Impact & Analysis of Improved Bilateral Filter on TEM Images

Garima Goyal
Assistant Professor, Department of Information Science, Jyothy Institute of Technology, Bangalore, India

Abstract: TEM images are rapidly gaining prominence in various sectors like life sciences, pathology, medical science, semiconductors, forensics, etc. Hence, there is a critical need to know the effect of existing image restoration and enhancement techniques available for TEM images. This paper primarily focuses on denoising Bilateral Filter. The simulation is carried on greyscale and colored images separately. To do so different types of noise (Gaussian Noise, Salt & Pepper Noise, Salt & Pepper Noise & Poisson Noise) varying from 1% to 9% is incorporated into image. Each degraded image is denoised by filters.

Keywords: TEM Image, Bilateral Filter, denoising, SNR, PSNR

1. Literature Survey

In 1984, a method for removing impulse noises from images was proposed whereby the filtering scheme is based on replacing the central pixel value by the generalized mean value of all pixels inside a sliding window. The concepts of thresholding and complementation which are shown to improve the performance of the generalized mean filter are introduced. The threshold is derived using a statistical theory. The actual performance of the proposed filter is compared with that of a file commonly used median filter by filtering noise corrupted real images. The hardware complexity of the two types of filters is compared indicating the advantages of the generalized mean filter [4].

By 1988, two algorithms using adaptive-length median filters are proposed for improving impulse-noise-removal performance for image processing. The algorithms achieved significantly better image quality than regular (fixed-length) median filters when the images are corrupted by impulse noise. One of the algorithms, when realized in hardware, requires rather simple additional circuitry. Both algorithms can easily be integrated into efficient hardware realizations for median filters [5].

Tomasi and Manduchi proposed the bilateral filter in 1998 [94] as an appealing algorithm for noise removal from images. As such, this algorithm was posed as an alternative to locally adaptive well-known algorithms such as the anisotropic diffusion (AD), the weighted least-squares (WLS), and the robust estimation (RE) techniques. However, no theoretical background supporting the bilateral filter was suggested. Bayesian approach is also in the core of the bilateral filter parameters has an important influence on its behavior and performance. They have to be chosen considering the end application. In the case of noise removal, the parameters have to be adapted to the noise level, while the bilateral filter adapts itself to the image details content.

Bilateral filter [11] is firstly presented by Tomasi and Manduchi in 1998. The concept of the bilateral filter was also presented in [12] as the SUSAN filter and in [13] as the neighborhood filter. It is mentionable that the Beltrami flow algorithm is considered as the theoretical origin of the bilateral filter [14] [15] [16], which produces a spectrum of linear Gaussian filtering to the 1 L non-linear flows. The bilateral filter takes a weighted sum of the pixels in a local neighborhood; the weights depend on both the spatial distance and the intensity distance. In this way, edges are preserved well while noise is averaged out. Mathematically, at a pixel location x, the output of a bilateral filter is calculated as follows.

\[ I'(x,y) = \frac{1}{C} \sum_{y \in N(x)} e^{-|y-x||^2/(2\sigma_g^2)} e^{-|r(y)-r(x)||^2/(2\sigma_r^2)} I(x,y) \] (2.1)

where \( \sigma_g \) and \( \sigma_r \) are parameters controlling the fall-off of weights in spatial and intensity domains, respectively, \( N(x) \) is a spatial neighborhood of pixel \( I(x) \), and \( C \) is the TEMization constant: It reduces noise in images while preserving edges by means of nonlinear combination of local pixel values. Its formulation and implementation are both simple. However, the BF is not parameter-free. The set of the bilateral filter parameters has an important influence on its behavior and performance. They have to be chosen considering the end application. In the case of noise removal, the parameters have to be adapted to the noise level, while the bilateral filter adapts itself to the image details content.

