $r_{\max} = 3$, $r_{\min} = 0.8$).



Figure 4: Top: original image. Bottom: Bottom: left, result of classical HE; right, result of MLHE HE (default parameters: $L_{\max} = 7$, $A_{\min} = 20$, $r_{\max} = 3$, $r_{\min} = 0.8$).



Figure 5: Top: original image. Bottom: Bottom: left, result of classical HE; right, result of MLHE HE (default parameters: $L_{\max} = 7$, $A_{\min} = 20$, $r_{\max} = 3$, $r_{\min} = 0.8$).

Computational Complexity. The complexity of the method is proportional to the number of pixels in the image (N) and the number of recursion levels (L_{\max}). At each level of the recursion a total of $4N$ pixels are examined to compute the connected components of the level sets. This operation is performed L_{\max} times (by default $L_{\max} = 7$), which implies that the complexity of the algorithm is $O(L_{\max}N)$. By using a smaller value of L_{\max} (e.g. $L_{\max} = 3$) and limiting the processing to large enough connected components (e.g. setting $A_{\min} = 20$) it is possible to speed up the algorithm. As a reference of the computation time, a 2000×1300 image was processed in $1.2s$ using $L_{\max} = 3$ and $A_{\min} = 20$, and in $3.2s$ using $L_{\max} = 7$ and $A_{\min} = 0$. Classical global histogram equalization of the same image took $0.88s$. Computations were performed on a desktop computer with four Intel Core i5-4460 CPU at $3.20GHz$ and $16MB$ of RAM.

4 Improving the Original Method

The core of the algorithm described in the previous section is histogram equalization (HE). However, it is well known that HE excessively enhances noise and produces saturation in some parts of the images (see for example the white part of the boat in Figure 7-(a), the contrast is low due to the excessive brightness).

Several modifications of HE have been proposed in the literature to reduce these problems. In this section we describe two of them, and discuss how the results of MLHE improve when this modifications are incorporated into the recursive method.

Contrast-Limited Histogram Equalization. CLAHE (Contrast-Limited Adaptive Histogram Equalization) [38] is a popular technique for local contrast enhancement that differs from classical local equalization (as described in [31] and in the footnote in Section 1) in two aspects:

- 1) Histogram equalization is performed on non-overlapping subimages; to prevent the visibility of the subimages' boundaries bilinear interpolation is used: the final value at each pixel is a weighted average of the values assigned to the pixel using the histogram equalization mappings of the neighboring subimages.
- 2) In order to limit the enhancement of noise, the histogram equalization mapping function (i.e. the cumulative histogram of the input values) is modified as follows: a maximum number of pixels in each bin of the histogram of input values is permitted, if a bin contains more than this number of pixels its value is clipped; after clipping, the clipped mass of this bin is redistributed uniformly on all histogram bins to keep the histogram count identical. This procedure limits the slope of the cumulative histogram function and prevents excessive contrast enhancement.

We are interested in the second feature of CLAHE (the procedure for contrast reduction). Algorithm 7 describes the method, which depends on a single parameter c , the percentage of the number of pixels in the image that shall be clipped by the algorithm. Figure 6-center shows the result of applying this algorithm to the cumulative histogram in Figure 6-left. The associated images (results of global histogram equalization using these cumulative histograms) are displayed in Figures 7 (a) and (c). In this case the value of the parameter c was set to 0.01 . Other values of the parameter are tested in Figures 7 (b) and (d). In general, more contrasted but darker results are obtained with small values of c . In our experiments, we have fixed the value of the parameter to $c = 0.01$.

Controlled Piecewise Affine Histogram Equalization. The control over the slope of the cumulative distribution function in CLAHE is indirect, through the use of a clipping threshold in the histogram function. Lisani et al. proposed, in [23], a method which permitted an explicit control of

Algorithm 7: ControledHistogramEqualization (CLAHE-based version)

Input : input image (one channel), $I(\mathbf{x}) \in \{0, 1, \dots, 255\}$, $\mathbf{x} \in \Omega$ (Ω : image domain)
Input : set of pixels on which histogram equalization is performed, $\mathcal{S} \subseteq \Omega$
Input : (min, max) , limits of the range of values of the pixels in \mathcal{S} , $I(\mathcal{S}) \in [min, max]$
Output : output image, I'
Parameters: c , percentage of the number of pixels in \mathcal{S} that shall be clipped by the algorithm

```

//Histogram (normalized) of intensity values for pixels in  $\mathcal{S}$ 
1  $h(\lambda) = \frac{|\{\mathbf{x} \in \mathcal{S} / I(\mathbf{x}) = \lambda\}|}{|\mathcal{S}|}$  //  $|\cdot|$  denotes number of pixels in the set
//Clip values above threshold
2  $p_{\text{excess}} = 0$  //Percentage of samples discarded due to clipping
3 for  $\lambda \in [min, max]$  do
4   if  $h(\lambda) > c$  then
5      $p_{\text{excess}} = p_{\text{excess}} + (h(\lambda) - c)$ 
6      $h(\lambda) = c$ 
//Distribute discarded samples among all the bins in the range  $[min, max]$ 
7  $p_{\text{each}} = \frac{p_{\text{excess}}}{max - min + 1}$ 
8 for  $\lambda \in [min, max]$  do
9    $h(\lambda) = h(\lambda) + p_{\text{each}}$ 
//Discrete Cumulative Distribution Function
10  $H(\lambda) = \sum_{\mu \leq \lambda} h(\mu)$ 
//Compute output values
11  $I'(\mathbf{x}) = [min + (max - min) \cdot H(I(\mathbf{x}))]$  //  $[\cdot]$  denotes rounding to the closest integer
    
```

