A Metaphorical Scrutiny in Banishing Salt and Pepper Noise Using Time Domain and Frequency Domain Filters

Gopika. S. Kumar, S. Karthikeyan and N. Ambily

Government Engineering College, Trivandrum, Kerala, Sathyabama University, Chennai, India

doi: http://dx.doi.org/10.13005/bbra/1596

(Received: 15 August 2014; accepted: 10 October 2014)

In this paper, an attempt is made to de-noise the image using isolation model and a comparison is made between the existing time domain and frequency domain filters based upon their performance analysis, and parametric estimations. An image gets corrupted by various noises during transmission through the channels. One of the noise types that include for image noising is salt and pepper noise. Salt and pepper noise is also known as Impulse noise. The main objective of filter is to remove the impulse noise without affecting the original image. The filters used for de-noising are of two domains, they are time or spatial domain and frequency domain. The time domain filters include mean, median, standard median filter and Decision Based Unsymmetrical Trimmed Median filter (DBUTM). The frequency domain includes the discrete wavelet transform (DWT) using shrinkage rule. The experiment result shows that the isolation algorithm provides better de-noising than the time domain filters and DWT.

Key words: DBUTM, DWT, frequency domain, Isolation, mean, median, and time domain.

An image gets affected by noisy pixels during the acquisition and transmission through various channels. The severity of noise in the image can be identified by the bit error values. The noise signals are classified into two main types based on the distribution of noisy pixels - (i) fixed-Valued noise (ii) random valued impulse noise [Dodda Shekar et al, 2011] [Chi-Yuan Lien et al]. The fixed-valued noise is also known as salt and pepper noise. In images that are corrupted by salt and pepper noise contains the corrupted pixel with maximum or minimum grey level [S.Esakkirajan et al, 2011]. The minimum gray level (0) is said to be as salt or white noise and the maximum gray level (255) is represented as pepper or black noise. The denoising methods involved in salt and pepper noise are classified into two types (1) Low – complex methods and (2) High-complex methods [Venkatusubramanian Adhinarayanan et al, 2012]. The filtering methods [S.Karthikeyan et al, 2014] like mean, median and DBUTM belongs to low complexity technique and the methods like DWT, isolation are included in the High-complexity techniques.

Review on time domain filters

In [Dodda Shekar et al, 2011], the DBUTM for the removal of salt and pepper noise is proposed as a modified concept of decision based algorithm. It is a modified sheer algorithm and found to be more efficient than the median filter in edge preserving. In [K.Vasanth et al, 2012], the proposed Unsymmetrical Trimmed variants algorithm is found to be effective than the conventional existing algorithm and adaptive median filter till 30% of impulse noise. In
[S. Esakkirajan et al., 2011], a modified Decision based unsymmetrical algorithm is proposed which seems to have better PSNR and IEF compared to median filter, adaptive median filter, and decision based algorithm. In [Gajanand Gupta, 2011], a mathematical model for mean, median and improved median filter is designed for the corrupted lena image consisting of various noise types and noise levels using steady fixed window length size. In [M. Prema Kumar et al., 2011], a new concept of decision based IQA algorithm is used to remove salt and pepper noise from the image and concluded that this algorithm removes only corrupted pixels by the median value or by its neighboring pixels without disturbing the original image. As a result it de-noises effectively even at 90% noise level and preserves the edges without any lose up to 80% noise level. In [Vishal Gautam et al., 2012], an improved median filtering for the removal of salt and pepper noise is presented and compared with simple mean and median filter based on the PSNR Values and found that the improved algorithm preserves the integrity of edge and detailed image information. In [Gurmeet Kaur et al., 2013], a conclusion is made that the wavelet transform performance is better than the Gaussian, average, wiener, and median filter by comparing the parameters such as PSNR, RMSE, Entropy and correlation of the image. In [Sanjay Singh et al., 2012], the method of de-noising the salt and pepper noise by applying median filter and high pass filter is made and found that the median filter removes the noise and high pass filter preserves the edges. In [Slam saleh Al-amri et al., 2010], the de-noising of three types of noises such as salt and pepper noise, speckle noise, and Random Valued Impulse noise is studied using mean filter, adaptive wiener filter, Gaussian filter, Standard median filter and adaptive median filter and concluded that adaptive median filter is better to remove Salt and pepper noise and Standard median filter is good when noise level is less than 40%. In [T.K. Thivakaran et al., 2010], adaptive median filtering method is proposed in comparison with median filter and central weighted median filter to de-noise an image and found AMF provides better performance. In [32], an attempt is made to study the performance analysis between median filter and wiener filter. He found that wiener is better in removing the salt and pepper noise than wiener.

