Non-Local Means Algorithm for Image De-noising

DIXIT AARYA
Department of Electronics and Telecommunication
Maharashtra Institute of Technology
Pune - 411038
INDIA
aarya_dixit@hotmail.com

Abstract: - Images are often corrupted with noise during acquisition, transmission and retrieval from storage media. So the need for efficient image de-noising methods has grown with the massive and easy production of digital images and movies. Furthermore, de-noising is often necessary as a pre-processing step in image compression, segmentation recognition etc. Therefore, de-noising has been an important and widely studied problem in image processing and computer vision. Basically, the image de-noising methods are divided into two types: local and non-local. The methods that only exploit the spatial redundancy in local neighborhoods are referred as Local methods. The methods that estimate pixel intensity based on information from the whole image and thereby exploiting the presence of similar patterns and features in an image are referred as Non-Local. A non local method called as Non-Local Means[3] estimates a noise-free pixel intensity as a weighted average of all pixel intensities in the image, and the weights are proportional to the similarity between the local neighbourhood of the pixel being processed and local neighbourhoods of surrounding pixels The method is quite spontaneous and powerful that results in comparable PSNR and visual quality to other non-local methods.

Key-Words: - denoising; local neighbourhood; MSE; Non-local Means; PSNR; visual quality

1 Introduction

Over the years a variety of methods have been introduced to remove noise from digital images, such as Gaussian filtering, Anisotropic filtering, Total Variation minimization etc. However, many of these algorithms remove the fine details and structure of the image in addition to the noise because of assumptions made about the frequency content of the image. The Non-Local Means algorithm does not make these assumptions, but instead assumes that the image contains an extensive amount of redundancy. These redundancies can then be exploited to remove the noise in the image. Previous methods attempt to separate the image into the smooth part (true image) and the oscillatory part (noise) by removing the higher frequencies from the lower frequencies. However, not all images are smooth. Images can contain fine details and structures which have high frequencies. When the high frequencies are removed, the high frequency content of the true image will be removed along with the high frequency noise because the methods cannot tell the difference between the noise and true image. This will result in a loss of fine detail in the de-noised image. Also, the issue to remove the low frequency noise from the image was a concern. Low frequency noise will remain in the image even after de-noising. Due to the loss of image details, Baudes A., Coll B., Morel J.M. have developed the Non-Local Means (NLMeans) algorithm [3]. In this paper, NL-Means algorithm for standard database images and natural photograph images captured by general digital camera is implemented. The paper is organised as follows: Section 1 gives brief introduction, section 2 deals with the implementation of NL-Means algorithm, its Pseudo code, the noise model to generate uncorrelated (Adaptive White Gaussian Noise i.e. AWGN) and correlated (coloured) noise and the post process used, section 3 with experimentation and results while conclusions are given in section 4.

2 Basic NL-Means Algorithm[3], Pseudo code, Noise Model And Post Processing Filter [1]

2.1 Basic NL-Means algorithm
The self-similarity assumption can be exploited to de-noise an image. Pixels with similar neighborhoods can be used to determine the de-noised value of a pixel. Weights are assigned to
Mathematically, it can be expressed as:

\[ \text{NL} [u](x) = \frac{1}{C(x)} \int e^{- \frac{(G_a * [u(x+\cdot) - u(y+\cdot)])^2(0)}{h^2} u(y) dy} \]  

(1)

The integration is carried out over all the pixels in the search window. Where

\[ C(x) = \int e^{- \frac{(G_a * [u(x+\cdot) - u(y+\cdot)])^2(0)}{h^2} dz} \]  

(2)

\( C(x) \) is a normalizing constant. \( G_a \) is a Gaussian kernel and \( h \) is a filtering parameter [3].

2.2 Pseudo code for NL-Means algorithm -

For each pixel \( x \)

Step 1. Take a window centered in \( x \) and size \((2m+1 \times 2m+1), A(x,m)\)

Step 2. Take a window centered in \( x \) and size \((2n+1 \times 2n+1), W(x,n)\).

\( w_{\text{max}} = 0; \)

Step 3. For each pixel \( y \) in \( A(x,m) \) and \( y \) different from \( x \),

Compute the difference between \( W(x,n) \) and \( W(y,n) \) as \( d(x,y) \).

