2021/11/21 v0.6 IPOL article class

Published in Image Processing On Line on 2026–09–00.
Submitted on 2024–12–29, accepted on 2026–07–03.
ISSN 2105–1232 © 2026 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.2026.595

A Modified Structure of Local Adaptive Hysteresis Smoothing for Image Denoising

Ali Sabet Nasab, Reza P.R. Hasanzadeh, Pooriya Takhtechian

Department of Electrical Engineering, University of Guilan, Iran
Corresponding author email: hasanzadehpak@guilan.ac.ir

Communicated by Valéry Dewil *Demo edited by* Valéry Dewil

Abstract

Conventionally, Hysteresis Smoothing (HS)-based techniques rely on hysteresis thresholds to eliminate noise. The most efficient of the HS techniques, namely LAHS, performs well in image denoising; however, its structure has limitations in removing noise while preserving details. In this paper, a new framework, called Modified LAHS (MLAHS), is proposed to overcome these problems by modifying the cursor-width strategy and the HS process. The proposed method is evaluated quantitatively and qualitatively against other noise-reduction methods. The results show that the proposed structure significantly improves the performance of LAHS and also outperforms several efficient and widely used noise-suppression techniques.

Keywords: hysteresis smoothing; additive white Gaussian noise; image denoising; modified local adaptive hysteresis smoothing

Source Code

The source code of the algorithm and an online demonstration are accessible at the IPOL web page of this article[1]. Usage instructions are included in the README.txt file of the archive.

1 Introduction

Noise is an unavoidable and undesirable component in digital images that can significantly degrade image quality. It originates from a range of sources, including sensor limitations, environmental conditions, and compression techniques. Among the various noise models used in different noise suppression methods, additive white Gaussian noise (AWGN) is a simple and widely adopted model for approximating noise in natural images. Therefore, this work focuses on AWGN removal to validate the proposed method. Noise suppression, or denoising, is a critical step in image processing for several reasons. Effective denoising preserves important details and textures in the image, improving the accuracy of subsequent image processing tasks like segmentation, feature extraction, and object

[1] https://doi.org/10.5201/ipol.2026.595

recognition. Gaussian [12], Order Statistics [8], and Total Variation (TV) filters [21] are among the earliest methods developed for removing Gaussian noise. In Gaussian filtering, noise is assumed to be a high-frequency component, and a smoother image is created using a low-pass filter with weights defined by a Gaussian function [12]. In contrast, the basic idea behind Order Statistics filters is to sort the pixel values within a local neighborhood and select a specific order statistic, such as the minimum, maximum, or median value, as the output value for the center pixel of the neighborhood [8]. The main idea behind the TV filters is that regions of an image with fewer rapid changes in pixel values are smoother and less noisy. By minimizing the total variation, the filter enhances the image to have piecewise smooth regions with sharp transitions at the edges. Preserving fine image details along with the noise reduction process is the most important advantage of these kinds of methods [21]. While TV is simple and efficient, it can lead to issues such as blurring and staircase effects. In addition, the effectiveness of TV decreases significantly at high noise levels.

The Wiener filter is another fundamental noise suppression technique known for its effectiveness in reducing noise while being relatively simple to implement. This linear filter seeks to minimize the mean square error between the filtered and original signals. In the case of the Signal Adaptive Wiener (SAW) filter [19], the coefficients are adaptively adjusted for each region of the image, considering the local characteristics of both the signal and the noise. This adaptive approach enables the filter to respond more effectively to variations in signal and noise properties, leading to enhanced noise reduction and better preservation of signal details [19].

Transform-domain filters represent a distinct category of noise removal methods. Unlike spatial domain methods that operate directly on pixel values, transform domain techniques employ mathematical transformations, such as the Fourier or wavelet transform, to decompose an image into its transform-domain components. The primary advantage of these filters is their ability to efficiently target and reduce noise while preserving the integrity of important image details. The Discrete Wavelet Transform (DWT) [14] method is one of these transform-domain approaches. The DWT operates by decomposing an image into a set of wavelet coefficients using a collection of functions known as the wavelet basis, which consists of scaled and translated versions of a single prototype wavelet function. This filter applies a series of filtering and down-sampling operations to the input image to obtain the wavelet coefficients. Finally, the image denoising process is completed by transforming the processed coefficients back into the spatial domain [14]. Despite its strengths, transform domain filters can be computationally intensive and may introduce artifacts if not implemented precisely.

The Non-Local Means (NLM) algorithm is another efficient approach for noise removal [3]. Unlike traditional denoising methods, NLM considers the similarities between non-local patches instead of relying solely on local pixel neighborhoods. By averaging the values of similar patches, NLM effectively reduces noise while preserving important image features, such as edges and textures, which are crucial for maintaining image quality. However, NLM can be computationally expensive, especially for large images, due to the search and comparison operations involved [3].

Another class of noise-removal methods comprises hybrid filters, which combine multiple denoising techniques to produce superior noise reduction while preserving finer image details. These filters leverage the advantages of different approaches, such as spatial-domain filtering, transform-domain filtering, and redundancy of non-local patches. One of the most popular hybrid filtering techniques is the Block-Matching and 3D Filtering (BM3D) algorithm [4]. BM3D utilizes both spatial and frequency domain processing, first grouping similar image patches into 3D arrays and then applying collaborative filtering through 3D transformation, coefficient shrinkage, and inverse transformation [4]. This multi-step process allows for effective noise suppression while maintaining fine structures and textures. The success of the BM3D algorithm has led to the development of several versions and variants that aim to address specific challenges or enhance its denoising performance. Two notable examples are Block-matching and 4D filtering (BM4D) [5] and BM3D with Shape-Adaptive Principal Component Analysis (BM3D-SAPCA) [13].

In recent years, deep-learning methods have been developed for various applications, including image denoising [24, 22]. Convolutional Neural Networks (CNNs) leverage deep architectures to denoise images by learning feature representations through convolutional layers. The predominant training paradigm is supervised learning, which relies on the availability of pairs of clean and noisy ground-truth images. Pioneering methods like DnCNN [27] demonstrated that state-of-the-art performance could be achieved with a relatively small set of training pairs. However, subsequent, higher-capacity networks designed to improve upon these early methods often require larger and more diverse datasets of (clean, noisy) pairs, which can be complex and costly to acquire in real-world scenarios [6, 10].

Hysteresis Smoothing (HS)-based methods represent another category of spatial-domain noise-removal techniques. The Standard HS (SHS) method [17], initially proposed for one-dimensional signals, serves as the foundation for these techniques, which were later adapted for two-dimensional images in [23, 9]. The primary concept behind these filters is to employ a cursor with an upper and a lower threshold to suppress distortions while preserving essential structures, with the aim of effectively maintaining edges and important details while minimizing unwanted variations such as noise. This approach offers an efficient solution for noise suppression while also preserving the integrity of the original signal or image. One of the most recognized HS-based techniques is Local Adaptive HS (LAHS) [9], which adjusts the optimal cursor value based on the characteristics of the processing window. Consequently, studies [16, 20] have aimed to enhance HS-based noise removal quality by implementing strategies such as reducing the weight of windows containing edges and image details [16], incorporating an NLM approach [20], and using fuzzy-logic techniques [7]. Therefore, given LAHS's simplicity, efficiency, and potential for improving hysteresis-based methods, this paper aims to address its limitations and improve its competitiveness with more complex methods. In the proposed method, called Modified LAHS (MLAHS), we aim to improve denoising quality by varying the cursor size in areas with edges and details compared to smooth areas of the image. Additionally, in MLAHS, by introducing a new structure for the output values instead of using the middle points of the cursor, we improve the denoising performance, particularly at high noise levels.

The rest of this paper is organized as follows. Section 2 provides an overview of the hysteresis process. Section 3 details our proposed method. Section 4 presents experiments and results on various images. Finally, Section 5 concludes the work.