**Frequency domain filters**

**Wavelet Transform**

Wavelet is based on the mathematical functions that involves in splitting data into various frequency components and then approach each component with a resolution matched to its scale. It provides better de-noising than the time domain filters by separating noisy signal from the image. The wavelet also provides better energy compaction than the previous filters. A wavelet transform is a representation of various wavelet functions. The de-noising is done by finding the threshold value for each part by using Shrinkage rules. The steps involved in this are:

- **Step 1:** To decompose the corrupted image apply DWT.
- **Step 2:** Calculate the threshold value.
- **Step 3:** Apply soft threshold concept to the noisy coefficients.
- **Step 4:** Apply inverse DWT.

In this each decomposed part is threshold by comparing it with the threshold value calculated Bayes Shrink. The variance is the important parameter in Bayes shrinkage calculation. It is calculated by summing the variance of the original pixel with the variance of noise applied on the image. The Bayes Shrink is calculated by using

\[ \sigma_y^2 = \sigma_x^2 + \sigma^2 \]  
\[ \sigma_y^2 \] is the variance of noisy image.  
\[ \sigma_x^2 \] is the variance of original image.  
\[ \sigma^2 \] is the noise variance.  
\[ \sigma_y^2 = \frac{\sum b_m^2}{M} \]  

Where \( b_m \) is the coefficient of wavelets and \( m=1 \).

\[ M \] is the number of sub band coefficients.

\[ \sigma_x^2 = \sqrt{\max (\sigma_y^2 - \sigma^2, 0)} \]  
\[ \sigma = \text{median} \left[ d(i,j) \right] / 0.6745R \]  
\[ \lambda = \sigma^2 / \sigma_x^2 \]  

The equation (1) provides the calculation of variance of a Bayer shrinkage and equation (2), (3) and (4) represents the formulae to calculate the variance of noisy pixel, original pixel and the variance of the noise added to the image. The Bayes Shrink is selected for its efficient in providing smoothening of reconstructed image.

**Isolation algorithm**

In general, the non–corrupted pixels in
the smooth region should be of nearer values or of slightly varying values. The corrupted pixel in a matrix of widow size 3x3 can be determined by the distribution of larger different values with the surrounding pixels.

\[
\begin{array}{ccc}
    a & b & c \\
    d & M(i,j) & e \\
    f & g & h \\
\end{array}
\]

A current pixel in a matrix can be determined by the smoothness of its surrounding pixels. In this algorithm, a noisy pixel can be considered as an isolation point. If the difference between the noisy pixel and its surrounding pixels are found to be large, then the following steps are to be followed.

Step 1: Find the maxima and minima luminance values of a matrix.
Step 2: Consider the first row as top half and the third row as bottom half in a 3x3 window.
Step 3: Find the maximum and minimum value of top half and is represented by the \( \alpha_{\text{max}} \) and \( \alpha_{\text{min}} \). The difference in luminance value is given by \( \alpha_d = \alpha_{\text{max}} - \alpha_{\text{min}} \). Similarly, the maximum and minimum value of bottom half is given by the symbol \( \beta_{\text{max}} \) and \( \beta_{\text{min}} \). The difference is given by \( \beta_d = \beta_{\text{max}} - \beta_{\text{min}} \).
Step 4: To find whether the surrounding pixels form a smooth region, compare the \( \alpha_d \) and \( \beta_d \) values with the Threshold value \( \text{Th}_1 \).

Steps to find a noisy pixel
Step 5: Consider a pixel \( P(i,j) \) and center pixel is represented by \( M(i,j) \).
Step 6: Calculate \( [M(i,j) - \alpha_{\text{max}}] \) and \( [M(i,j) - \alpha_{\text{min}}] \). Then Calculate \( [M(i,j) - \beta_{\text{max}}] \) and \( [M(i,j) - \beta_{\text{min}}] \). The values obtained are compared with the threshold value \( \text{Th}_2 \). If the values exceed the threshold value, then it is a noisy pixel.
Step 7: The edge in the current window is determined by the edge preserving method. In this, only eight different directions (D1 TO D8) in a window are consider to eliminate noisy pixel. The directions through the suspected noisy pixels are eliminated.

