Recursive trimmed filter in eliminating high density impulse noise from digital image

Ilyasa Pahlevi Reza Yulianto, Bima Ramadhan, Achmad Abdurrazzaq*

Indonesia Defense University, Indonesia

ARTICLE INFO

Article History
Received : 04-01-2023
Revised : 04-02-2023
Accepted : 24-02-2023
Published : 30-04-2023

Keywords:
Dynamic Programming; Recursive; Estimator; Impulse Noise.

*Correspondence: E-mail: achmad.abdurrazzaq@idu.ac.id

Doi: 10.24042/djm.v6i1.16382

INTRODUCTION

Digital images often get noise during the process of downloading or uploading from the internet. Therefore, image analysis is used to remove noise by various methods. This noise can also be found during the image digitization process, when taking pictures, as well as the time factor. Impulse noise is one of the types of noise that often arises as a result of the digitization process, there are two types of impulse noise namely random valued noise and salt and pepper noise (SAP). SAP spoils image quality so much such that many methods have been developed to filter this type of noise (Fan et al., 2018; Jayaraman et al., 2009; Karthik et al., 2021; Rosin & Collomosse, 2012; Singh et al., 2018).

A method often used due to its ease of use is the mean filter and median filter, where pixels recognized as noise are replaced by the mean or median of the surrounding 3 × 3 window (Chan et al., 2005; Huang et al., 1979; Lin & Willson, 1988). This method produces good results when the noise from the image is small, because if the noise is large, then the surrounding window full of corrupted pixels (Arias-Castro & Donoho, 2009; Sun & Neuvo, 1994).

To overcome this problem, methods such as the adaptive mean or median filter...
have been developed, where the selected window will be enlarged until undamaged pixels are found and the mean or median will be assigned to that window (Qin et al., 2022). However, when the noise level is high, it will produce a large window intake so that the significance value of free-noise pixels becomes small. This also allows for inaccurate edge selection that causes image edges to become blurred or damaged (Lin & Willson, 1988).

Several methods that have been developed to eliminate SAP noise include the Progressive Switching Median Filter (PSMF) which allows to detect impulse noise and filter noise progressively in an iterative way. The processed noisy pixels will be used for the next iterative process. However, the PSMF method still produces poor results in high noise levels (Wang et al., 2010; Zhang & Wang, 2015; Zhou Wang & Zhang, 1999).

The Modified Decision-Based Unsymmetric Trimmed Median Filter (MDBUTMF) is a derivative of the median filter, where the MDBUTMF works based on the number of noisy pixels in the window. MDBUTMF usually replaces noisy pixels with the surrounding mean or median so it has the same problem as SMF. Because when the noise level is high, the selected free noise becomes less so that the recovery result becomes less significant (Goel et al., 2020).

In the Noise Adaptive Fuzzy Switching Median Filter (NAFSMF) method, the window acquisition is the same as the AMF and the new pixel values are determined using a fuzzy method using two parameters. The inconsistent use of two parameters causes this method to lose quality (Toh & Isa, 2010). Another method that uses the fuzzy method is the Type-2 Fuzzy Filter (T2FF) using the type 2 fuzzy approach to remove noise, the Type 2 fuzzy filter works poorly on high density noise.

Algebraic approaches, especially tropical algebra, have also been proposed to filter images such as in Tropical-Singular Value Decomposition (TSVD), Tropical Algebra based Median Filter (TMF), and Adaptive Tropical Algebra Filter (Abdurrazzaq et al., 2019a, 2019b, 2020; Fan et al., 2018). In TSVD, Singular Value Decomposition is used to obtain the processed window and minimax algebra is used to obtain the recovery pixel value. Almost the same as TSVD, TMF uses Singular Value Decomposition, but for the approach of getting pixel values from TMF recovery using a median filter.

Different Applied Median Filter (DAMF) has good results for low- and high-density levels, DAMF combining an adaptive median filter with an iterative process, and only takes the free-noise pixels (Erkan et al., 2018). The results of the first iteration will be reused for the second iteration process. The difference between the first and second iterations is the window size used, produces a new pixel value that is close to the original value. This method works well for both low- and high-density noise.