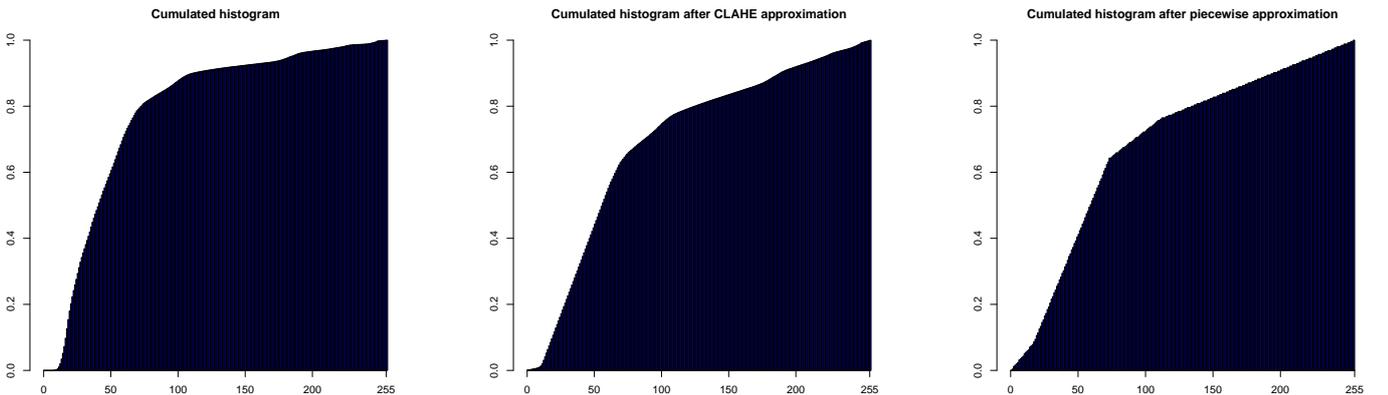


Figure 6: From left to right: original cumulative histogram, cumulative histogram after CLAHE modification, cumulative histogram after piecewise affine modification.