For example, if \( a, b, \) and \( c \) is a de-noised pixel and the remaining are noisy, then the directions through \( a, b, \) and \( c \) alone are considered to find the estimated value of a current pixel \( P(i,j) \). It is calculated using the formula \( \frac{a+2xb+c}{4} \).

Step 8: If an edge is considered, find the minimum difference (Dmin) among the chosen directions.
Step 9: Then take the mean of the pixels which constitute the minimum direction.
Step 10: To preserve the edge of the image apply median to the estimated pixel \( P'(i,j) \) and its surrounding four neighboring pixels.

RESULTS

The parameter that determines the image Quality is MSE, RMSE, PSNR etc.,

Mean Square Error: The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE represents the lower value of the error.

The MSE brings out the difference between the original image and the noisy image and averaging the difference with the product of the number of pixels used. The MSE value varies based upon the noise level applied to an image. It is given by the formulae

\[
\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x(i,j) - y(i,j))^2
\]

Algorithm to calculate Mean Square Error:
Step 1: Calculate the difference between the noisy pixel and original pixel.
Step 2: Calculate the size of the matrix
Step 3: Each of the pixels in the matrix has to be squared.
Step 4: Calculate the sum of the matrix obtained in step 3
Step 5: Calculate the MSE by taking the ratio of value in Step 4 to the value in Step 2.
Step 6: Calculate the RMSE by taking the square root of step 5.

Signal to Noise Ratio: The parameter that determines the image quality is given by PSNR values. The increase in PSNR values shows the increase in image quality. The expression for it is given by

\[
\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right)
\]

Algorithm to calculate Peak Signal to Noise Ratio:
Step 1: Calculate the difference between the noisy pixel and original pixel.
Step 2: Calculate the size of the matrix
Step 3: Each of the pixels in the matrix has to be squared.
Step 4: Calculate the sum of the matrix obtained in step 3.
Step 5: Calculate the MSE by taking the ratio of value in step 4 to the value in step 2.
Step 6: Calculate the RMSE by taking the square root of step 5.
Step 7: Calculate the PSNR by dividing 255 with RMSE. Then take log base 10 of that and multiply it with 10.

The Simulation is performed using Matlab version 7.10.0. The image used to study the performance of various de-noising methods is a standard baboon.jpg image.

The Table 1 explains the analysis report MSE values of various filters on the image.

The RMSE of the filtered output images of various filters are given in Table 2.

The Table 3 provides the most important parameter PSNR that determines the quality of an image. The higher the PSNR value higher the quality of an image. It is found to be high for the isolation algorithm.

The images shown below are an output of the removal of 50% salt and pepper noise from the JPEG image. Only the d) wavelet image appeared to be in black and white image since it involves color to grey conversion. The image e) isolation image has better quality than the other filters.

The figure 1.a) represents an original baboon/jpeg image and the output image of mean, median and DBUTM filter is given in figure 1.b), c) and d) respectively. The figure 1.e) and f) gives the output of DWT and isolation method.

The Figure 2 explains the graphical representation of the above discussed filters by plotting noise density Vs PSNR value. The Red colored graph indicates the values of isolation method and is found to be higher than the other methods.