Of all the above approaches or methods, there is a decrease in performance for high density noise. Usually this is caused by taking too large a window or taking a recovery pixel value that is based on the previous recovery value and not based on the free-noise pixel value, resulting in an increasingly accumulated error. Therefore, in this paper, a new method is developed to minimize the above shortcomings and filter the noise when the noise level is high by not blurring the image and the edges of the image.

The method that produces pretty good result is Recursive Mean Filter (RMF). RMF is developed by the same author who made DAMF, RMF is created using the sum of noisy pixel around neighborhood of noise as conditions for calling the function again (Recursive) (Erkan et al., 2018, 2019). This method works well for both low- and high-density noise.
noise, as well as good at maintaining the texture and the edge of the image. Although the result of RMF can be improved for high level noise. Therefore, combination of both method DAMF and RMF is created using recursive nature of RMF and Conditional of DAMF, formed Recursive Trimming Mean Filter (RTMF) to minimize the shortcoming of the method that has been developed.

Recursion itself is a procedure or function where within the function there is a step that calls the function itself. A famous example is the Fibonacci function:

\[ f(n) = f(n - 1) + f(n - 2) \]

It indicates that the function \( f \) will be called back to determine the value of the function. The recursive procedure itself is often used in optimization problems in Dynamic Programming. Therefore, a recursive method was chosen to solve this problem (Giegerich et al., 2004).

**METHOD**

The use of recursive methods in the proposed method is so that the window is not too large. This method is performed by using mean or a median filter (in this method using conditional mean) by taking a window of fixed size around the noisy pixels (3 × 3), if there are less than two uncorrupted pixels in the window, the pixels are skipped, and the mean process continues on other pixels. This process will be carried out continuously so that there are no more pixels known as noise.

This method is primarily using mean filter as method for searching the new values for noisy pixel. Trimming is used as method for improving the result of mean filter as trimming reduce the amount of corrupted pixel being processed by cutting the minimum and maximum of uncorrupted pixel so that the result of mean filter closer to original value.

Some basic definitions and notions, as well as the flowchart of the proposed method are given as follows.

**Definition 1.1.2.1.** Let \( A = [A(i,j)]_{mn} \) be an original (corrupted) image with the size \( m,n \) such that \( 0 \leq A(i,j) \leq 255 \). Then \( A(i,j) \) is called noise if \( A(i,j) = 0 \) or \( A(i,j) = 255 \) otherwise it is called uncorrupted pixel.

**Definition 1.1.2.2.** Suppose \( m,n \) is the size of original image \( A \), then \( S_i = m \) and \( S_j = n \).

**Definition 1.1.2.3.** Let \( A \) is the image matrix then \( M \) is true if there are exist of \( A(i,j) \) that is considered as noise.

**Definition 1.1.2.4.** Let \( A(i,j) \) is the processed pixel then the window that’s is computed as \( A(i - 1:i + 1,j - 1:j + 1) \) so that \( x \) is sorted list of uncorrupted pixels in that window.

**Definition 1.1.2.5.** Let \( x \) is a sorted list of uncorrupted pixels then \( l x \) is the length of the list \( x \).

**Definition 1.1.2.6.** Let \( x \) is a sorted list of uncorrupted pixels then trim \( x \) is removing the first and last entry of list \( x \).

**Definition 1.1.2.7.** Let \( A \) be an original(corrupted) image then \( B \) is equal to \( A \) that which already being processed with proposed method.
RESULTS AND DISCUSSION

In this simulation, the proposed method (RTMF) will be compared with existing methods, such as Modified Based Asymmetric Truncated Median Filter (MDBUTMF), Noise Adaptive Fuzzy Switching Median Filter (NAFSMF), Differential Applied Median Filter (DAMF), Fuzzy Type-2 (T2FF), Tropical Algebra based Median Filter (TMF), Thresholding Value Filter (TVF) (Charmouti et al., 2022), and Recursive Median Filter (RMF) using several images with a resolution of 512×512 pixels namely ‘Lena’, 'Plane', 'Mandrill', 'Cameraman', and 'Pepper' with various levels of noise, and quantitative and qualitative comparisons will be made for the tested methods (Image Database. (n.d.), n.d.).
The quantitative comparison is PSNR and SSIM. PSNR and SSIM are used to evaluate the quality of the processed image based on the uncorrupted image. PSNR calculates the ratio of the maximum signal value to the amount of noise affecting an image and is usually measured in decibels (dB) (Faragallah et al., 2021). While SSIM measures the loss or error of the processed image against the uncorrupted image. The following equation is used to calculate PSNR and SSIM:

\[ \text{PSNR}(x, y) = \frac{10 \log_{10}(255 \times 255)}{MSE} \]

and MSE:

\[ MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij} - y_{ij})^2 \]