Its formulation is simple: each pixel is replaced by a weighted average of its neighbors. This aspect is important because it makes it easy to acquire intuition about its behavior, to adapt it to application-specific requirements, and to implement it. It depends only on two parameters that indicate the size and contrast of the features to preserve. It can be used in a non-iterative manner. This makes the parameters easy to set since their effect is not cumulative over several iterations.

The bilateral filter is also defined as a weighted average of nearby pixels, in a manner very similar to Gaussian
convolution. The difference is that the bilateral filter takes into account the difference in value with the neighbors to preserve edges while smoothing. The key idea of the bilateral filter is that for a pixel to influence another pixel, it should not only occupy a nearby location but also have a similar value.

The bilateral filter smooths an input image while preserving its edges. Each pixel is replaced by a weighted average of its neighbors. Each neighbor is weighted by a spatial component that penalizes distant pixels and range component that penalizes pixels with a different intensity. The combination of both components ensures that only nearby similar pixels contribute to the final result. The weights shown apply to the central pixel.

An important characteristic of bilateral filtering is that the weights are multiplied: if either of the weights is close to zero, no smoothing occurs. As an example, a large spatial Gaussian coupled with narrow range Gaussian achieves limited smoothing despite the large spatial extent. The range weight enforces a strict preservation of the contours.

The bilateral filter can split an image into two parts: the filtered image and its “residual” image. The filtered image holds only the large-scale features, as the bilateral filter smoothed away local variations without affecting strong edges. The residual image, made by subtracting the filtered image from the original, holds only the image portions that the filter removed. Depending on the settings and the application, this removed small-scale component can be interpreted as noise or texture. Applications such as tone management and style transfer extend this decomposition to multiple layers.

To conclude, bilateral filtering is an effective way to smooth an image while preserving its discontinuities scales. As we will see, the bilateral filter has many applications, and its central notion of assigning weights that depend on both space and intensity can be tailored to fit a diverse set of applications. Unlike Gaussian convolution that smooths the images without respecting their visual structures, the bilateral filter preserves the object contours and produces sharp results.

### 3. Algorithm

B = bfilter2 (A, W, SIGMA) performs 2-D bilateral filtering for the greyscale or color image A. A should be a double precision matrix of size NxMx1 or NxMx3 (i.e., greyscale or color images, respectively) with normalized values in the closed interval [0, 1]. The half-size of the Gaussian bilateral filter window is defined by W. The standard deviations of the bilateral filter are given by SIGMA, where the spatial-domain standard deviation is given by SIGMA (1) and the intensity-domain standard deviation is given by SIGMA (2).

bfilter2 (A, w, sigma, B)
1. Read A, w, sigma.
2. Verify A, if f exists || valid, display “Image is undefined or invalid”.
3. Verify if A in double precision matrix else display “A must be a double precision matrix”.
4. Verify w, sigma.
5. if (size(A,3)==1)
   6. B = bfltGray (A,w,sigma(1),sigma(2))
   7. else
   8. B = bfltColor (A,w,sigma(1),sigma(2))
   9. endif
10. end
11. return B

bfltGray(A,w,sigma_d,sigma_r, B)
1. Obtain [X,Y] = meshgrid(-w:w,-w:w)
2. G = exp(-(X.^2+Y.^2)/(2*sigma_d^2))
3. dim = size(A)
4. B = zeros(dim)
5. Repeat for i = 1:dim(1)
6. Repeat for j = 1:dim(2)
7. iMin = max(i-w,1)
8. iMax = min(i+w,dim(1))
9. jMin = max(j-w,1)
10. jMax = min(j+w,dim(2))
11. I = A(iMin:iMax,jMin:jMax)
12. H = exp(-(I-A(i,j)).^2/(2*sigma_r^2))
13. F = H.*G((iMin:iMax)-i+w+1,(jMin:jMax)-j+w+1)
14. B(i,j) = sum(F(:).*I(:))/sum(F(:))
15. end for
16. end for
17. return B