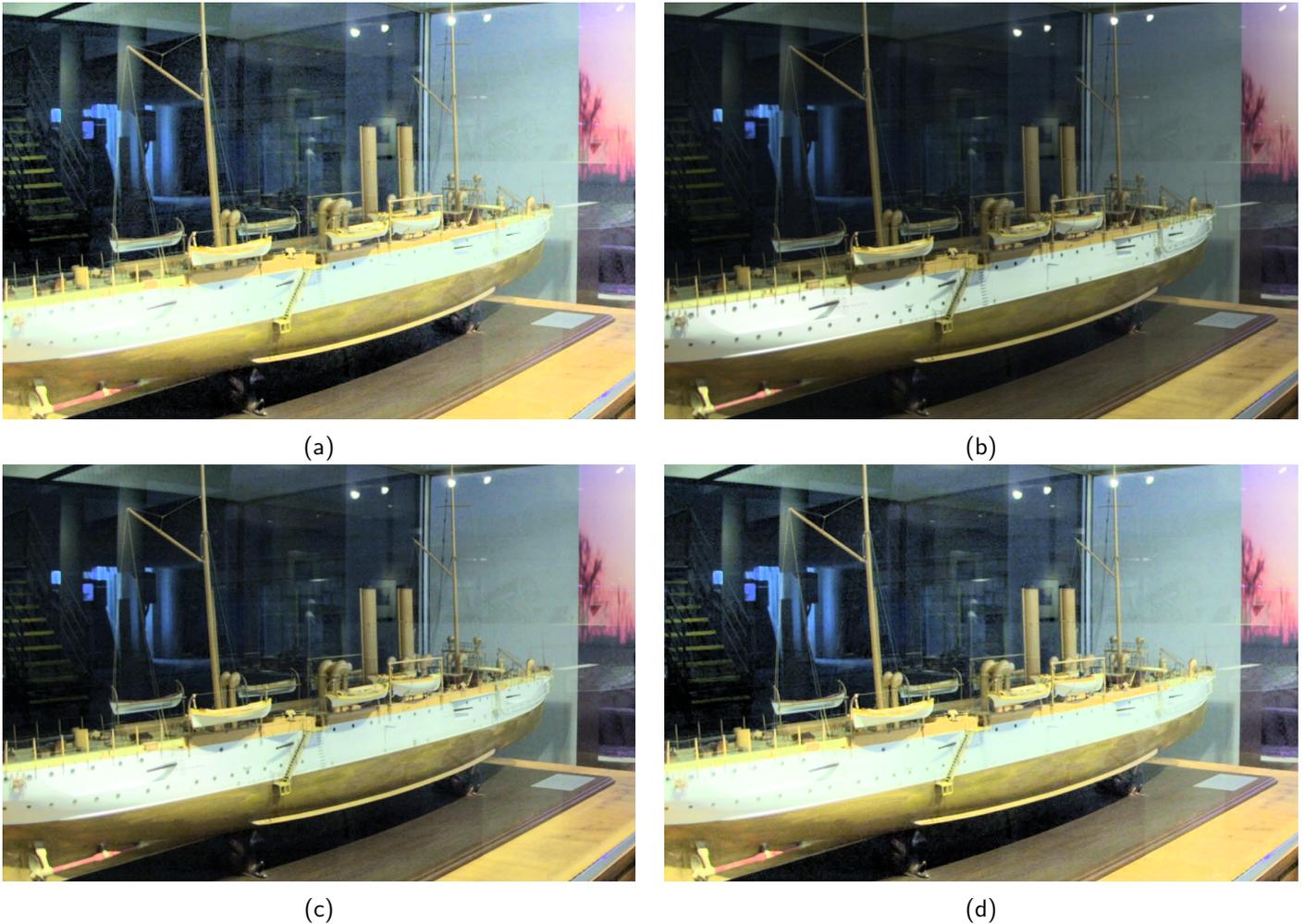


Figure 7: From left to right and from top to bottom: result of histogram equalization, results with contrast limited enhancement ($c = 0.005$, $c = 0.01$ and $c = 0.02$).

the slope: the cumulative distribution function was approximated by a series of linear segments with controlled slope. The technique is called Piecewise Affine Equalization (PAE) and depends on three parameters: N , the number of linear segments; and s_{\min} , s_{\max} their minimum and maximum allowed slope. The method is summarized in Algorithm 8, adapted to the case where the output range is arbitrary ($[min, max]$) and not just $[0, 255]$ as in [23]. It must be noted that for some combinations of the parameters the output range of PAE may be larger than the allowed output range. This may happen when both s_{\min} and s_{\max} are high. In that case, the original implementation of PAE just clips the values to the maximum (i.e. 255), producing the saturation of some levels (and the loss of the associated level lines). In our implementation, we rescale the values so that all of them fit in the $[min, max]$ range. This produces a reduction in the actual slopes of the linear segments. It is also possible, when both s_{\min} and s_{\max} are low, that the full output range is not used, producing a reduction in contrast. In this case our implementation rejects the approximation and the original values are kept.

Figure 6-right displays the result of applying this algorithm to the cumulative histogram in Figure 6-left. The result is similar to that of CLAHE (Figure 6-center). The associated images (results of global histogram equalization using these cumulative histograms) are displayed in Figures 8 (a) and (e). In this case the values of the parameters were $N = 10$, $s_{\min} = 0.5$ and $s_{\max} = 3$. Other values of the parameters are tested in Figures 8 and 9.

Algorithm 8: ControlledHistogramEqualization (PAE version)

```

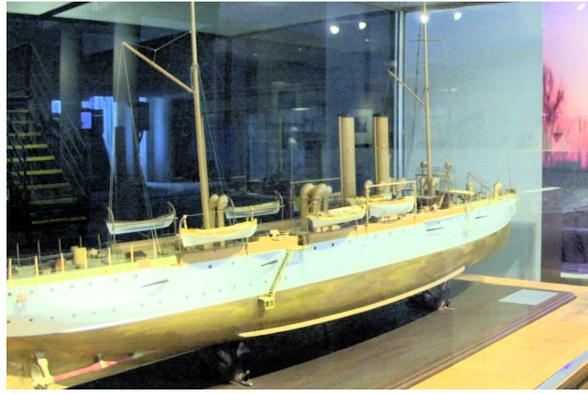
Input      : input image (one channel),  $I(\mathbf{x}) \in \{0, 1, \dots, 255\}$ ,  $\mathbf{x} \in \Omega$  ( $\Omega$  : image domain)
Input      : set of pixels on which histogram equalization is performed,  $\mathcal{S} \subseteq \Omega$ 
Input      :  $(min, max)$ , limits of the range of values of the pixels in  $\mathcal{S}$ ,  $I(\mathcal{S}) \in [min, max]$ 
Output     : output image,  $I'$ 
Parameters:  $N$  number of linear segments,  $s_{min}$  minimum slope of the linear segment,  $s_{max}$ 
               maximum slope of the linear segment

//Discrete Cumulative Distribution Function, for pixels in  $\mathcal{S}$ 
1  $H(\lambda) = \frac{|\{x \in \mathcal{S} / I(x) \leq \lambda\}|}{|\mathcal{S}|}$  //  $|\cdot|$  denotes number of pixels in set
//Partition points
2  $y_k = min + \frac{max-min}{N}k$ ,  $k = 0, 1, \dots, N$ 
3  $x_k = H^{-1}\left(\frac{k}{N}\right)$ ,  $k = 0, 1, \dots, N$  //Inverse distribution:  $H^{-1}(p) = \min\{\lambda / H(\lambda) \geq p\}$ 
4 for  $k = 0, 1, \dots, N - 1$  do
5    $m_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$ 
6   if  $m_k < s_{min}$  then
7      $m_k = s_{min}$ 
8   if  $m_k > s_{max}$  then
9      $m_k = s_{max}$ 
10   $y_{k+1} = T_k(x_{k+1})$ , with  $T_k(x) = y_k + m_k(x - x_k)$ 
11  Transform the values  $x \in [x_k, x_{k+1}]$  into  $y \in [y_k, y_{k+1}]$  by  $y = T_k(x)$ 
//Check final range of values
12 if  $y_N > max$  then
13   //this implies saturation, rescale output values so that  $y_N = max$ 
14    $y_{max} = y_N$ 
15   for  $k = 1, \dots, N$  do
16      $y_k = min + \frac{max-min}{y_{max}-min}(y_k - min)$ 
17 if  $y_N < max$  then
18   //The dynamic range is not fully used (it may happen when  $\mathcal{S}$  contains too
19   few values)
20 Return

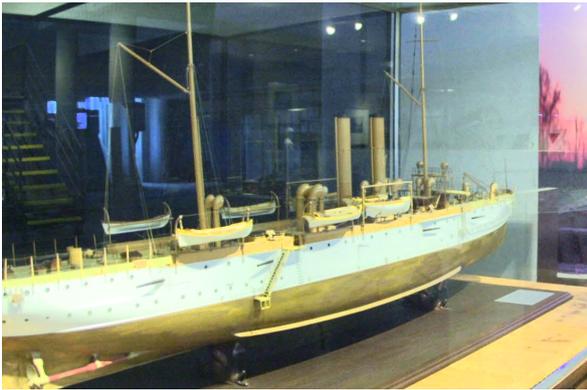
```