Where \( m, n \) is the image size, \( x_{ij} \) = the pixel value of \( i,j \) in the uncorrupted image, and \( y_{ij} \) = the pixel value of \( i,j \) in the processed image.

\[ \text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\mu_x^2 + \mu_y^2 + C_1(\sigma_x^2 + \sigma_y^2 + C_2)} \]

where \( \mu_x = \text{mean pixel of} \ x, \mu_y = \text{mean pixel of} \ y, \sigma_{xy} = \text{covariance of} \ x \text{ and} \ y, \sigma_x^2 = \text{variance of} \ x, \sigma_y^2 = \text{variance of} \ y, \) and \( C_1, C_2 \) as independent variables for comparison of processed and uncorrupted images.

### Table 1. PSNR and SSIM average results of the existing methods from multiple images with different noise levels

<table>
<thead>
<tr>
<th>Noise</th>
<th>Assess</th>
<th>MDBU</th>
<th>T2FF</th>
<th>TSVD</th>
<th>TMF</th>
<th>TVF</th>
<th>RMF</th>
<th>DAMF</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>PSNR</td>
<td>36.43</td>
<td>38.52</td>
<td>39.66</td>
<td>39.75</td>
<td>40.29</td>
<td>41.34</td>
<td>40.30</td>
<td>40.09</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9735</td>
<td>0.9850</td>
<td>0.9864</td>
<td>0.9861</td>
<td>0.9872</td>
<td>0.9898</td>
<td>0.9873</td>
<td>0.9891</td>
</tr>
<tr>
<td>0.2</td>
<td>PSNR</td>
<td>32.77</td>
<td>34.77</td>
<td>35.45</td>
<td>35.39</td>
<td>36.42</td>
<td>37.99</td>
<td>36.35</td>
<td>36.76</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9326</td>
<td>0.9673</td>
<td>0.9686</td>
<td>0.9685</td>
<td>0.9721</td>
<td>0.9790</td>
<td>0.9725</td>
<td>0.9771</td>
</tr>
<tr>
<td>0.3</td>
<td>PSNR</td>
<td>31.60</td>
<td>32.12</td>
<td>32.52</td>
<td>32.55</td>
<td>34.02</td>
<td>35.38</td>
<td>34.01</td>
<td>34.65</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9280</td>
<td>0.9450</td>
<td>0.9448</td>
<td>0.9466</td>
<td>0.9469</td>
<td>0.9634</td>
<td>0.9551</td>
<td>0.9630</td>
</tr>
<tr>
<td>0.4</td>
<td>PSNR</td>
<td>30.32</td>
<td>29.66</td>
<td>30.30</td>
<td>30.45</td>
<td>32.13</td>
<td>32.27</td>
<td>32.09</td>
<td>32.91</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9066</td>
<td>0.9136</td>
<td>0.9137</td>
<td>0.9191</td>
<td>0.9345</td>
<td>0.9446</td>
<td>0.9348</td>
<td>0.9468</td>
</tr>
<tr>
<td>0.5</td>
<td>PSNR</td>
<td>29.10</td>
<td>27.31</td>
<td>28.14</td>
<td>28.52</td>
<td>30.63</td>
<td>31.43</td>
<td>30.51</td>
<td>31.31</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.8782</td>
<td>0.8686</td>
<td>0.8694</td>
<td>0.8845</td>
<td>0.9112</td>
<td>0.9212</td>
<td>0.9109</td>
<td>0.9265</td>
</tr>
<tr>
<td>0.6</td>
<td>PSNR</td>
<td>28.04</td>
<td>25.29</td>
<td>26.12</td>
<td>26.74</td>
<td>29.21</td>
<td>29.60</td>
<td>29.10</td>
<td>29.73</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.8454</td>
<td>0.8108</td>
<td>0.8070</td>
<td>0.8369</td>
<td>0.8825</td>
<td>0.8893</td>
<td>0.8820</td>
<td>0.9002</td>
</tr>
<tr>
<td>0.7</td>
<td>PSNR</td>
<td>27.01</td>
<td>23.17</td>
<td>24.01</td>
<td>24.71</td>
<td>27.74</td>
<td>26.74</td>
<td>25.64</td>
<td>28.21</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.8082</td>
<td>0.7349</td>
<td>0.7181</td>
<td>0.7729</td>
<td>0.8461</td>
<td>0.8423</td>
<td>0.8444</td>
<td>0.8666</td>
</tr>
<tr>
<td>0.8</td>
<td>PSNR</td>
<td>25.78</td>
<td>20.97</td>
<td>21.72</td>
<td>22.65</td>
<td>25.93</td>
<td>25.37</td>
<td>26.04</td>
<td>26.44</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.7608</td>
<td>0.6390</td>
<td>0.5878</td>
<td>0.6827</td>
<td>0.7940</td>
<td>0.7711</td>
<td>0.7958</td>
<td>0.8174</td>
</tr>
<tr>
<td>0.9</td>
<td>PSNR</td>
<td>22.36</td>
<td>18.45</td>
<td>18.51</td>
<td>20.03</td>
<td>23.25</td>
<td>22.64</td>
<td>23.74</td>
<td>23.93</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.6461</td>
<td>0.5229</td>
<td>0.4015</td>
<td>0.5434</td>
<td>0.7043</td>
<td>0.6636</td>
<td>0.7127</td>
<td>0.7335</td>
</tr>
</tbody>
</table>