2 Hysteresis Smoothing Techniques

All HS-based filters derive from the Standard Hysteresis Smoothing (SHS) approach. In SHS, unwanted changes in a one-dimensional input signal are eliminated by using hysteresis thresholding, which is achieved through an upper and a lower threshold [17]. The upper threshold level (UTL) and lower threshold level (LTL) are determined by a parameter Δ, which defines half the width of a conceptual 'cursor'. The total cursor width (CW) is therefore 2Δ. In SHS, during horizontal movement along a line in the image, the intensity of each subsequent pixel is compared to the current cursor range. The output of the hysteresis process is the midpoint of the cursor. If a pixel's intensity falls outside the cursor range, the cursor moves to center on that new value. Conversely, if the pixel intensity is within the range, its value is replaced by the current cursor midpoint. Figure 1 shows how the SHS method is implemented. Assuming zero mean additive white Gaussian noise (AWGN), the noisy image, y, is modeled as follows

$$y = x + n, \tag{1}$$

where n is Gaussian noise, and x and y are the original and noisy images, respectively. Therefore, SHS is formulated as follows to estimate the noise-free image

$$\begin{aligned} UTL_0 &= m + \Delta, \\ LTL_0 &= m - \Delta. \end{aligned} \tag{2}$$

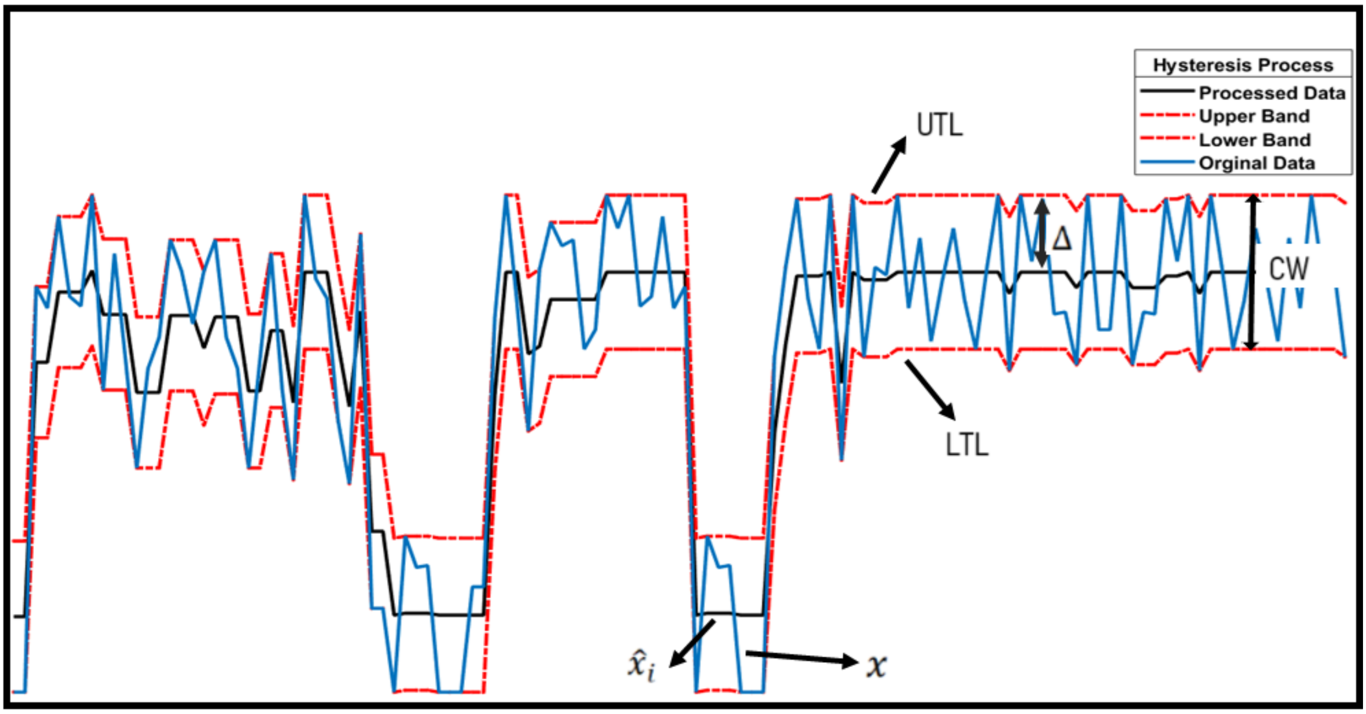


Figure 1: Illustration of SHS applied to one-dimensional noisy data.

$$\begin{aligned} UTL_i &= \begin{cases} y_i, & \text{if } y_i > \hat{x}_{i-1} + \Delta, \\ \hat{x}_{i-1} + \Delta, & \text{if } \hat{x}_{i-1} - \Delta \leq y_i \leq \hat{x}_{i-1} + \Delta, \\ y_i + 2\Delta, & \text{if } y_i < \hat{x}_{i-1} - \Delta, \end{cases} \qquad i \neq 0, \\ LTL_i &= \begin{cases} y_i - 2\Delta, & \text{if } y_i > \hat{x}_{i-1} + \Delta, \\ \hat{x}_{i-1} - \Delta, & \text{if } \hat{x}_{i-1} - \Delta \leq y_i \leq \hat{x}_{i-1} + \Delta, \\ y_i, & \text{if } y_i < \hat{x}_{i-1} - \Delta, \end{cases} \qquad i \neq 0, \end{aligned} \tag{3}$$

where UTL_i and LTL_i are the upper and lower trigger levels, and m is the mean or median value of the local window. In our work, the median is chosen because its robustness to outliers helps preserve edge information in noisy environments better than the mean. In addition, CW=2Δ and $\hat{x}_0$=m. Then, the value of $\hat{x}_i$ can be calculated as follows,

$$\hat{x}_i = \begin{cases} \hat{x}_{i-1}, & \text{if } LTL < y_i < UTL, \\ y_i \pm \Delta, & \text{if } y_i < LTL \text{ or } y_i > UTL, \end{cases} \qquad i \neq 0. \tag{4}$$

where y$_i$ and $\hat{x}_i$ are the noisy and denoised values estimated for the i-th pixel, respectively. Due to the limitation of cursor movement in one direction, SHS does not lead to good results for image noise removal. The complex hysteresis smoothing (CHS) method [23] addresses this problem by expanding cursor movement to two dimensions, allowing for better preservation of image structure. However, it is time-consuming to execute, and no optimal cursor configuration has been proposed [23]. To overcome the above-mentioned issues, Local HS (LHS) [9] has been introduced. Instead of applying

HS to the entire image, LHS performs it locally using 3×3 and 5×5 windows. Various scanning paths have been proposed for processing local windows, among which the *spiral* pattern shown in Figure 2 yields the best results [1, 9]. Additionally, LHS introduces a new value for the CW, assuming that both the image and noise are constant within the local windows [9]. Because the above assumption does not necessarily hold, LAHS was introduced to exploit the non-stationary statistical properties of images in local windows to determine the optimal Δ value [16].