In general, for the same values of s_{min} and s_{max} , higher values of N produce darker results (compare images (b), (d), (f) with (c), (e), (g) in Figure 8). For the same value of N and s_{min} , an increase in s_{max} produces brighter results (compare images (b) and (c), (d) and (e) and (f) and (g) in Figure 9), which sometimes leads to over-enhancement of the noise. Finally, for the same value of N and s_{max} , an increase in s_{min} increases the contrast in bright regions of the image, at the expense of producing a darker result (compare images (b), (d) and (f), and (c), (e) and (g) in Figure 9). The figures are better viewed in the HTML page provided as supplementary material.

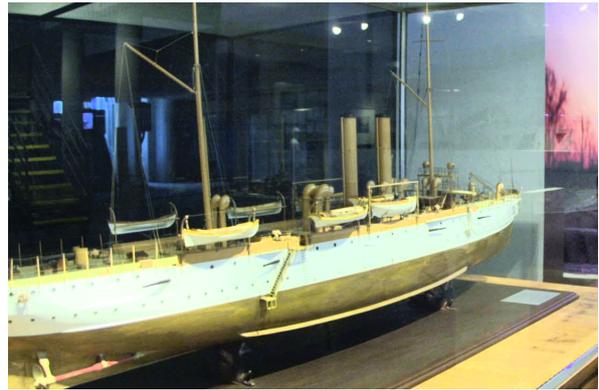
In our experiments we have fixed the values of the parameters to $N = 5$, $s_{min} = 1$ and $s_{max} = 3$.



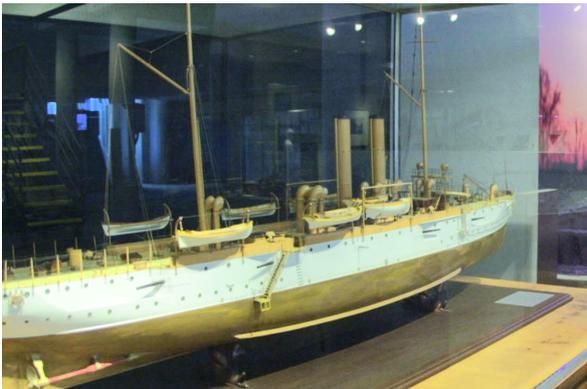
(a)



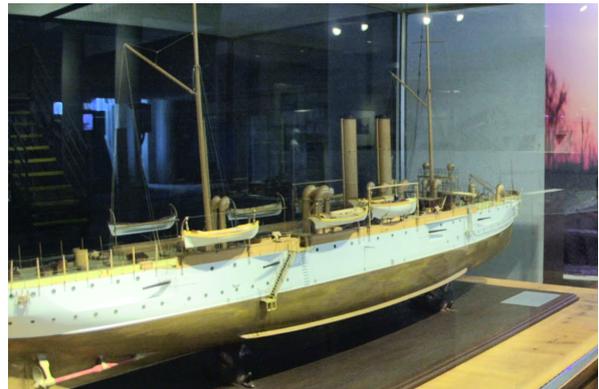
(b)



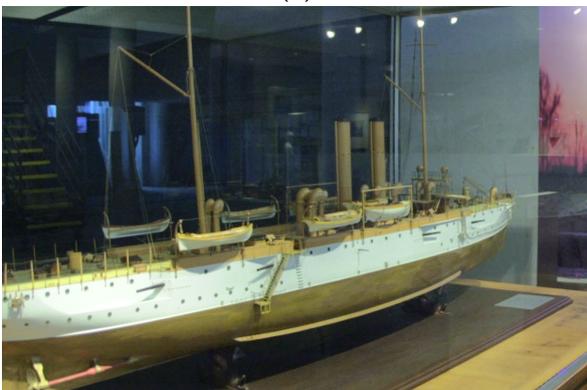
(c)