Compute the weight from the distance \( d(x,y) = \exp(-d(x,y)/h) \); If \( w(x,y) \) is bigger than \( w_{\text{max}} \) then

\( w_{\text{max}} = w(x,y); \)

Compute the average,

\( \text{average} + = w(x,y) \times u(y); \)

Carry the sum of the weights,

\( \text{totalweight} + = w(x,y); \)

Step 4. Give to \( x \) the maximum of the other weights,

\( \text{average} + = w_{\text{max}} \times u(x); \)

\( \text{totalweight} + = w_{\text{max}}; \)

Compute the restored value,

\( \text{rest}(x) = \frac{\text{average}}{\text{totalweight}}; \)

Step 5. The distance is calculated as follows:

\[ \text{distance} = (u(x) - u(y))^2; \]

for \( k = 1 \) until \( n \) {

for each \( i=(1, 2) \) pair of integer numbers such that

\( \text{max}(i[1], i[2]) = k \) {

\( \text{distance} += (u(x + i) - u(y + i))^2; \)

}\}

\( \text{aux} = \text{distance} / (2*k + 1)^2; \)

2.3 The Noise Model

This module generates Power Spectral Density (PSD) based on [6] for different types of noise to add to the Original Clean Image and get a Noisy Image for testing the performance of the algorithm. Figure 1(a-b) shows 3D Mesh plot for two of the following noise types:

i. Adaptive White Gaussian Noise (uncorrelated)

ii. Circular symmetric noise (correlated)

iii. Single line pattern noise (correlated)

iv. Double line pattern noise (correlated)

![Fig. 1(a).3D mesh plot- Circular Symmetric Noise](image)

![Fig 1(b). 3D mesh plot- Double Line Pattern Noise](image)

Figure 2 shows the images corrupted by the noise generated by the noise models. The “Well” image is a natural image captured by a general digital camera, the “Barbara” and “Couple” are the standard database images.
process and later a local filter is applied in those particular areas to remove the remaining noise and improve the PSNR. This filter is based on Karhunen-Loève Transform (KLT) that shows the best de-correlating capability with the uncorrelated coefficients and zero mean.

3 Experimentation and Results
The proposed algorithm is implemented in MATLAB R2009a and is tested on PC, Core i5 and 4Gb RAM. Original Test images are added with different types of noise generated by the noise model and processed with the proposed de-noising algorithm.

The functional parameters used are:

i. Noise variance, Sigma = 25
ii. Search Window Size = 9 x 9
iii. Bandwidth Parameter, h = 2.1\sigma
iv. KLT block size = 16

Figure 3(a-d) shows the images corrupted by different types of noises with PSNR value 20.18 dB. NL-Means estimated and post filtered de-noised versions of the corrupted images with improved PSNR are also displayed.

Table 1. summarizes the PSNR (in dB) computed w.r.t. the Original Image for different images with different noise types.

<table>
<thead>
<tr>
<th>TABLE 1. Comparison of PSNR (in dB) results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Type</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>AWGN</td>
</tr>
<tr>
<td>Circular symmetric pattern</td>
</tr>
<tr>
<td>Double line pattern</td>
</tr>
</tbody>
</table>

Table 2. summarizes the Mean Square Error (MSE) results of the de-noising experiment with AWGN of varied $\sigma$ values.
TABLE 2. Comparison of MSE results for different \( \sigma \) values of AWGN

<table>
<thead>
<tr>
<th>Image</th>
<th>For the proposed algorithm ( \sigma =20 )</th>
<th>For the proposed algorithm ( \sigma =25 )</th>
<th>For the proposed algorithm ( \sigma =50 )</th>
<th>For the Original NL-Means [3] ( \sigma =20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baboon</td>
<td>97</td>
<td>136</td>
<td>315</td>
<td>292</td>
</tr>
<tr>
<td>Lena</td>
<td>37</td>
<td>47</td>
<td>103</td>
<td>68</td>
</tr>
</tbody>
</table>

Fig 3(a). AWGN noise in “Lena” image

Fig. 3(b). Circular Symmetric noise in “Well” image

Fig. 3(c). Single Line Pattern noise in “Barbara” image

Fig 3(d). Double Line Pattern noise in “Couple” image

Table 1. summarizes the PSNR (in dB) computed w.r.t. the Original Image for different images with different noise types.

4 Conclusion

The de-noising results of the proposed algorithm are comparable to that of the original NL-Means [3] in terms of MSE. Improvement in MSE is achieved with reference to the original NL-Means [3]. Smaller MSE indicates that the estimate is closer to the Original Image as shown in Table 2 for \( \sigma =20 \). The proposed algorithm performs the best on periodic textures like the stripped scarf, trousers in the Barbara image. The Natural photograph like “Well” also have enough redundancy to exploit by NL-Means, that’s why the results are comparable with the Standard Test images. With the proposed algorithm, the de-noising is achieved with smoother reconstruction and less artifacts. It proves that the algorithm is spontaneous and powerful. The proposed de-noising algorithm accomplished its goal of de-noising, i.e. improving PSNR and preserving the details, especially the edges.

References: