Image Denoising Algorithm Based on Local Adaptive Nonlinear Response Diffusion

Yuanxiu Xing¹*, Jiwu Yu², Fan Zhang¹ and Yicheng Gong¹
¹College of Science, Wuhan University of Science and Technology, Wuhan, China
²Wuhan Vocational College of Software and Engineering, Wuhan, China

*Corresponding author

Abstract. We describe an image denoising algorithm to improve the denoising performance and the time efficiency, while keeping the edge texture of the image from being blurred. First, the algorithm discriminates the pixel whether is an edge pixel or a noise pixel according to the similarity region values of the neighborhoods, and calculates the adaptive diffusion coefficients to construct an adaptive influence function, which can control the noise and the edges to have different diffusion speeds. Then, a 5-layer feedback convolution network with 16 small filters in each layer generates the reconstructed image through convolution, diffusion and deconvolution. Last, the algorithm uses multiple noise levels training sets to train and optimize the network parameters, and generates the final image denoising model. The experimental results show that the proposed algorithm using adaptive diffusion coefficients can maintain the edges and the smaller filter can improve the time efficiency. The algorithm also has higher denoising accuracy for images with unknown noise level by training on multiple-noise-level training sets.

Keywords: Image denoising; Nonlinear reaction diffusion; Local adaptive diffusion coefficient; Feedback convolution network.

1. Introduction

The performance of image recognition and classification is restricted because of the drawback of the noise pollution[1], and it is necessary to use image denoising technology to improve the image quality. In recent years, a number of image-denoising algorithms have been developed. They are divided into two categories: traditional denoising approaches and neural-network-based denoising approaches. The traditional denoising approaches are mainly based on unsupervised learning method. They denoise a single image directly according to the correlation of the local neighborhood and can achieve high denoising accuracy. However, uptime of this approach is relatively high to meet the requirements of real-time applications. The traditional image denoising methods mainly include non-local block similarity method[2], interpolation method[3], image deconvolution method[4], etc. To improve the denoising performance, BM3D algorithm[5] which is hailed one of the state-of-art denoising method and improved algorithm[6] applied wavelet transforms to non-local similar blocks and achieved good denoising performance. The advantage of traditional denoising approaches is that the denoising performance is well to additive Gaussian noise. However, the procedure of finding non-local similar blocks needs a long time and they are also not conducive to parallel computation on GPU. The neural-network-based approaches can be thought of as universal function approximator and has strong representation ability. They also can be run in parallel on GPU and have high time efficiency. As a result, a large number of neural-network-based denoising methods have been developed in recent years. Burger and Schuler used a plain multi-layer perceptron (MLP)[7] to map from a noisy image to a noise-free image. It could reduce such as Gaussian noise, salt and pepper noise, etc., but the networks used
full connection which contained too many parameters, and the image denoising efficiency was not high. Then the Markov random field (MRF) \cite{8} was proposed to achieve good image denoising performance by considering the dependence between pixels and surrounding and establishing the undirected probability map network model. The main disadvantage was that the calculation process was complex because it involved probability graph. Jain and Suen \cite{10} proposed to use convolutional neural network (CNN) to denoise the image and achieved better denoising performance. Furthermore, Kaiming he and Xiangyu Zhang \cite{10} proposed to use the depth residual learning network (ResNET) to obtain a lower training error while increasing the network depth, and effectively solved the network degradation problem. The image denoising algorithm based on non-linear reaction diffusion makes some achievements and has many interesting applications. The network is simple and can obtain good denoising performance. The trainable nonlinear reaction diffusion (TNRD) algorithm \cite{11} is a representative to this type of algorithm. The algorithm used the reaction term to meet various image restoration. Lijun Zhao and Jie Liang \cite{12} proposed a locally activity-tuned image filtering (LAD-AD), which adjusted the diffusion coefficient of the diffusion function. The edge and texture details were reserved well. In this paper, a local adaptive nonlinear reaction-diffusion algorithm is proposed to improve the denoising performance and the time efficiency while keeping the edge texture of the image from being blurred. It uses the local adaptive diffusion coefficient to adjust the diffusion rate, and the noise multi-noise-level images are used as the training set, and the small size filters are used as the convolution kernel. The experimental results show that the proposed algorithm can achieve good denoising performance and high time efficiency.