It can be seen from the PSNR value contained in the table, that the PSNR value using the proposed method (RMF) has a higher value than the PSNR value of other methods at various noise levels. This shows that using the proposed method produces an image that can restore noisy pixel values to be close to the original values. Similar to PSNR, SSIM also produces larger results than the other methods at various noise levels, indicating a loss of information from the resulting image. less processed than other methods.
**Figure 2.** Graph of changes in PSNR values from several existing methods.

**Figure 3.** Filtering results from several methods for the image 'Plane' with resolution $512 \times 512$ and noise level 80%. (a) Noise, (b) MDBUTMF, (c) T2FF, (d) TSVD, (e) TMF, (f) DAMF, (g) TVF, (h) RMF

It can be seen that the image with a high noise level of 80%, the proposed method can filter the image well, RMF can filter while still preserving the edges of the image, it can be seen that the F18 text can still be seen in the RMF image, while the other methods have blurred edges, so the outline of the plane or writing cannot be seen clearly.
Similar to the noise level is 80%, when the noise level is 90% the edges of the plane can be seen and the writing can still be seen clearly. This means that this method can correctly preserve the outline or shape of an image with a high level of noise.
The image processed using RMF seems to be able to clean or filter the noise from the existing satellite image, it seems that the shape of the building is still clearly visible even if the noise level is as high as 90%. The results of the 'Aircraft' image and the satellite image above show that RMF can be an image filtering tool to overcome interference or noise present in defense imagery images that demand higher detail than other fields.

CONCLUSIONS AND SUGGESTIONS

In this paper, a new filtering method has been proposed that is capable of working at high noise levels. This method produces an image with clear edges so that the outline or shape of an object is still maintained. This method takes a window that is not large so that the estimated value is not too different from the pixel values of the surrounding pixels. After that, it is also shown that the Recursive Median Filter is better in filtering images that have a high level of noise compared to other methods according to PSNR and SSIM. For future studies, the process of determining new pixel values should be changed in addition to using a mean estimator, this aims to increase the sensitivity when determining pixel values. For further research, it is necessary to improve the performance of this method so that the resulting output has clearer and smoother edges. In addition, this method needs to be developed to eliminate impulse noise in general.

REFERENCES


https://doi.org/10.1049/iet-ipr.2018.6201