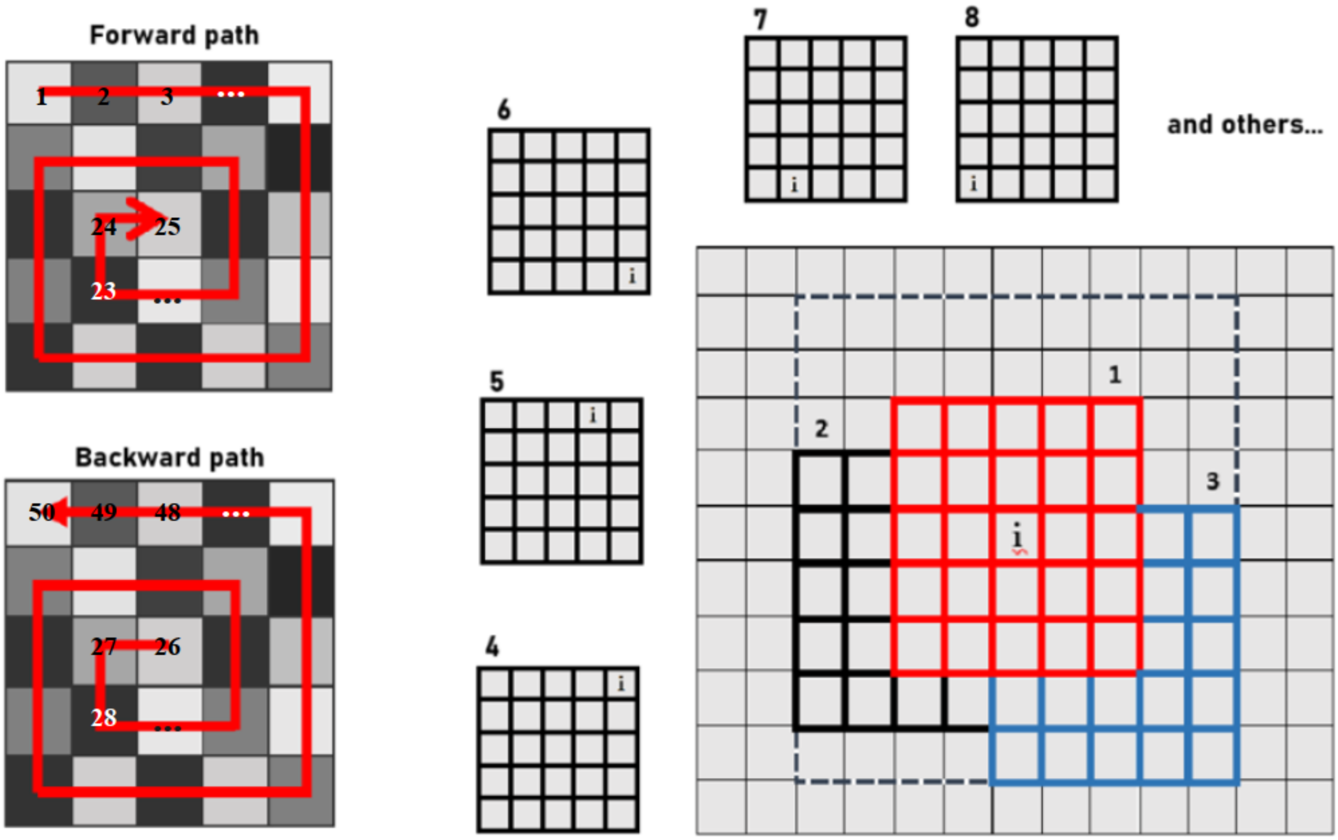


Figure 2: (Left) The *spiral* scanning pattern in a 5×5 local window, with the forward and backward paths indicated by the up and down figures, respectively. (Right) The *spiral* pattern in overlapping states for the target pixel, *i*. The overlapping states are represented by colored squares (1, 2, 3 and . . ., T), with each state acting as a window that influences the target pixel. Let P be the number of scan paths, r the radius of the local window, and L the number of local estimates per pixel. The total number of estimates for each pixel in the image is given by the equation $T = L + 4Pr(r + 1)$. The average of these T values is used to obtain the final estimate. According to this equation, with a 5×5 local window and *spiral* pattern, the value of P=2, L=2 and r=2, a total of T=50 estimates are calculated for each pixel.

As previously mentioned, the most crucial factor in the HS process is the determination of the optimal size and position of the CW=2Δ. A large value of Δ degrades edges and details, and a small value preserves noise in the image. For this purpose, in LAHS, according to the local statistical characteristics of the image, Δ is suggested as follows [9]

$$\Delta = \frac{\sigma_n^2}{\sigma_y}, \tag{5}$$

where σ_n^2 denotes the variance of noise, and σ_y represents the standard deviation of the noisy image, y, within the local window being processed. As can be seen in (5), if the local window contains edges or details, the value of σ_y increases, as a result, Δ decreases and less smoothing will occur, but if the window contains smooth areas, the size of the cursor increases and as a result we will have more smoothing.

The complete LAHS process applies the adaptive hysteresis operator over multiple overlapping local windows using the *spiral* scanning pattern in both forward and backward directions, as illustrated in Figure 2. Since the LAHS method is applied locally, let P, r, and L be the number of scan paths, the radius of the local window, and the number of local estimates per pixel, respectively. Therefore, the total number of estimates for each pixel is given by $T = L + 4Pr(r+1)$. Half of these estimates correspond to the forward scanning path, while the other half correspond to the backward scanning path. The final pixel estimate is obtained by averaging all these values [9].

3 Proposed Method

The adaptive cursor in LAHS, calculated via Equation (5), achieves a fundamental trade-off: a smaller Δ preserves edges but reduces its denoising effectiveness, while a larger Δ increases smoothing but may blur fine image details. This balance, although effective, represents the core limitation that our modifications seek to overcome. The proposed method, called Modified LAHS (MLAHS), consists of two parts. The classical LAHS approach is characterized by (1) a single-rate cursor adjustment in the smoothing process and (2) the absence of local-statistics calculations when computing the output value if the pixel value falls outside the cursor. Subsection 3.1 presents a new approach by using a two-rate scaling of the CW to improve the noise suppression process in smooth areas. Also, Subsection **??** proposes a new output structure to solve noise removal problems in areas with edges and details.

3.1 MLAHS Cursor-Width Optimization

To address the inherent disadvantage of the trade-off in the single-rate cursor of LAHS, we introduce a two-rate scaling that dynamically adjusts the cursor width based on local content. To distinguish smooth regions from regions containing edges and details while preserving their structure, the new values for UTL and LTL are proposed as follows

$$\begin{aligned} UTL_0 &= m + \alpha_1 \Delta, \\ LTL_0 &= m - \alpha_1 \Delta. \end{aligned} \tag{6}$$

$$\begin{aligned} UTL_i &= \begin{cases} y_i, & \text{if } y_i > \hat{x}_{i-1} + \Delta, \\ \hat{x}_{i-1} + \alpha_2 \Delta, & \text{if } \hat{x}_{i-1} - \Delta \le y_i \le \hat{x}_{i-1} + \Delta, \\ y_i + 2\alpha_2 \Delta, & \text{if } y_i < \hat{x}_{i-1} - \Delta, \end{cases} \\ LTL_i &= \begin{cases} y_i - 2\alpha_2 \Delta, & \text{if } y_i > \hat{x}_{i-1} + \Delta, \\ \hat{x}_{i-1} - \alpha_2 \Delta, & \text{if } \hat{x}_{i-1} - \Delta \le y_i \le \hat{x}_{i-1} + \Delta, \\ y_i, & \text{if } y_i < \hat{x}_{i-1} - \Delta, \end{cases} \qquad i \neq 0. \end{aligned} \tag{7}$$

where α_1 controls the cursor at the beginning of processing each window, and α_2 ($\alpha_2 < \alpha_1$) controls the cursor as it encounters regions containing edges and details. Also, m is the median of pixel intensity values over a local window. As can be seen in (6) and (7), the cursor width at the beginning of the hysteresis process in each window is $CW = 2\alpha_1\Delta$. After encountering the first edge or detail, it decreases to $CW = 2\alpha_2\Delta$. This two-stage cursor-width strategy allows for better denoising in the smooth areas of the image, and also preserves edges and details properly.

3.2 MLAHS Output Structure

By examining (4), we find that during the hysteresis process, any input (y_i) that falls outside the cursor range can adversely affect the output. In this case, the output ($\hat{x}_i$) is equal to the input, which is modified only by a constant term $\pm\Delta$. Hence, any input beyond the cursor, even if due to noise, significantly different from the previous one ($\hat{x}_{i-1}$), is only modified by Δ. Therefore, in this situation, the value of the previous output has no effect on the new output. This is a major drawback of the hysteresis structure. To address this problem, we introduce a structure that accounts for the difference between each input and the average of the corresponding window.

Based on the second-order statistics of the spatial Wiener estimation given by [11], the output in MLAHS is modified as follows

$$\hat{x}_i = \begin{cases} \hat{x}_{i-1}, & \text{if } LTL < y_i < UTL, \\ y_i + \dfrac{\sigma_n^2}{\sigma_y^2}\left(\overline{y} - y_i\right), & \text{if } y_i < LTL \text{ or } y_i > UTL, \end{cases} \qquad i \neq 0. \tag{8}$$

where $\overline{y}$ is the mean intensity of the local window. As mentioned in (8), any input (y_i) that is outside the width of the cursor is modified by the three factors σ_n^2, σ_y^2, and the difference between each input and the average of the local window. Unlike the LAHS method, which uses a fixed term to calculate the output, this method takes advantage of the difference between each input and the continuously updated local-window mean. The amount of this modification is controlled by $\frac{\sigma_n^2}{\sigma_y^2}$. Therefore, in the MLAHS method, a strong effect of noise on an input does not cause the output to be oversmoothed. Algorithm 1 summarizes the MLAHS procedure.

4 Experimental Results

4.1 Performance Evaluation Metrics

The objective quality of denoising results is evaluated using the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) [26]. The PSNR (in dB) between the original image x and the denoised image $\hat{x}$ is computed as

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right), \quad \text{where} \quad \text{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\hat{x}(i,j) - x(i,j)\right)^2.$$

The SSIM index is calculated as defined in [26],

$$\text{SSIM}(x,\hat{x}) = \frac{(2\mu_x\mu_{\hat{x}} + c_1)(2\sigma_{x\hat{x}} + c_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)},$$

where μ_x, $\mu_{\hat{x}}$ are local means, σ_x, $\sigma_{\hat{x}}$ are standard deviations, $\sigma_{x\hat{x}}$ is the cross-covariance, and c_1, c_2 are stabilization constants.

The quantitative evaluation is supplemented by visual comparisons, including the inspection of zoomed regions and one-dimensional line-scan plots, to illustrate the performance in preserving edges and textures.