2. Local Adaptive Nonlinear Reaction Diffusion Model

2.1. Nonlinear Reaction Diffusion Model

The nonlinear reaction diffusion model \cite{12} is used to model the change of pixel values during the process of denoising iteration from the point of view of motion. It can retain important edge texture and other information by using of regularization. The partial differential equation is given by:

\[
\begin{align*}
\hat{u}_t &= \text{div}(\nabla u) - \nabla \cdot (2\gamma \nabla u) \\
\hat{u}_{t=0} &= f
\end{align*}
\]

where \( \nabla \) is the gradient of the image and \( \nabla u = \sqrt{u_x^2 + u_y^2} \) is the gradient magnitude. \( t \) is the iteration number, and \( f \) is the initial image. The divergence \( \text{div}(z) = \nabla \cdot (z) \) and \( \phi \) is known as edge-stopping function. A typical function of \( \phi \) is given by:

\[
\phi(\nabla u) = \exp(-\frac{\|\nabla u\|^2}{\gamma})
\]

where \( \gamma \) is a parameter to control the strength of \( \phi \).

A reaction term is used to meet different image restoration problems \cite{11}. It is the gradient of a data term and given by in this paper:

\[
\psi'(u, f) = \nabla u D'(u, f) = \lambda'(u - f) = \frac{\lambda^i}{2} \|

D'(u, f)
\]

where \( \lambda^i \) is related to the strength of the reaction term.

The denoised image can be achieved through several iteration, and the nonlinear reaction diffusion model is formulated as:

\[
u_t = \nu_{t-1} - \left[ \sum_{i=1}^{N_i} K_{i} \psi'(K_{i} \nu_{t-1}) + \psi'(u_{t-1}, f) \right]
\]
where \( u_t \in \mathbb{R}^N \) is the denoised image after \( t \) times iteration. \( K_i \in \mathbb{R}^{N \times N} \) and \( K_i u \) is the 2D convolutional by using the filter kernel \( k_i \), i.e., \( K_i u \Leftrightarrow k_i * u \). \( K_i^T \) is the corresponding reversed convolution kernel. \( N_k \) is the number of the filter kernel and \( \phi(z) \) is the function which defined as:

\[
\phi_k(z) = \sum_{m=1}^{M} \phi_m(z - \mu_m)
\]

where \( z \) is the convoluted feature image. \( M \) is the number of the standard radial basis function and the value of \( M \) is 51. \( \mu_m \) takes the uniform value in \([-250 : 10 : 250]\).

**2.2. Local Adaptive Diffusion Coefficients**

The larger the diffusion coefficient is, the greater the diffusion degree is. A small value of \( \gamma \) is used to keep the edge texture from being less affected by the neighborhood. On the contrary, a large value of \( \gamma \) should be used to smooth the noise pixel using its surrounding neighborhoods. In this paper, every pixel is determined whether it is a noise pixel or a pixel on an edge, and the different adaptive diffusion coefficient of each pixel is calculated.

Let \( x \) be the processed pixel and \( N_x \) represent the neighborhood composed of \( N \times N \) pixels. Figure 1 shows four cases of \( 3 \times 3 \) neighborhood where the black represents the pixels similar to \( x \) and the white represents the pixels not similar. It can be seen that if \( x \) is on the edge, there will be at least three pixels similar to it. Therefore, the number of similar pixels in the neighborhood can effectively distinguish the edge pixel from the noise pixel, and the different diffusion coefficients can be used to make the edge pixels diffuse more slowly, while the noise pixels accelerate the diffusion speed. The processed pixel in Figure 1(a) and (b) will be determined to be on the edge, and the ones in Figure 1(c) and (d) will be determined as noisy pixel.

**Figure 1.** Four cases of \( 3 \times 3 \) neighbourhood.

The degree of similarity between pixels \( r \) and \( r_0 \) is defined as:

\[
c(r, r_0) = \exp\left(-\left(\frac{f(r) - f(r_0)}{\tau}\right)^{\beta}\right)
\]

where \( f(r) \) is the grey value of \( r \). \( \tau \) is a threshold parameter and the value is 23.