<table>
<thead>
<tr>
<th>Noise Density In %</th>
<th>MSE</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Median</td>
<td>Dbutm</td>
<td>Wavelet</td>
<td>Isolation</td>
</tr>
<tr>
<td>10</td>
<td>72.3966</td>
<td>42.4365</td>
<td>4.9279</td>
<td>21.2888</td>
</tr>
<tr>
<td>20</td>
<td>86.3031</td>
<td>45.2216</td>
<td>10.486</td>
<td>36.7029</td>
</tr>
<tr>
<td>30</td>
<td>93.5664</td>
<td>49.5127</td>
<td>15.847</td>
<td>49.7239</td>
</tr>
<tr>
<td>40</td>
<td>98.3936</td>
<td>54.9268</td>
<td>21.984</td>
<td>58.7196</td>
</tr>
<tr>
<td>50</td>
<td>100.976</td>
<td>62.3416</td>
<td>28.438</td>
<td>72.6914</td>
</tr>
<tr>
<td>60</td>
<td>103.691</td>
<td>72.1617</td>
<td>36.941</td>
<td>79.9328</td>
</tr>
<tr>
<td>70</td>
<td>105.395</td>
<td>82.9187</td>
<td>50.024</td>
<td>96.4967</td>
</tr>
<tr>
<td>80</td>
<td>105.733</td>
<td>96.8950</td>
<td>74.974</td>
<td>109.3794</td>
</tr>
<tr>
<td>90</td>
<td>106.322</td>
<td>112.272</td>
<td>128.51</td>
<td>128.5133</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noise Density In %</th>
<th>RMSE</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Median</td>
<td>Dbutm</td>
<td>Wavelet</td>
<td>Isolation</td>
</tr>
<tr>
<td>10</td>
<td>8.5086</td>
<td>6.4935</td>
<td>2.2198</td>
<td>4.6784</td>
</tr>
<tr>
<td>30</td>
<td>9.6729</td>
<td>7.0566</td>
<td>3.9808</td>
<td>7.0515</td>
</tr>
<tr>
<td>40</td>
<td>9.9193</td>
<td>7.4348</td>
<td>4.6888</td>
<td>7.6628</td>
</tr>
<tr>
<td>50</td>
<td>10.0486</td>
<td>7.8845</td>
<td>5.3327</td>
<td>8.5259</td>
</tr>
<tr>
<td>60</td>
<td>10.1828</td>
<td>8.4590</td>
<td>6.0779</td>
<td>8.9405</td>
</tr>
<tr>
<td>70</td>
<td>10.2662</td>
<td>9.1585</td>
<td>7.0728</td>
<td>9.8232</td>
</tr>
<tr>
<td>80</td>
<td>10.2826</td>
<td>9.8311</td>
<td>8.6587</td>
<td>10.4584</td>
</tr>
</tbody>
</table>
Based on the above analysis isolation method provides better performance than the DWT except for the noise levels 10% and 90%. The wavelet provides better PSNR VALUES than the mean, median and DBUTM. Also the PSNR values of Isolation method are found to be nearer to DWT. But the output image of DWT occurs blurred since the wavelet is applied only to the frequency component.

**REFERENCES**


2. Dodda Shekar and Rangu Srikanth, “Removal of High Density Salt & Pepper Noise in Noisy Images Using Decision Based UnSymmetric Trimmed Median Filter (DBUTM)”,

---

**Table 3. : PSNR values of different filters**

<table>
<thead>
<tr>
<th>Noise Density In %</th>
<th>Mean</th>
<th>Median</th>
<th>Dbutm</th>
<th>Wavelet</th>
<th>Isolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>29.5383</td>
<td>31.8823</td>
<td>41.211</td>
<td>40.9537</td>
<td>38.1039</td>
</tr>
<tr>
<td>20</td>
<td>28.7827</td>
<td>31.5907</td>
<td>37.924</td>
<td>38.5485</td>
<td>38.3195</td>
</tr>
<tr>
<td>30</td>
<td>28.4364</td>
<td>31.1599</td>
<td>36.133</td>
<td>37.3461</td>
<td>38.2735</td>
</tr>
<tr>
<td>40</td>
<td>28.2250</td>
<td>30.7071</td>
<td>34.711</td>
<td>36.4976</td>
<td>37.9450</td>
</tr>
<tr>
<td>50</td>
<td>28.1163</td>
<td>30.1955</td>
<td>33.593</td>
<td>35.7408</td>
<td>37.3500</td>
</tr>
<tr>
<td>60</td>
<td>28.0072</td>
<td>29.5846</td>
<td>32.457</td>
<td>34.9580</td>
<td>36.6927</td>
</tr>
<tr>
<td>70</td>
<td>27.9397</td>
<td>28.8943</td>
<td>31.39</td>
<td>34.3276</td>
<td>35.7620</td>
</tr>
<tr>
<td>80</td>
<td>27.9319</td>
<td>28.2788</td>
<td>29.381</td>
<td>33.8421</td>
<td>34.4912</td>
</tr>
<tr>
<td>90</td>
<td>27.9064</td>
<td>27.6015</td>
<td>27.042</td>
<td>33.0945</td>
<td>32.7771</td>
</tr>
</tbody>
</table>

---

**Fig. 1.** The output images for 50% noise level

(a) Original image (b) mean (c) median (d) DBUTM (e) DWT (f) Isolation

**Fig. 2.** Plot for PSNR values of various filters