4.2 MLAHS Parameter Determination

The parameters α_1 and α_2 in the proposed algorithm are determined empirically by maximizing the PSNR over several noisy images. These parameters are crucial because they control the two-stage cursor width adaptation of the algorithm, and inappropriate values may adversely affect edge

Algorithm 1: MLAHS Algorithm

Require: Degraded image y with size $M \times N$, noise variance σ_n^2, parameters α_1 and α_2
Ensure: Denoised image $\hat{x}$

Select spiral scanning pattern with $P = 2$, $L = 2$, $r = 2$
Local window size: $(2r + 1) \times (2r + 1) = 5 \times 5$, $T = 50$ (as in Figure 2)
for $a = 1$ to M **do**
  for $b = 1$ to N **do**
    Calculate local variance σ_y^2, local mean $\overline{y}$, and initial m (median) in window centered at $y(a, b)$
    Calculate $\Delta = \frac{\sigma_n^2}{\sigma_y}$ using Equation (5)
    Calculate UTL_0 and LTL_0 using Equation (6) with α_1 from Table 1
    for $k = -r$ to r **do**
      for $l = -r$ to r **do**
        Based on sequence $i = 1, \ldots, T$ and indices k, l indicating spiral order (Figure 2)
        for $i = 1$ to T **do**
          if $y(a - l(i), b - k(i)) < LTL$ **then**
            Compute UTL and LTL using Equation (7) with α_2 from Table 1
            $LTL = y(a - l(i), b - k(i))$
            $UTL = y(a - l(i), b - k(i)) + (2 \cdot \alpha_2 \cdot \Delta)$
            $m = y(a - l(i), b - k(i)) + \frac{\sigma_n^2}{\sigma_y^2}(\overline{y} - y(a - l(i), b - k(i)))$
          else if $y(a - l(i), b - k(i)) > UTL$ **then**
            $UTL = y(a - l(i), b - k(i))$
            $LTL = y(a - l(i), b - k(i)) - (2 \cdot \alpha_2 \cdot \Delta)$
            $m = y(a - l(i), b - k(i)) + \frac{\sigma_n^2}{\sigma_y^2}(\overline{y} - y(a - l(i), b - k(i)))$
          else if $LTL < y(a - l(i), b - k(i)) < UTL$ **then**
            $m = m$ {no change}
          end if
          $\hat{x}(a - l(i), b - k(i)) = \hat{x}(a - l(i), b - k(i)) + m$
        end for
      end for
    end for
    $\hat{x}(a, b) = \hat{x}(a, b)/T$
  end for
end for
return $\hat{x}$

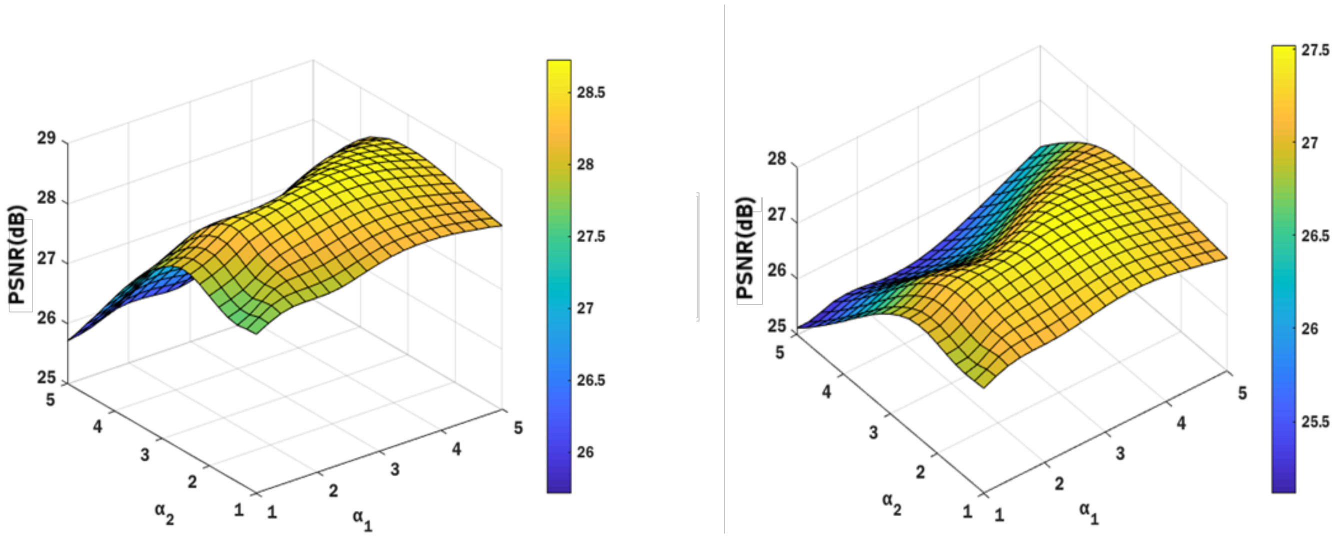
preservation and overall denoising performance. To ensure a thorough search, we explored a grid of values with a step of 0.2 for both parameters, covering the range $[1, 5]$ for each parameter. To achieve the maximum PSNR, we used a set of standard images shown in Figure 3, including Pirate, House, Hill, and Living, corrupted by AWGN at various noise levels ($\sigma_n = 10, 20, 30, 40, 50$). Figures 4 and 5 show four representative examples of the obtained PSNR surfaces for two noise levels ($\sigma_n = 30$ and 50). As illustrated in these figures, the two-rate cursor width strategy with the optimal coefficients α_1 and α_2 outperforms a single-width cursor setting. The evaluations show consistent behavior of the optimal parameters across different images. The overall results from the complete set of images indicate that the optimal values of α_1 and α_2 are concentrated in narrow intervals, demonstrating low sensitivity to image content and noise level. The optimal ranges for α_1 and α_2, which determine the cursor width, are presented in Table 1. According to these experiments, $\alpha_1 = 3.4$ and $\alpha_2 = 2.2$

Figure 3: The benchmark images used in the parameter optimization. From left to right: Pirate, House, Hill, and Living.

are identified as appropriate values for various noise levels. To limit potential overfitting during parameter selection and ensure a fair evaluation, parameter optimization was performed on a small set of images. The images used for the final evaluation in Section 4.3 are different from those used for parameter optimization.

Level of noise	α_1	α_2
Optimal range	[3 3.6]	[2 2.6]
Our choice	3.4	2.2

Table 1: Optimal values of parameters for MLAHS.

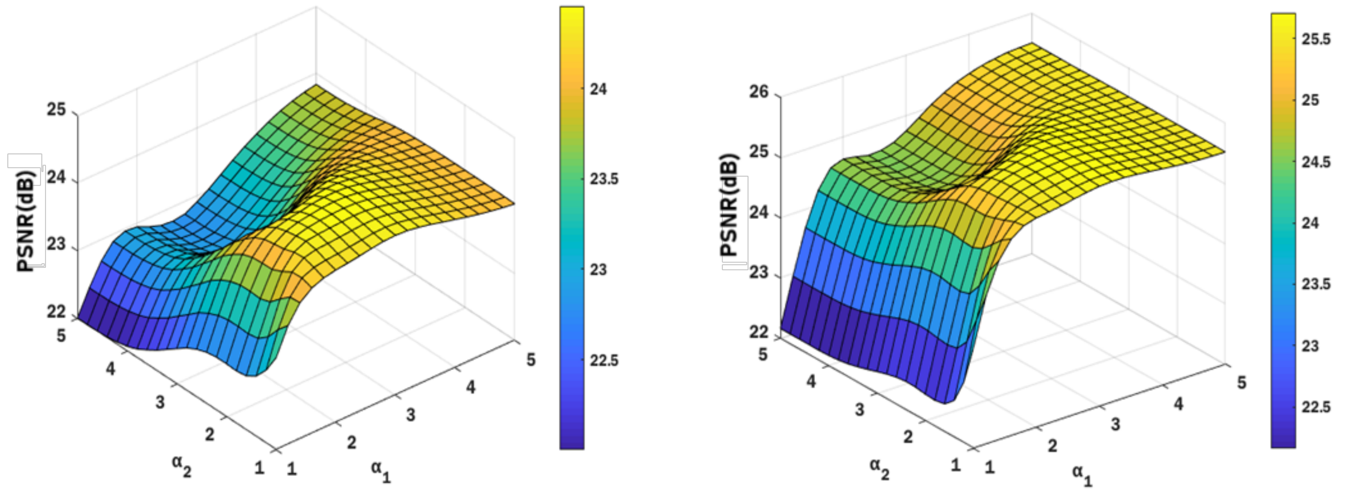


(a) (b)

Figure 4: PSNR surface of MLAHS for different values of α_1 and α_2 evaluated on the images (a) House and (b) Pirate corrupted by AWGN with $\sigma_n = 30$. The PSNR peak and the corresponding optimal parameter pair [α_1, α_2] are (a) 28.73 and [3.4, 2.6] and (b) 27.52 and [3.4, 2.4]. The PSNR results of LAHS for (a) and (b) are 28.1 and 27.01 respectively.

4.3 Experimental Results

To validate the proposed MLAHS, we evaluate it on the benchmark grayscale images shown in Figure 6, including Cameraman, Montage, Monarch, and Peppers. To further demonstrate the robustness and generalizability of the proposed method, we also extend our evaluation to the larger CBSD68 color image dataset [15]. This dataset consists of 68 natural color images, providing a more diverse and challenging benchmark.