We use integral-image to accelerate the calculation speed of \( F \). The similar region of \( x \) is defined as:

\[
n_s = \sum_{k \in N_x} c(u_k, u_k) = \sum_{k \in N_x} \exp\left(-\left(\frac{u_k - u_k}{\tau}\right)^{\beta}\right)
\]

The local adaptive threshold \( \gamma \) in (2) is given by:

\[
\gamma = \frac{\beta}{n_s}
\]

where \( \beta \) is adjust parameter, and in this paper, it is set as 25. If the processed pixel \( x \) is located on the edge, \( n_s \) will be larger and the value of \( \gamma \) will be relatively larger, and edge diffusion speed will slow down. Otherwise, the reverse.

**2.3. Framework**

We use the diffusion network architecture with a feedback step like [11]. We train the network with \( T = 5 \) stages, and the loss function is defined as:
where \( L(\Theta) \) is the loss which represents the difference between output denoised image \( u^t \) and ground-truth image \( u^t_{gt} \). \( \Theta_t = \{ \lambda^t, \phi^t, k^t \} \) represents network model parameters in stage \( t \). \( S \) is the number of the train samples and \( f^T \) is the noisy image and \( u^t_{gt} \) is the relatively ground-truth image.

During every train stage, the input image is convolved to generate feature images followed by using influence function. And the deconvolution processing is performed to recover the image. The network parameters should be adjusted according to minimize the loss before entering the next iteration stage.

### 3. Experiments and Analysis

#### 3.1. Data Selection

In this section, we use the open dataset FoETra iningSets180 which include 400 samples of size \( 180 \times 180 \) to train and test the proposed algorithm. We select 390 samples and add noise with noise level \( \sigma = 5, 10, 45 \) to generate 3900 train samples with different noise level. Similar to the train set, the remaining 10 samples form a test set with 100 noisy images. The algorithm is evaluated with the single-threaded implementation on Intel (R) Xeon e5-2680 @ 2.80GHz CPU with memory of 6GB. The proposed algorithm is compared with TNRD [11] and TLAD-AD [12] algorithm which are the state-of-art denoising algorithms to evaluate the denoising performance of PSNR and run time.

#### 3.2. Experiments with Standard Image Set

We trained our model on 3900 image pairs, and the test performances are shown in Figure 2 and Table 1-3. Figure 2 and Table 1 show the results testing on the man image with noisy level \( \sigma = 25 \). It can be seen that TLAD-AD uses the least run time, however the PSNR and visual effect are a little worse because there is still some noise remain in the denoised image. TNRD and our algorithm have better visual effects, and the noise are removed efficiently and the image edge and texture details are reserved. Meanwhile, our algorithm uses less time than TNRD.

![Figure 2. The comparison of different noise level \( \sigma = 25 \).](image)

| noise level | PSNR/dB | Run time/s |
|-------------|---------|-----------|
| \( \sigma = 25 \) | TNRD | TLAD-AD | The proposed | TNRD | TLAD-AD | The proposed |
| 29.876 | 24.046 | 30.168 | 1.9233 | 0.4651 | 1.4031 |

We also use test set to test the performance on different noise levels. It can be seen from Table 2 that the three algorithms can achieve better denoising effect when the noise level is low. However, both the PSNR value and the visual effects of the denoised image drop with the increase of noise level. Our algorithm achieves the highest average PSNR because it trains the model on various noise level and uses...
adaptive diffusion coefficients. It also can be seen from Table 3 that TLAD-AD uses the least run time and the proposed algorithm uses less run time than TNRD because our model has less network parameters than TNRD.

| Table 2. Average PSNR (dB) on 100 test images with different noise level |
|-------------------------------------------------|
| PSNR/dB | noise level |
|        | σ=5 | σ=15 | σ=25 | σ=35 | σ=45 |
| TNRD    | 30.864 | 30.095 | 29.643 | 23.333 | 19.579 |
| TLAD-AD | 30.731 | 28.249 | 25.047 | 21.578 | 17.731 |
| The proposed | 30.871 | 30.703 | 30.217 | 25.417 | 21.654 |

| Table 3. Average run time on 100 test images with different noise level |
|-------------------------------------------------|
| PSNR/dB | Run time/s |
|        | σ=5 | σ=15 | σ=25 | σ=35 | σ=45 |
| TNRD    | 1.877931 | 1.989672 | 1.911630 | 1.892235 | 2.134496 |
| TLAD-AD | 0.556812 | 0.544521 | 0.444501 | 0.489234 | 0.523590 |
| The proposed | 1.456692 | 1.349721 | 1.589052 | 1.457922 | 1.622194 |

3.3. Experiments with Mineshaft Image
Three denoising methods are compared on mineshaft image and the result are shown in Figure 3 and Table 4.