bfltColor(A, w, sigma_d, sigma_r, B)
1. Convert input sRGB image to CIELab color space.
2. Obtain [X,Y] = meshgrid(-w:w,-w:w)
3. G = exp(-(X.^2+Y.^2)/(2*sigma_d^2))
4. sigma_r = 100*sigma_r
5. dim = size(A)
6. B = zeros(dim)
7. Repeat for i = 1:dim(1)
8. Repeat for j = 1:dim(2)
9. iMin = max(i-w,1)
10. iMax = min(i+w,dim(1))
11. jMin = max(j-w,1)
12. jMax = min(j+w,dim(2))
13. I = A(iMin:iMax,jMin:jMax)
14. DL = I(:,:,1)-A(i,j,1)
15. da = I(:,:,2)-A(i,j,2)
16. db = I(:,:,3)-A(i,j,3)
17. H = exp(-(DL.^2+da.^2+db.^2)/(2*sigma_r^2))
18. F = H.*G((iMin:iMax)-i+w+1,(jMin:jMax)-j+w+1)
19. norm_F = sum(F(:))
20. B(i,j,1) = sum(sum(F.*I(:,:,1)))/norm_F
21. B(i,j,2) = sum(sum(F.*I(:,:,2)))/norm_F
22. B(i,j,3) = sum(sum(F.*I(:,:,3)))/norm_F
23. end for
24. end for
25. Convert filtered image back to sRGB color space.
26. Return B.

The procedure bflt2 decides whether the input image is greyscale or colored image and calls the appropriate sub-
procedure. If image is greyscale image, firstly Gaussian
domain weights are computed in line 1 & 2. Then, local
region is extracted from line 5 to line 10. The, Gaussian
intensity weights are determined in line 11 & 12. Then, the
weights are multiplied and the key characteristic is that if
either of the weights is zero, no smoothing occurs. Each
pixel is replaced by a weighted average of its neighbors.
Each neighbour is weighted by a spatial component that
penalizes distant pixels and range component that penalizes
pixels with different intensity. Thus this combination of both
spatial and range components ensure that only nearby
similar pixels contribute to the final result i.e. for a pixel to
influence another pixel, it should not only occupy a nearby
location, but also have a similar value.