(a) (b)

Figure 5: PSNR surface of MLAHS for different values of α_1 and α_2 evaluated on the images (a) Living Room and (b) Hill corrupted by AWGN with $\sigma_n = 50$. The PSNR peak and the corresponding optimal parameter pair [α_1, α_2] are (a) 24.45 and [3.4, 2.2] and (b) 25.71 and [3.2, 2.2]. The PSNR results of LAHS for (a) and (b) are 25.05 and 23.90 respectively.

Figure 6: The benchmark images used in the evaluation. From left to right: Cameraman, Monarch, Montage, and Peppers.

We select LAHS [9], NLAHS [20], LEE [11], GOA [25], FGP [2], NLM [3], and ProbShrink [18] for comparison. Recent advanced methods such as BM3D [4] and DnCNN [27] achieve significantly better denoising quality but involve considerably greater computational complexity. Since MLAHS relies only on spatial features, we primarily compare it with methods belonging to the same general class. In addition, because LAHS serves as the basis for extensions such as NLAHS [20], we apply the proposed modifications to NLAHS to assess their compatibility with previous improvements. We refer to this modified version as Modified NLAHS (MNLAHS).

As depicted in Table 2, the parameters for all algorithms are configured to their optimal values as recommended in their respective reference articles. All experiments are performed in MATLAB R2018b on a laptop with an Intel Core i7-4510U CPU @ 2.00GHz and 8.0 GB RAM.

The aforementioned reference images are contaminated with Gaussian noise at levels $\sigma_n = 10, 20, 30, 40,$ and 50. The PSNR and SSIM values over all images are reported in Tables 3 and 4, while sample visual results for two grayscale images and three color images are shown in Figures 7–11.

An analysis of Tables 3 and 4 reveals a subtle behavior: at the lowest noise level ($\sigma_n = 10$), the PSNR gain of MLAHS over the original LAHS is marginal. This result is expected and stems from the core design of the hysteresis process. Under low-noise conditions, the standard LAHS cursor is already small and near-optimal for removing faint noise without harming image details, whereas the mechanism introduced in MLAHS has minimal spurious noise to correct. Consequently, their activation may occasionally lead to negligible over-smoothing or slight adjustment of legitimate fine

Filter	Parameters
GOA [25]	Best result after 5 iterations
NLM [3]	5×5 similarity windows and 11×11 search windows
LEE [11]	5×5 windows
FGP [2]	Regularization parameter λ=0.08 and iteration=50
PBshrink [18]	7×7 windows
LAHS [9] and MLAHS	Spiral scanning pattern with 5×5 windows
NLAHS [20] and MNLAHS	Spiral scanning pattern with 3×3 windows, h = $0.9\sigma_n$ and 21×21 search windows

Table 2: Experimental parameters.

details, resulting in similar or slightly lower PSNR values. However, in the MNLAHS version, the incorporation of non-local data redundancy enables the proposed method to consistently outperform NLAHS in terms of PSNR, thereby providing superior overall performance.

To assess the preservation of image structure, the SSIM values for the proposed methods and other techniques at various noise levels are provided in Table 3. Notably, MNLAHS shows a relative advantage in preserving image structure in most cases.

The computational times of the proposed method and other image denoising methods are shown in Table 5. In this comparison, a set of images with different dimensions were used and the average results were reported. According to the results, MLAHS has a runtime comparable to, and in some cases slightly lower than, that of LAHS under similar conditions, while this technique can offer superior noise removal quality.

The noise reduction results for the Cameraman and Peppers images can be seen in Figures 7 and 8. As can be seen, the proposed method not only achieves better quantitative results but also provides good visual quality and effectively preserves details while suppressing noise. To further validate the effectiveness of the proposed method on color images, we present denoising results at a high noise level. Figures 9, 10, and 11 show results for additional natural scenes. Additionally, Figure 12 shows the outputs of MLAHS, LAHS, NLAHS, and MNLAHS for a horizontal line-scan in the Monarch image. The obtained curves demonstrate the superiority of the proposed method in following the original intensity profile. By examining the points marked with yellow ovals on the curves, we can conclude that the MLAHS and MNLAHS approaches exhibit better stability at the edges and in areas of image transition, providing a more appropriate response to sudden changes in the signal.

5 Conclusion

In this paper, we modified the LAHS structure to improve image denoising. We utilize two distinct cursor widths during the hysteresis process. Additionally, since the output in the LAHS method does not consider neighboring pixels, we have proposed a new framework so that the effect of strong noise on each pixel is mitigated using simple local statistics from the corresponding window. The results demonstrate that MLAHS significantly improves both noise removal and detail preservation while maintaining the simplicity of the LAHS method. In addition, as shown in the experimental results, the proposed modifications are also very effective for the extended LAHS variant NLAHS [20].

Filter			**Cameraman**		
σ_n	**10**	**20**	**30**	**40**	**50**
Noisy Image	28.31/0.6386	22.47/0.4104	19.06/0.2984	16.67/0.2323	14.90/0.1884
LEE	26.88/0.7932	26.23/0.7414	24.82/0.6146	22.75/0.4636	20.56/0.3478
GOA	32.00/0.8795	28.17/0.8072	26.13/0.7319	24.54/0.6632	22.97/0.5765
FGP	27.65/0.8079	27.29/0.8033	26.48/0.7718	24.80/0.6378	22.62/0.4790
NLM	32.04/0.9017	29.41/0.8121	27.41/0.7278	25.57/0.6487	23.96/0.5790
ProbShrink	32.91/0.9055	29.00/0.8194	26.91/0.7619	25.52/0.7058	**24.55**/0.6714
LAHS	33.06/0.9081	28.84/0.8069	26.32/0.7216	24.37/0.6575	22.83/0.6010
NLAHS	33.46/0.9148	29.70/**0.8444**	27.45/0.7967	25.62/0.7532	24.41/0.7137
MLAHS(ours)	32.96/0.9141	28.93/0.8228	26.95/0.7576	25.32/0.6953	23.76/0.6248
MNLAHS(ours)	**33.53/0.9150**	**29.85**/0.8432	**27.83/0.7995**	**26.02/0.7571**	24.54/**0.7157**
Filter			**Monarch**		
σ_n	**10**	**20**	**30**	**40**	**50**
Noisy Image	28.16/0.7294	22.17/0.5081	18.78/0.3853	16.47/0.3083	14.77/0.2551
LEE	28.23/0.8771	27.04/0.8198	24.80/0.6772	22.16/0.5228	19.74/0.4094
GOA	32.56/0.9306	28.71/0.8692	26.82/0.8337	25.38/0.7775	23.97/0.7150
FGP	28.23/0.8714	27.71/0.8608	26.67/0.8256	24.93/0.7194	22.79/0.5825
NLM	32.64/0.9403	29.61/0.8729	27.54/0.8015	25.93/0.7358	24.48/0.6778
ProbShrink	33.02/0.9406	28.87/0.8846	26.61/0.8317	25.23/0.7944	24.19/0.7591
LAHS	32.70/0.9330	28.56/0.8622	26.29/0.8016	24.40/0.7458	22.67/0.6895
NLAHS	**33.70**/0.9466	29.76/0.8966	27.49/0.8504	25.71/0.8061	24.33/0.7652
MLAHS(ours)	32.33/0.9367	28.34/0.8786	26.45/0.8276	25.15/0.7801	23.85/0.7235
MNLAHS(ours)	33.66/**0.9468**	**29.85/0.8979**	**27.76/0.8541**	**26.07/0.8114**	**24.62/0.7691**
Filter			**Montage**		
σ_n	**10**	**20**	**30**	**40**	**50**
Noisy Image	28.26/0.6094	22.57/0.3742	19.30/0.2670	17.01/0.2060	15.28/0.1663
LEE	28.28/0.8769	27.46/0.8217	25.96/0.7092	23.94/0.5583	21.79/0.4229
GOA	32.36/0.8975	29.09/0.8756	26.83/0.8143	25.14/0.7212	23.05/0.6075
FGP	30.78/0.9092	29.92/0.8902	28.36/0.8462	26.04/0.7201	23.45/0.5519
NLM	34.09/0.9398	31.28/0.8749	28.98/0.8070	26.83/0.7400	24.88/0.6777
ProbShrink	34.71/0.9326	30.33/0.8581	28.06/0.8098	26.20/0.7636	24.63/0.6714
LAHS	34.74/0.9392	30.42/0.8680	27.75/0.7950	25.75/0.7314	23.99/0.6780
NLAHS	36.03/0.9498	32.03/0.9037	29.62/0.8650	27.69/0.8309	25.92/0.7991
MLAHS(ours)	34.51/0.9462	30.54/0.8920	28.33/0.8344	26.80/0.7789	25.45/0.7237
MNLAHS(ours)	**36.08/0.9504**	**32.25/0.9058**	**29.92/0.8685**	**28.19/0.8359**	**26.64/0.8054**
Filter			**Peppers**		
σ_n	**10**	**20**	**30**	**40**	**50**
Noisy Image	28.18/0.6925	22.22/0.4468	18.79/0.3188	16.44/0.2427	14.71/0.1929
LEE	29.24/0.8673	27.98/0.8068	25.51/0.6471	22.58/0.4731	19.96/0.3494
GOA	31.66/0.9009	28.30/0.8276	26.51/0.7820	25.14/0.7334	23.57/0.6424
FGP	29.14/0.8558	28.44/0.8374	27.34/0.7971	25.46/0.6800	23.12/0.5291
NLM	33.53/0.9169	30.22/0.8429	27.95/0.7666	26.19/0.6952	24.75/0.6341
ProbShrink	33.60/0.9103	29.86/0.8505	27.66/0.7963	26.11/0.7526	24.94/0.7271
LAHS	33.14/0.9097	29.20/0.8339	26.72/0.7652	24.84/0.7083	23.40/0.6605
NLAHS	**33.97/0.9229**	30.23/0.8678	27.84/0.8167	26.05/0.7696	24.71/0.7282
MLAHS(ours)	32.86/0.9104	29.27/0.8504	27.28/0.8004	25.67/0.7447	24.33/0.6945
MNLAHS(ours)	33.95/0.9232	**30.42/0.8702**	**28.16/0.8210**	**26.40/0.7743**	**25.04/0.7327**