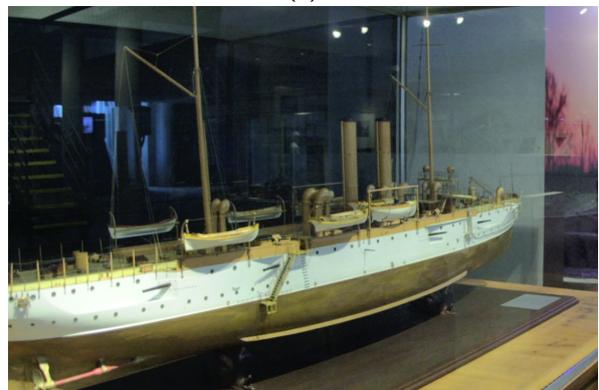
(d)



(e)



(f)



(g)

Figure 8: Top: (a) result of histogram equalization. Bottom, different results of piecewise affine equalization. First column, from top to bottom: (b) $\{N = 5, s_{\min} = 0, s_{\max} = 3\}$, (d) $\{N = 5, s_{\min} = 0.5, s_{\max} = 3\}$, (f) $\{N = 5, s_{\min} = 1, s_{\max} = 3\}$, Second column, from top to bottom: (c) $\{N = 10, s_{\min} = 0, s_{\max} = 3\}$, (e) $\{N = 10, s_{\min} = 0.5, s_{\max} = 3\}$, (g) $\{N = 10, s_{\min} = 1, s_{\max} = 3\}$,

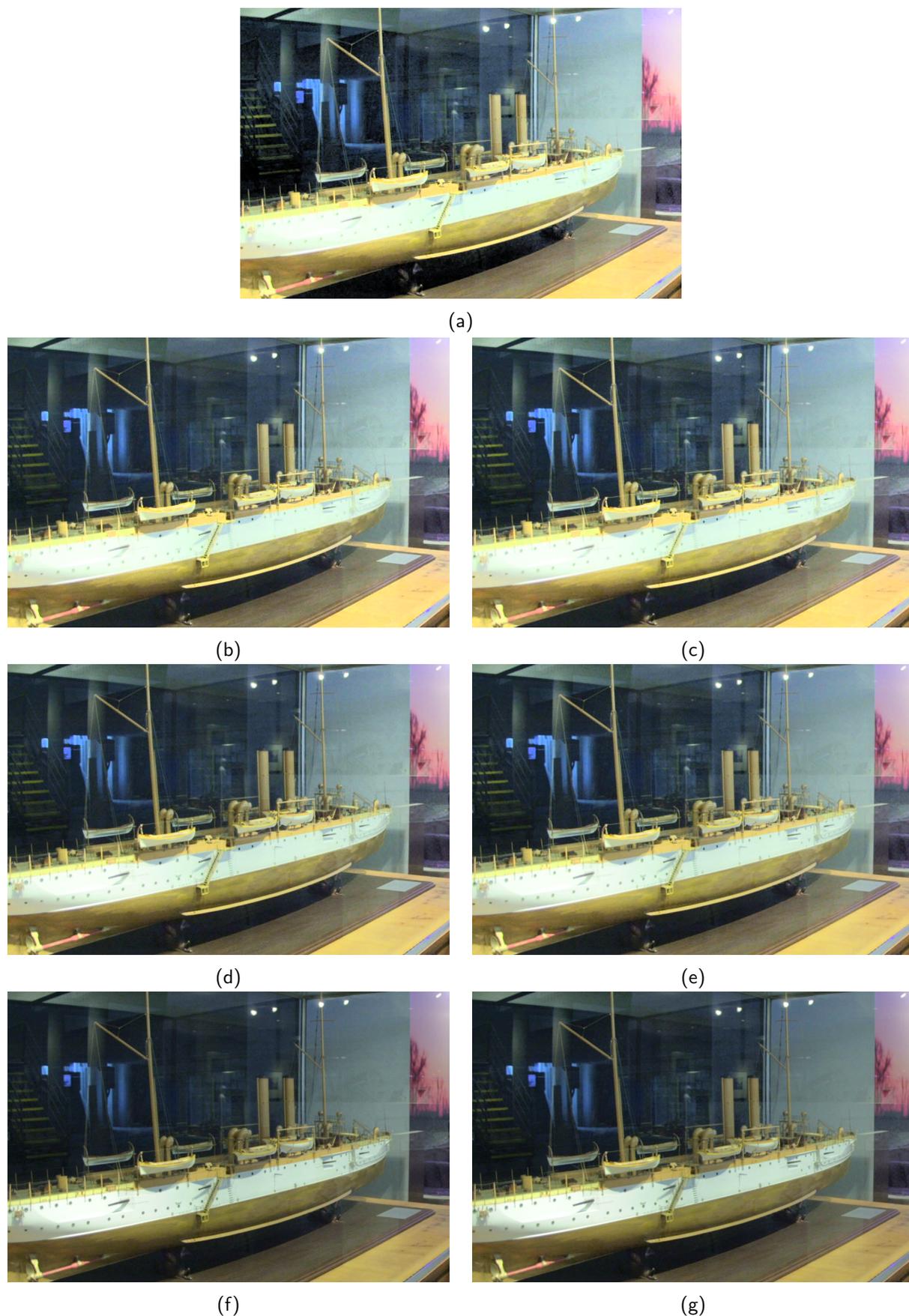


Figure 9: Top: (a) result of histogram equalization. Bottom, different results of piecewise affine equalization. First column, from top to bottom: (b) $\{N = 5, s_{\min} = 0, s_{\max} = 3\}$, (d) $\{N = 5, s_{\min} = 0.5, s_{\max} = 3\}$, (f) $\{N = 5, s_{\min} = 1, s_{\max} = 3\}$. Second column, from top to bottom: (c) $\{N = 5, s_{\min} = 0, s_{\max} = 5\}$, (e) $\{N = 5, s_{\min} = 0.5, s_{\max} = 5\}$, (g) $\{N = 5, s_{\min} = 1, s_{\max} = 5\}$.