4. Experimental Results

<table>
<thead>
<tr>
<th>Noise Intensity</th>
<th>Mean MSE</th>
<th>SNR</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Noise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>169.6005</td>
<td>9.75E+01</td>
<td>28.1477</td>
</tr>
<tr>
<td>0.002</td>
<td>169.983</td>
<td>9.75E+01</td>
<td>28.1241</td>
</tr>
<tr>
<td>0.003</td>
<td>170.034</td>
<td>1.04E+02</td>
<td>28.0868</td>
</tr>
<tr>
<td>0.004</td>
<td>170.1437</td>
<td>3.69E+01</td>
<td>28.0111</td>
</tr>
<tr>
<td>0.005</td>
<td>199.7344</td>
<td>1.04E+02</td>
<td>27.9393</td>
</tr>
<tr>
<td>0.006</td>
<td>170.3498</td>
<td>1.70E+02</td>
<td>27.9641</td>
</tr>
<tr>
<td>0.007</td>
<td>170.5581</td>
<td>1.04E+02</td>
<td>27.9641</td>
</tr>
<tr>
<td>0.008</td>
<td>170.6391</td>
<td>1.04E+02</td>
<td>27.871</td>
</tr>
<tr>
<td>0.009</td>
<td>170.6391</td>
<td>1.04E+02</td>
<td>27.8687</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noise Intensity</th>
<th>Mean MSE</th>
<th>SNR</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speckle Noise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>167.8911</td>
<td>1.70E+01</td>
<td>35.8157</td>
</tr>
<tr>
<td>0.002</td>
<td>167.8896</td>
<td>4.95E+01</td>
<td>35.5895</td>
</tr>
<tr>
<td>0.003</td>
<td>167.878</td>
<td>1.91E+01</td>
<td>35.3151</td>
</tr>
<tr>
<td>0.004</td>
<td>167.8507</td>
<td>2.02E+01</td>
<td>35.0828</td>
</tr>
<tr>
<td>0.005</td>
<td>167.8649</td>
<td>2.10E+01</td>
<td>34.9846</td>
</tr>
<tr>
<td>0.006</td>
<td>167.8256</td>
<td>2.22E+01</td>
<td>34.6764</td>
</tr>
<tr>
<td>0.007</td>
<td>167.8477</td>
<td>2.35E+01</td>
<td>34.4147</td>
</tr>
<tr>
<td>0.008</td>
<td>167.853</td>
<td>2.45E+01</td>
<td>34.2334</td>
</tr>
<tr>
<td>0.009</td>
<td>167.8498</td>
<td>2.56E+01</td>
<td>34.0588</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noise Intensity</th>
<th>Mean MSE</th>
<th>SNR</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salt &amp; Pepper Noise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>169.9453</td>
<td>2.04E+01</td>
<td>35.0371</td>
</tr>
<tr>
<td>0.002</td>
<td>169.9899</td>
<td>2.62E+01</td>
<td>33.9485</td>
</tr>
<tr>
<td>0.003</td>
<td>168.0367</td>
<td>3.26E+01</td>
<td>32.9921</td>
</tr>
<tr>
<td>0.004</td>
<td>168.0638</td>
<td>3.78E+01</td>
<td>32.3611</td>
</tr>
<tr>
<td>0.005</td>
<td>168.1384</td>
<td>4.60E+01</td>
<td>31.5029</td>
</tr>
<tr>
<td>0.006</td>
<td>168.1544</td>
<td>4.99E+01</td>
<td>31.1496</td>
</tr>
<tr>
<td>0.007</td>
<td>168.1956</td>
<td>5.36E+01</td>
<td>30.8411</td>
</tr>
<tr>
<td>0.008</td>
<td>168.2498</td>
<td>6.31E+01</td>
<td>30.1284</td>
</tr>
<tr>
<td>0.009</td>
<td>168.2771</td>
<td>7.51E+01</td>
<td>29.6921</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noise Intensity</th>
<th>Mean MSE</th>
<th>SNR</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson Noise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>167.8979</td>
<td>235.62</td>
<td>16.0744</td>
</tr>
</tbody>
</table>

5. Flowchart

6. Conclusion

It is evident from the results obtained that even for high
resolution nanoscopic TEM images; the bilateral filter
smoothed away local variations without affecting strong
edges. Bilateral filtering is an effective way to smooth an
image while preserving its discontinuities. Because of
smoothing, very fine details are lost, so this bilateral filter
can be modified to preserve these fine details, this can
further be improved by combing it with the features of
wavelets.

References

[1] Tolga Tasdizen, Ross Whitaker, Robert Marc, Bryan
Jones, “Automatic Correction of Non-uniform
Illumination in Transmission Electron Microscopy
Images”, Scientific Computing and Imaging Institute,
University of Utah.
“Enhancement of cell boundaries in Transmission
Electron Microscopy Images”, proceeding in
International Conference of Image Processing. 2005;
2:129-132.
[3] Pawan Patidar, Manoj Gupta, Sumit Srivastava, Ashok
Kumar Nagawat, “Image Denoising by Various Filters
for Different Noise”, International Journal of Computer

Applications (0975 – 8887), Volume 9– No.4, November 2010.


Pictorial Results

<table>
<thead>
<tr>
<th>BILATERAL FILTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Image</td>
</tr>
<tr>
<td>Greyscale Normal Image</td>
</tr>
<tr>
<td>Greyscale Colored Image</td>
</tr>
</tbody>
</table>
**Author Profile**

**Garima Goyal** is an Assistant Professor at the Department of Information Science, Jyothy Institute of Technology, Bangalore, India. Her research interest is in the area of image processing. She has published more than ten papers in reputed journals.