Table 3: PSNR/SSIM results of the proposed method compared with various well-known algorithms. The best results are shown in boldface.

(a) Original image $\infty/1$

(b) Noisy image 19.06/0.2984

(c) Lee 24.82/0.6146

(d) GOA 26.13/0.7319

(e) FGP 26.48/0.7718

(f) NLM 27.41/0.7278

(g) LAHS 26.32/0.7216

(h) MLAHS 26.95/0.7576

(i) ProbShrink 26.91/0.7619

(j) NLAHS 27.45/0.7967

(k) MNLAHS 27.83/0.7995

Figure 7: The results of the proposed method in comparison with several noise suppression methods. (a) Original image, (b) Noisy image ($\sigma_n = 30$), (c) Lee, (d) GOA, (e) FGP, (f) NLM, (g) LAHS, (h) MLAHS, (i) ProbShrink, (j) NLAHS, and (k) MNLAHS. The PSNR/SSIM values are shown below the images.

(a) Original image $\infty/1$ (b) Noisy image 18.79/0.3188

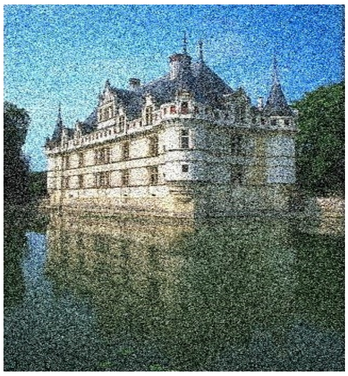
(c) Lee 25.51/0.6471 (d) GOA 26.51/0.7820 (e) FGP 27.34/0.7971

(f) NLM 27.95/0.7666 (g) LAHS 26.72/0.7652 (h) MLAHS 27.28/0.8004

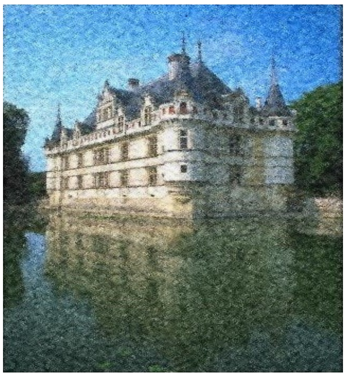
(i) ProbShrink 27.66/0.7963 (j) NLAHS 27.84/0.8167 (k) MNLAHS 28.16/0.8210

Figure 8: The results of the proposed method in comparison with several noise suppression methods. (a) Original image, (b) Noisy image ($\sigma_n = 30$), (c) Lee, (d) GOA, (e) FGP, (f) NLM, (g) LAHS, (h) MLAHS, (i) ProbShrink, (j) NLAHS, and (k) MNLAHS. The PSNR/SSIM values are shown below the images.

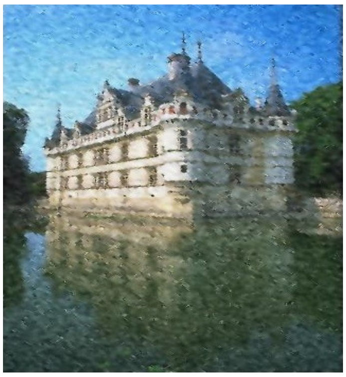
(a) Original image $\infty/1$



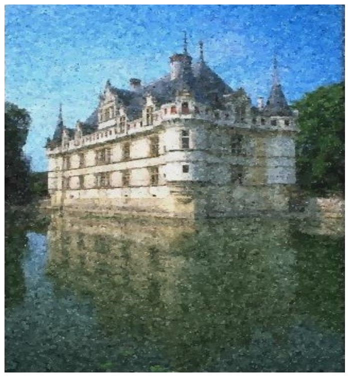
(b) Noisy image 15.00/0.36



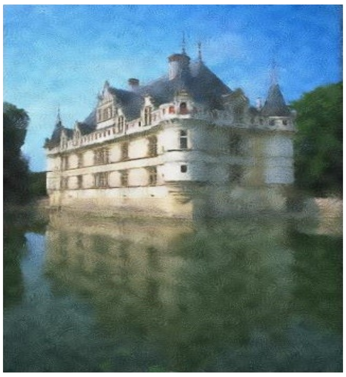
(c) Lee 20.80/0.5903



(d) GOA 22.87/0.7181



(e) FGP 22.96/0.7039



(f) NLM 23.96/0.7767

(g) LAHS 23.39/0.7574

(h) MLAHS 24.10/0.7782

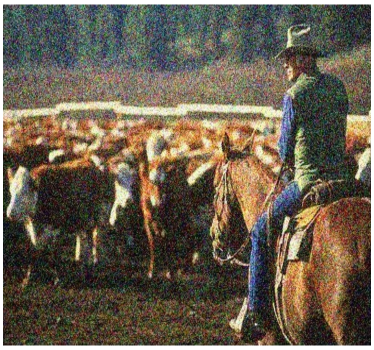
(i) ProbShrink 22.99/0.7692

(j) NLAHS 24.54/0.8241

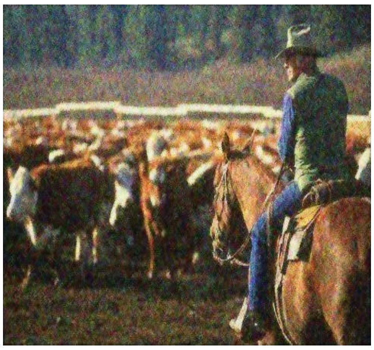
(k) MNLAHS 24.64/0.8258

Figure 9: House from the CBSD68 dataset. (a) Original image, (b) Noisy image ($\sigma_n = 50$), (c) Lee, (d) GOA, (e) FGP, (f) NLM, (g) LAHS, (h) MLAHS, (i) ProbShrink, (j) NLAHS, and (k) MNLAHS. The PSNR/SSIM values are shown below the images.

(a) Original image $\infty/1$



(b) Noisy image 15.12/0.3120



(c) Lee 20.48/0.5472

(d) GOA 23.35/0.7464

(e) FGP 22.96/0.6971

(f) NLM 24.18/0.7872

(g) LAHS 23.34/0.7845

(h) MLAHS 24.29/0.8125

(i) ProbShrink 23.05/0.7672

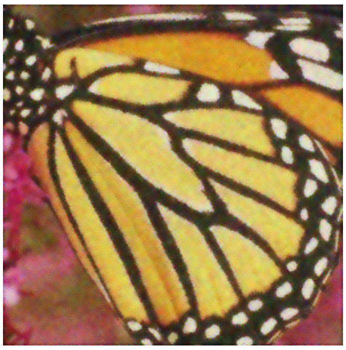
(j) NLAHS 24.68/0.8556

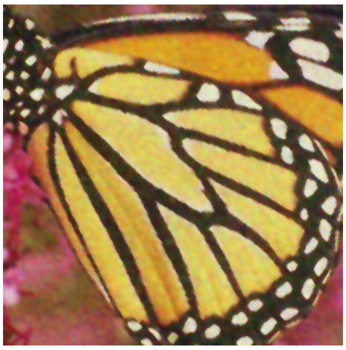
(k) MNLAHS 24.82/0.8580

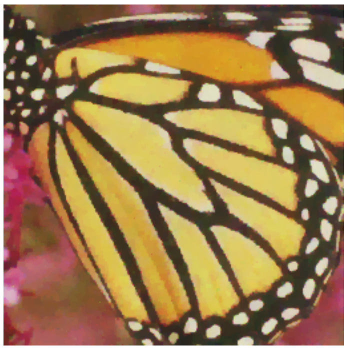
Figure 10: Image from the CBSD68 dataset. (a) Original image, (b) Noisy image ($\sigma_n = 50$), (c) Lee, (d) GOA, (e) FGP, (f) NLM, (g) LAHS, (h) MLAHS, (i) ProbShrink, (j) NLAHS, and (k) MNLAHS. The PSNR/SSIM values are shown below the images.