4.1 MLHE with controlled histogram equalization

We have tested the use of Algorithms 7 and 8 in our recursive method. We just simply replace the call to Algorithm 5 in line 2 of Algorithm 2 by the call to one of these algorithms. In this case, the parameters r_{\min} and r_{\max} are not necessary, while L_{\max} and A_{\min} have been fixed to 7 and 20 in all our experiments. Moreover, the parameters of Algorithms 7 and 8 have been fixed to the values discussed in the previous section.

Figures 10 to 12 compare the results obtained with the different histogram equalization methods, which we denote as MLHE-HE (original implementation, using Algorithm 5), MLHE-CLAHE (implementation using Algorithm 7) and MLHE-PAE (implementation using Algorithm 8).

In Figures 10 and 11 the same original images shown in Figures 4 and 5 are used, so we can assess the improvement obtained with the new algorithms. We observe that both methods are able to reduce the enhancement of noise and the presence of bright spots in the processed images. Moreover, the obtained colors are less saturated and look more natural. In general, the results with MLHE-PAE are slightly darker than with MLHE-CLAHE, but the visibility is better in dark regions (e.g. the bottom of the image in Figure 11), and sometimes also in the bright zones (e.g. the lighthouse in Figure 12).



Figure 10: From left to right and from top to bottom: original image and results of MLHE-HE, MLHE-PAE and MLHE-CLAHE.



Figure 11: From left to right and from top to bottom: original image and results of MLHE-HE, MLHE-PAE and MLHE-CLAHE.



Figure 12: From left to right and from top to bottom: original image and results of MLHE-HE, MLHE-PAE and MLHE-CLAHE.

5 Comparison to CLAHE

In the final section of this article we compare the proposed improved MLHE methods with one of the most popular techniques for local contrast enhancement, namely the CLAHE method described in Section 4. We use the Matlab implementation of the method with its default parameters⁵. In order to perform a fair comparison, CLAHE is applied on the intensity component of the images and the final color result is obtained by applying Algorithm 6, as in the MLHE methods.

Figures 13 to 15 display the results obtained with CLAHE, MLHE-PAE and MLHE-CLAHE. In general, CLAHE obtains more contrasted images, although in some cases (e.g. the wooden platform inside the display cabinet in Figure 13) the enhancement seems excessive. In the case of the medical image in Figure 14, both versions of MLHE are able to retrieve more details from the original image than CLAHE, while the result of MLHE-CLAHE is more contrasted than the one of MLHE-PAE. In Figure 15 CLAHE is able to enhance the contrast of the girl's dress while both MLHE-CLAHE and MLHE-PAE saturate the color. However, the overall colors of the image look more natural in the MLHE results.



Figure 13: From left to right and from top to bottom: original image and results of CLAHE, MLHE PAE and MLHE CLAHE.

⁵Matlab's *adaphisteq* function. <https://es.mathworks.com/help/images/ref/adaphisteq.html>