![Mineshaft Image](image)

(a) mineshaft image (b) TNRD (c) TLAD-AD (d) The proposed

We can’t estimate the PSNR because it difficult to sample clean images in mineshaft, therefore we use the visual effect to judge the performance of comparison algorithms. Similar to man image, most of the noise is remove from the mineshaft images denoised by TNRD and the proposed algorithm and the visual effect is better than the denoised by TLAD-AD. In time efficiency aspect, TLAD-AD use the least run time to denoise the image, and time efficiency of the proposed algorithm is less than TNRD.

| Table 4. Run time (in seconds) on mineshaft image |
|-------------------------------------------------|
| Methods | TNRD | TLAD-AD | LA-TNRD |
| Run time (s) | 2.269843 | 0.4429 | 1.257165 |

4. Conclusion
In this paper, an image denoising algorithm was proposed based on local adaptive nonlinear response diffusion. The adaptive diffusion coefficient which could control the diffusion speed was estimated by
calculating similar region value and judging the processed pixel as noise or edge pixel. This strategy could further keep the edge texture feature while removing noise. Meanwhile, convolution network architecture with small size convolution filter was used and the denoising model was trained to improve the time efficiency. Furthermore, multi-noise level images in the training set were used to meet the real image denoising applications with unknown noise level. Experimental results show that our algorithm can improve the time efficiency and also has higher denoising accuracy for images with unknown noise level compared with other algorithms. It can be promoted in mineshaft denoising applications.

Acknowledgments
This work was supported by Hubei innovation and entrepreneurship training program for college students (201810488097), the National Natural Science Foundation of Hubei province (no. 2015CFB602).

Reference
[1] Dodge S, Karam L. Understanding How Image Quality Affects Deep Neural Networks[C]. IEEE. International Conference on Quality of Multimedia Experience, Portugal, 2016. Quality of Multimedia Experience. 2016:1-6
[2] Julien M, Francis R B, Jean P. Non-local sparse models for image restoration[C]. IEEE. International conference on computer vision, Kyoto, 2009. IEEE Translation on Image Processing. 2009: 2272-2279
[3] Yaniv R, Matan P, Michael E. Single image interpolation via adaptive nonlocal sparsity-based modeling[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 2014, 23(7): 849-862
[4] Weisheng D, Lei Z, Guangming S, Xin L. Nonlocally centralized sparse representation for image restoration[J]. IEEE Transactions on Image Processing, 2013, 22(4): 1620-1630
[5] Kostadin D, Alessandro F, Vladimir K. Image denoising by sparse 3-d transform-domain collaborative filtering[J]. IEEE Transactions on Image Processing 2007, 16(8): 2080-2095
[6] Yang J, Zhang X, Yue H. IBM3D: Integer BM3D for Efficient Image Denoising[J]. Circuits Systems and Signal Processing, 2019, 38(2): 750-763
[7] Harold C, Burger C. Image denoising: Can plain neural networks compete with BM3D?[C]. IEEE. Computer Vision and Pattern Recognition, Providence, 2012. IEEE Conference on Computer Vision and Pattern Recognition. 2012: 2392-2399
[8] Parise S, Welling M. Learning in markov random fields: An empirical study[J]. Joint Statistical Meeting, 2005
[9] Viren J, H S. Natural image denoising with convolutional networks[C]. Neural information processing systems, British Columbia, 2008. NIPS 2008. 2008: 769-776
[10] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. Computer vision and pattern recognition, 2016: 770-778
[11] Yunjin C, Thomas P. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1256-1272
[12] Lijun Z. Local Activity-tuned Image Filtering for Noise Removal and Image Smoothing. Signal Processing, 2018,157:62-72