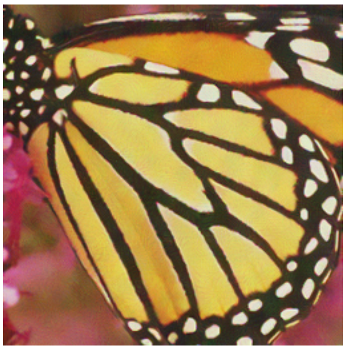

(a) Original image ∞/1

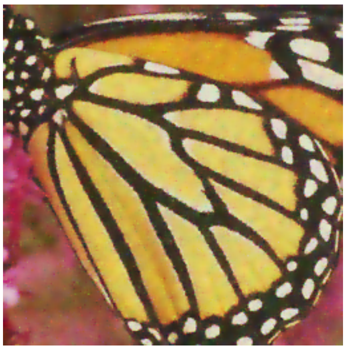

(b) Noisy image 19.01/0.7857

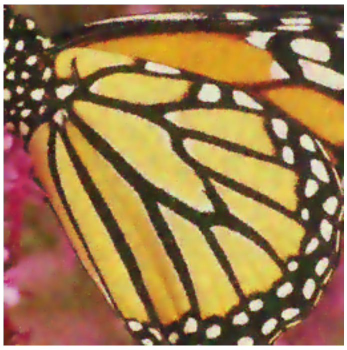

(c) Lee 23.93/0.9193


(d) GOA 25.14/0.9389


(e) FGP 25.61/0.9489


(f) NLM 26.97/0.9550


(g) LAHS 25.23/0.9410


(h) MLAHS 25.43/0.9432


(i) ProbShrink 24.14/0.9159


(j) NLAHS 26.55/0.9558


(k) MNLAHS 26.88/0.9582

Figure 11: Part of the Monarch image. (a) Original image, (b) Noisy image ($\sigma_n = 30$), (c) Lee, (d) GOA, (e) FGP, (f) NLM, (g) LAHS, (h) MLAHS, (i) ProbShrink, (j) NLAHS, and (k) MNLAHS. The PSNR/SSIM values are shown below the fragments.

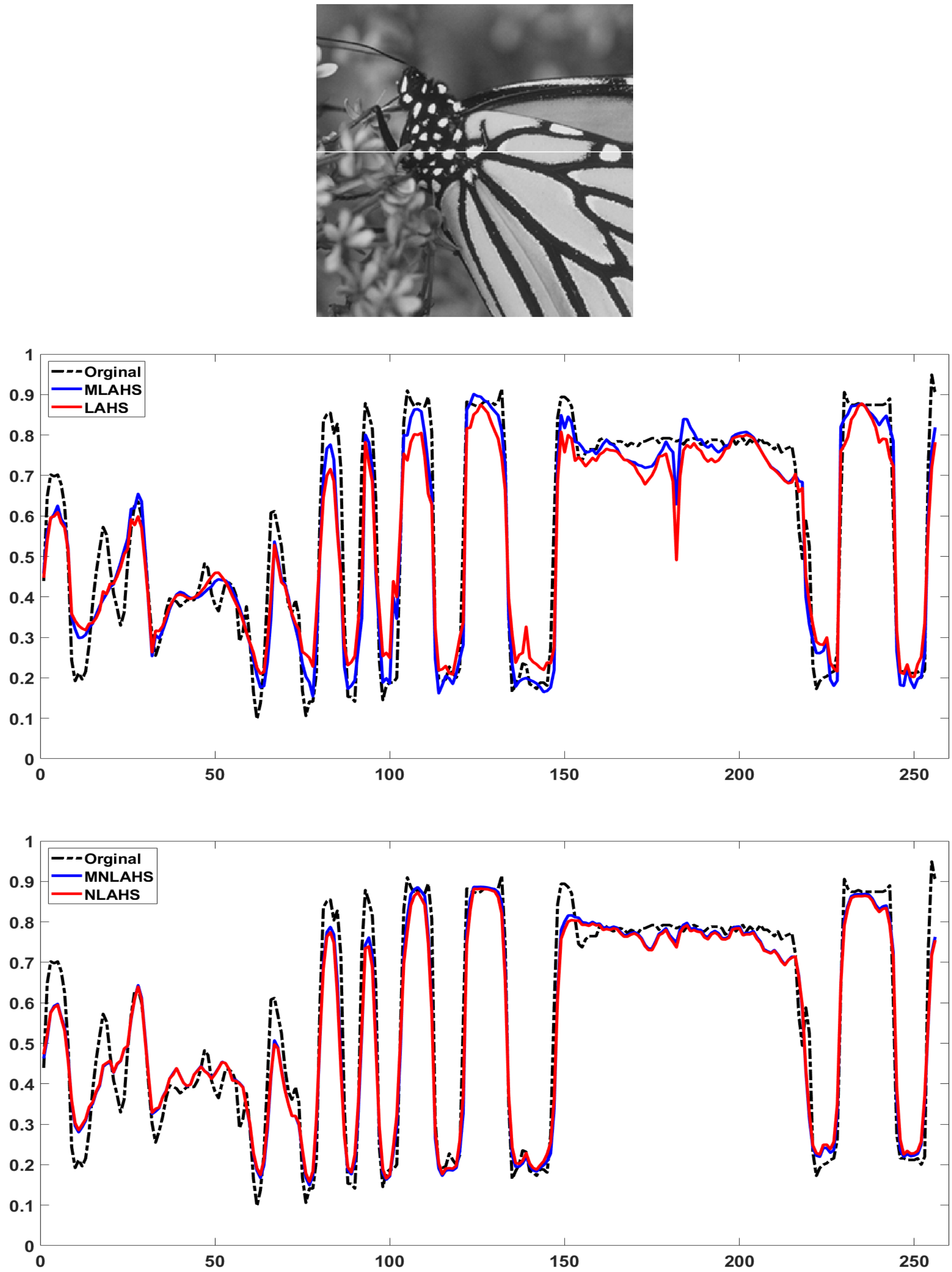


Figure 12: Outputs of denoising algorithms on the highlighted line of the Monarch image ($\sigma_n = 30$). (a) MLAHS and LAHS, (b) MNLAHS and NLAHS.

Method	$\sigma = 10$	$\sigma = 20$	$\sigma = 30$	$\sigma = 40$	$\sigma = 50$
Noisy Image	28.41/0.8216	22.56/0.6187	19.21/0.4780	16.91/0.3806	15.19/0.3105
LEE	27.83/0.8896	26.84/0.8535	24.86/0.7665	22.54/0.6505	20.36/0.5390
GOA	32.04/0.9448	27.76/0.8734	25.94/0.8230	24.48/0.7701	22.99/0.7044
FGP	27.56/0.8860	27.12/0.8744	26.30/0.8492	24.77/0.7841	22.77/0.6780
NLM	32.22/0.9518	29.03/0.8983	26.83/0.8427	25.17/0.7905	23.84/0.7431
ProbShrink	32.12/0.9329	27.95/0.8628	25.68/0.8048	24.13/0.7580	22.93/0.7202
LAHS	32.78/0.9517	28.60/0.8909	26.30/0.8329	24.66/0.7811	23.39/0.7326
NLAHS	33.28/0.9587	29.35/0.9136	27.21/0.8740	25.76/0.8401	24.71/0.8105
MLAHS(ours)	32.71/0.9542	28.82/0.9011	26.78/0.8497	25.37/0.8019	24.19/0.7561
MNLAHS(ours)	**33.34/0.9596**	**29.48/0.9152**	**27.38/0.8761**	**25.91/0.8424**	**24.81/0.8128**

Table 4: Average denoising performance on the CBSD68 [15] dataset (PSNR/SSIM).

Filter	**256×256**	**512×512**
GOA [25]	1.8	6.2
NLM [3]	19.1	74
LEE [11]	0.6	2
FGP [2]	0.3	1
PBshrink [18]	0.7	2.5
LAHS [9]	1.4	5.2
MLAHS	1.5	5

Table 5: Runtime (in seconds) of the denoising methods for 256×256 and 512×512 grayscale images.