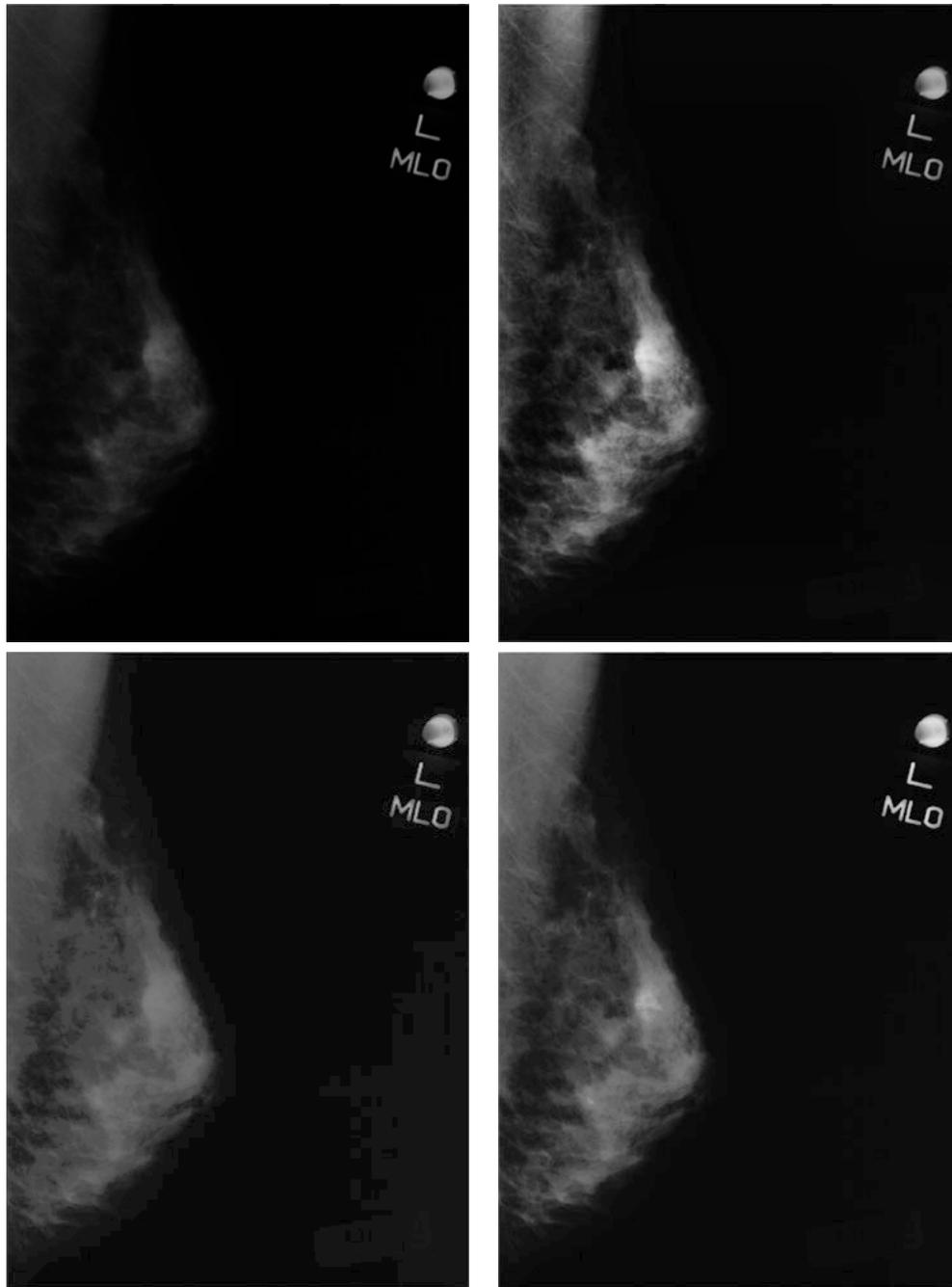


Figure 14: From left to right and from top to bottom: original image and results of CLAHE, MLHE PAE and MLHE CLAHE.



Figure 15: From left to right and from top to bottom: original image and results of CLAHE, MLHE PAE and MLHE CLAHE.

6 Conclusions

We have described in this paper the algorithm for local contrast enhancement proposed in [6]. The algorithm is based on the recursive application of histogram equalization on the connected components of the bi-level sets of the image and, by construction, it preserves the level sets structure of the image, thus avoiding in this sense the creation of ‘artifacts’. The drawbacks of the original implementation, related to an excessive enhancement of noise produced by the histogram equalization algorithm, are discussed and two possible modifications are proposed. The obtained results improve the ones obtained with the original algorithm and are comparable, in general, with the results of the popular CLAHE method.

Acknowledgements

The author was supported by grants TIN2014-53772, TIN2017-85572-P, TIN2014-58662-R and DPI2017-86372-C3-3-R.

Image Credits

All images from <https://dragon.larc.nasa.gov/retinex/pao/news/> except



CC by Juan Gabriel Gomila

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. <https://doi.org/10.1007/BF00375127>.
- [2] C. BALLESTER, V. CASELLES, AND P. MONASSE, *The tree of shapes of an image*, ESAIM: Control, Optimisation and Calculus of Variations, 9 (2003), pp. 1–18. <https://doi.org/10.1051/cocv:2002069>.
- [3] M. BERTALMIO, V. CASELLES, E. PROVENZI, AND A. RIZZI, *Perceptual color correction through variational techniques*, IEEE Transactions on Image Processing, 16 (2007), pp. 1058–1072. <https://doi.org/10.1109/TIP.2007.891777>.
- [4] A. BLAKE, *Boundary Conditions of Lightness Computation in Mondrian World*, Computer Vision Graphics and Image Processing, 32 (1985), pp. 314–327. [http://dx.doi.org/10.1016/0734-189X\(85\)90054-4](http://dx.doi.org/10.1016/0734-189X(85)90054-4).
- [5] V. CASELLES, B. COLL, AND J-M. MOREL, *Topographic maps and local contrast changes in natural images*, International Journal of Computer Vision, 33 (1999), pp. 5–27. <https://doi.org/10.1023/A:1008144113494>.
- [6] V. CASELLES, J.L. LISANI, J-M. MOREL, AND G. SAPIRO, *Shape preserving local histogram modification*, IEEE Transactions on Image Processing, 8 (1999), pp. 220–230. <https://doi.org/10.1109/83.743856>.

- [7] F. DRAGO, K. MYŠKOWSKI, T. ANNEN, AND N. CHIBA, *Adaptive logarithmic mapping for displaying high contrast scenes*, in Proceedings of EUROGRAPHICS, vol. 22, 2003.
- [8] F. DURAND AND J. DORSEY, *Fast bilateral filtering for the display of high-dynamic-range images*, ACM Transactions on Graphics, 21 (2002), pp. 257–266. <https://doi.org/10.1145/566654.566574>.
- [9] G. EILERTSEN, R. K. MANTIUK, AND J. UNGER, *A comparative review of tonemapping algorithms for high dynamic range video*, Computer Graphics Forum, State of the Art Reports, 36 (2017), pp. 565–592. <https://doi.org/10.1111/cgf.13148>.
- [10] R. FATTAL, D. LISCHINSKI, AND M. WERMAN, *Gradient domain high dynamic range compression*, ACM Transactions on Graphics, 21 (2002), pp. 249–256. <https://doi.org/10.1145/566654.566573>.
- [11] 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>.
- [12] B.K. HORN, *Determining Lightness from an Image*, Computer Graphics and Image Processing, 3 (1974), pp. 277–299. [http://dx.doi.org/10.1016/0146-664X\(74\)90022-7](http://dx.doi.org/10.1016/0146-664X(74)90022-7).
- [13] 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. <http://dx.doi.org/10.1109/83.597272>.
- [14] —, *Properties and performance of a center/surround retinex*, IEEE Transactions on Image Processing, 6 (1997), pp. 451–462. <http://dx.doi.org/10.1109/83.557356>.
- [15] R. KIMMEL, M. ELAD, D. SHAKED, R. KESHET, AND I. SOBEL, *A variational framework for Retinex*, International Journal of Computer Vision, 52 (2003), pp. 7–23. <http://dx.doi.org/10.1023/A:1022314423998>.
- [16] E.H. LAND, *The Retinex*, American Scientist, 52 (1964), pp. 247–264.
- [17] —, *The Retinex Theory of Color Vision*, Scientific American, 237 (1977), pp. 108–128.
- [18] —, *An alternative technique for the computation of the designator in the retinex theory of color vision*, Proceedings of the National Academy of Sciences, 83 (1986), pp. 3078–3080.
- [19] E.H. LAND AND J. McCANN, *Lightness and Retinex theory*, Journal of the Optical Society of America, 61 (1971), pp. 1–11.
- [20] N. LIMARE, J.L. LISANI, J-M. MOREL, A.B. PETRO, AND C. SBERT, *Simplest Color Balance*, Image Processing On Line, 1 (2011), pp. 297–315. <https://doi.org/10.5201/ipol.2011.11lmps-scb>.
- [21] N. LIMARE, A.B. PETRO, C. SBERT, AND J-M. MOREL, *Retinex Poisson Equation: a Model for Color Perception*, Image Processing On Line, 1 (2011), pp. 39–50. https://doi.org/10.5201/ipol.2011.lmps_rpe.
- [22] J.L. LISANI, *Adaptive local image enhancement based on logarithmic mappings*, in IEEE International Conference on Image Processing (ICIP), 2018, pp. 1747–1751. <https://doi.org/10.1109/ICIP.2018.8451655>.

- [23] 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>.
- [24] W. MA, J.M. MOREL, S. OSHER, AND A. CHIEN, *An L1-based variational model for retinex theory and its application to medical images*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 153–160. <http://dx.doi.org/10.1109/CVPR.2011.5995422>.
- [25] Z. MAI, H. MANSOUR, R. MANTIUK, P. NASIOPOULOS, R. WARD, AND W. HEIDRICH, *Optimizing a tone curve for backward-compatible high dynamic range image and video compression*, IEEE Transactions on Image Processing, 20 (2011), pp. 1558–1571. <https://doi.org/10.1109/TIP.2010.2095866>.
- [26] D. MARINI AND A. RIZZI, *A computational approach to color adaptation effects*, Image and Vision Computing, 18 (2000), pp. 1005–1014. [http://dx.doi.org/10.1016/S0262-8856\(00\)00037-8](http://dx.doi.org/10.1016/S0262-8856(00)00037-8).
- [27] J.M. MOREL, A.B. PETRO, AND C. SBERT, *A PDE formalization of retinex theory*, IEEE Transactions on Image Processing, 19 (2010), pp. 2825–2837. <http://dx.doi.org/10.1109/TIP.2010.2049239>.
- [28] N. MORONEY, *Local color correction using non-linear masking*, in IS&T/SID Eight Color Imaging Conference, 2000, pp. 108–111.
- [29] 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>.
- [30] 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 (2007), pp. 162–171. <https://doi.org/10.1109/TIP.2006.884946>.
- [31] R.E. WOODS R.C. GONZALEZ, *Digital Image Processing*, Addison Wesley, 1993. ISBN 0-201-50803-6.
- [32] E. REINHARD AND K. DEVLIN, *Dynamic range reduction inspired by photoreceptor physiology*, IEEE Transactions on Visualization and Computer Graphics, 11 (2005), pp. 13–24. <https://doi.org/10.1109/TVCG.2005.9>.
- [33] 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).
- [34] 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.
- [35] J. SERRA, *Image Analysis and Mathematical Morphology*, New York Academic, 1982. ISBN 0126372403.
- [36] —, *Image Analysis and Mathematical Morphology, Vol,2: Theoretical Advances*, New York Academic, 1988. ISBN 0126372411.

- [37] D. ZOSSO, G. TRAN, AND S. OSHER, *Non-local retinex—a unifying framework and beyond*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 787–826. <https://doi.org/10.1137/140972664>.
- [38] K. ZUIDERVELD, *Contrast limited adaptive histogram equalization*, in Graphic Gems IV, P.S. Heckbert, ed., Academic Press Professional, 1994, pp. 474–485. ISBN 0-201-50803-6.