A Additional Comparisons with Popular Baseline Methods

Table 6 reports PSNR values for the proposed methods (MLAHS and MNLAHS) and three advanced algorithms, namely BM3D [4], MLP [22], and DnCNN [27]. The results are given for commonly used test images and two noise levels. It is worth noting that the proposed methods are not designed to compete with advanced learning-based methods in terms of denoising quality. Instead, they aim to provide a simpler alternative with moderate denoising performance at substantially lower computational complexity. Furthermore, as an approximate measure of computational complexity, Table 7 compares the number of operations per pixel (Op./pixel) for each algorithm using a bottom-up arithmetic-operation counting strategy. The procedure consists of (1) decomposing each algorithm into its fundamental operations (additions, multiplications, divisions, comparisons, and memory accesses); (2) counting the number of operations for each basic processing unit (e.g., per patch, per block, or per layer); (3) converting the count to a per-pixel measure by accounting for patch or block overlap, stride, and redundancy; and (4) incorporating the main algorithmic optimizations explicitly described in the original articles.

Test Image	**Method**	$\sigma = 25$	$\sigma = 50$
Cameraman	Noisy	20.57	14.91
	BM3D [4]	29.45	26.13
	MLP [22]	29.61	26.37
	DnCNN-S [27]	30.18	27.03
	MLAHS	27.71	23.76
	MNLAHS	28.83	24.61
Peppers	Noisy	20.35	14.70
	BM3D [4]	30.16	26.68
	MLP [22]	30.30	26.68
	DnCNN-S [27]	30.87	27.32
	MLAHS	27.25	23.91
	MNLAHS	29.41	25.11
Monarch	Noisy	20.22	14.69
	BM3D [4]	29.25	25.82
	MLP [22]	29.61	26.26
	DnCNN-S [27]	30.28	26.78
	MLAHS	28.25	24.28
	MNLAHS	28.75	24.51

Table 6: Comparison of the proposed methods with three popular baseline denoising algorithms in terms of PSNR (dB) for two noise levels.

Algorithm	**Op./pixel**	**Parameters and assumptions**
BM3D [4]	30k	Block size: 8×8; search window: 39×39; step: 3; maximum group size: 16; transforms: 2D DCT + 1D Haar.
MLP [22]	1,060k	Layers: 4; units per layer: 2047; patch size: 17×17; stride: 3; fully-connected architecture.
DnCNN-S [27]	664k	Depth: 20 layers; feature channels: 64; kernel size: 3×3; includes convolution, batch normalization, and ReLU.
MLAHS	575	Window size: 5×5; number of paths: 2; number of estimates: 50; median and variance computed per window.
MNLAHS	6k	Search window: 21×21; number of patches: 10; local region: 3×3; hysteresis applied per patch.

Table 7: Approximate computational complexity of each method, expressed as operations per pixel (Op./pixel). Parameters are chosen according to the recommended settings in the original articles.

Image Credits

        Standard test images.

  Images from the CBSD68 color image dataset[15].

References

[1] J. F. P. Amerom, *Differential Hysteresis Process*, tech. report, University of Waterloo, Department of Systems Design Engineering, Waterloo, ON, 2002.

[2] A. Beck and M. Teboulle, *Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems*, IEEE Transactions on Image Processing, 18 (2009), pp. 2419–2434, https://doi.org/10.1109/TIP.2009.2028250.

[3] A. Buades, B. Coll, and J. M. Morel, *A Non-Local Algorithm for Image Denoising*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2005, pp. 60–65, https://doi.org/10.1109/CVPR.2005.38.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering*, IEEE Transactions on Image Processing, 16 (2007), pp. 2080–2095, https://doi.org/10.1109/TIP.2007.901238.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *BM3D Image Denoising with Shape-Adaptive Principal Component Analysis*, in Signal Processing with Adaptive Sparse Structured Representations (SPARS), 2009.

[6] L. Fan, F. Zhang, H. Fan, and C. Zhang, *Brief Review of Image Denoising Techniques*, Visual Computing for Industry, Biomedicine, and Art, 2 (2019), pp. 1–12, https://doi.org/10.1186/s42492-019-0016-7.

[7] H. Golshan and R. Hasanzadeh, *Fuzzy Hysteresis Smoothing: A New Approach for Image Denoising*, IEEE Transactions on Fuzzy Systems, 29 (2021), pp. 686–697, https://doi.org/10.1109/TFUZZ.2019.2961336.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2007. ISBN 9780132133647.

[9] R. P. R. Hasanzadeh and M. B. Daneshvar, *A Novel Image Noise Reduction Technique Based on Hysteresis Processing*, Optik, 126 (2015), pp. 3039–3046, https://doi.org/10.1016/j.ijleo.2015.07.066.

[10] A. E. Ilesanmi and T. O. Ilesanmi, *Methods for Image Denoising Using Convolutional Neural Network: a Review*, Complex & Intelligent Systems, 7 (2021), pp. 2179–2198, https://doi.org/10.1007/s40747-021-00428-4.

[11] J. S. Lee, *Digital Image Enhancement and Noise Filtering by Local Statistics*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (1980), pp. 165–168, https://doi.org/10.1109/TPAMI.1980.4766994.

[12] M. LINDENBAUM, M. FISCHER, AND A. BRUCKSTEIN, *On Gabor's Contribution to Image Enhancement*, Pattern Recognition, 1 (1994), pp. 1–8. https://doi.org/10.1016/0031-3203(94)90013-2.

[13] M. MAGGIONI, V. KATKOVNIK, K. EGIAZARIAN, AND A. FOI, *Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction*, IEEE Transactions on Image Processing, 22 (2013), pp. 119–133, https://doi.org/10.1109/TIP.2012.2210725.

[14] S. MALLAT, *A Wavelet Tour of Signal Processing*, New York: Academic, 2nd ed., 1999. ISBN 9780080520834.

[15] D. MARTIN, C. FOWLKES, D. TAL, AND J. MALIK, *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, in IEEE International Conference on Computer Vision (ICCV), vol. 2, 2001, pp. 416–423, https://doi.org/10.1109/ICCV.2001.937655.

[16] M. MAZHARI AND R. P. R. HASANZADEH, *Suppression of Noise in SEM Images Using Weighted Local Hysteresis Smoothing Filter*, Scanning, 38 (2016), pp. 634–643, https://doi.org/10.1002/sca.21311.

[17] E. OHO, N. ICHISE, W. H. MARTIN, AND K. PETERS, *Practical Method for Noise Removal in Scanning Electron Microscopy*, Scanning, 18 (1996), pp. 50–54, https://doi.org/10.1002/sca.1996.4950180108.

[18] A. PIZURICA AND W. PHILIPS, *Estimating the Probability of the Presence of a Signal of Interest in Multiresolution Single- and Multiband Image Denoising*, IEEE Transactions on Image Processing, 15 (2006), pp. 654–665, https://doi.org/10.1109/TIP.2005.863698.

[19] J. PORTILLA, V. STRELA, M. J. WAINWRIGHT, AND E. P. SIMONCELLI, *Adaptive Wiener Denoising Using a Gaussian Scale Mixture Model in the Wavelet Domain*, in IEEE International Conference on Image Processing (ICIP), vol. 2, 2001, pp. 37–40, https://doi.org/10.1109/ICIP.2001.958418.

[20] M. RAJABI AND R. P. R. HASANZADEH, *A Modified Adaptive Hysteresis Smoothing Approach for Image Denoising Based on Spatial Domain Redundancy*, Sensing and Imaging, 22 (2021), https://doi.org/10.1007/s11220-021-00364-0.

[21] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268, https://doi.org/10.1016/0167-2789(92)90242-F.

[22] U. SCHMIDT, J. JANCSARY, S. NOWOZIN, S. ROTH, AND C. ROTHER, *Cascades of Regression Tree Fields for Image Restoration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 38 (2016), pp. 677–689, https://doi.org/10.1109/TPAMI.2015.2441053.

[23] G. W. TAYLOR, *Techniques for Decreasing the Computational Time of Differential Hysteresis Processing*, in Computer Science, 2002, https://doi.org/10.1016/0167-2789(92)90242-F.

[24] C. TIAN, L. FEI, W. ZHENG, Y. XU, W. ZUO, AND C. LIN, *Deep Learning on Image Denoising: An Overview*, Neural Networks, 131 (2020), pp. 251–257, https://doi.org/10.1016/j.neunet.2020.07.025.

[25] D. V. D. Ville, M. Nachtegael, D. V. der Weken, E. E. Kerre, and W. Lemahieu, *Noise Reduction by Fuzzy Image Filtering*, IEEE Transactions on Fuzzy Systems, 11 (2003), pp. 429–436. https://doi.org/10.1109/TFUZZ.2003.814830.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image Quality Assessment: from Error Visibility to Structural Similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612, https://doi.org/10.1109/TIP.2003.819861.

[27] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, *Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising*, IEEE Transactions on Image Processing, 26 (2017), pp. 3142–3155, https://doi.org/10.1109/TIP.2017.2662206.