**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

International Conference on Mobility in Computing- ICMiC13, Organized by Mar Baselios College of Engineering and Technology during December 17-18, 2013 at Trivandrum, Kerala, India, pg.158 – 165

**SURVEY ARTICLE**

# Iterative Average Estimation Filter using BDND Algorithm for the Removal of High-Density Impulse Noise

**[1]Sruthi Ignatious, [2]Robin Joseph**
Department of Computer Science and Engineering, Mar Baselios College of Engineering and Technology, Trivandrum, India
[1] sruthieliza@gmail.com, [2] robin.joseph@mbcet.ac.in

***Abstract*— *The Boundary Discriminative Noise Detection (BDND) is one of the powerful methods for detecting the noise in the image. In this paper we are updating our detection map using the BDND algorithm. The iterative algorithm for filtering searches the noise-free pixels within a small neighbourhood, and then the noisy pixel is replaced with the average estimated value from noise-free pixels. This iterative process continues until all noisy pixels of the corrupted image are filtered. The experimental result shows that proposed scheme removes high density impulse noise and consumes less time compared to other filters.***

***The proposed algorithm promises less execution time and optimum size, which would be beneficial for smart phone application.***

***Keywords: Impulse Noise; Noise-free pixel; Noise detection; Average estimation.***

## I. INTRODUCTION

Noise is one of the major factors affecting the image, which is mainly produced in the processes of image acquisition, storage and transmission, thereby degrading the quality of images. Therefore a common problem in applied science and engineering is the restoration of the corrupted pixels (noisy pixels) included in the image. Image filtering not only improves the image quality but also is used as a pre-processing stage in many applications including pattern recognition, image compression, etc..There are many methods for removal of impulse noises from the images.

The impulse noise has two main properties [8]: a) Certain percentage of images pixels are corrupted with noise, b) The intensity of corrupted pixels is significantly different from noise free pixels. An image X corrupted with impulse noise can be defined as:

$$X_{i,j} = \begin{cases} n(i,j) & for \quad k \\ f(i,j) & for \quad 1-k \end{cases} \qquad (1)$$

where n(i,j) is the element of $[I_{min}, I_{max}]$ is the impulse noisy pixel which is located at (i,j), k represents the probability of noisy pixels and f(i,j) denotes the noise free pixels.Commonlly there are two types of impulse noise models. They are salt-and-pepper noise and random-valued noise. In this work we use salt-and-pepper noise for highlighting the performance. In salt and pepper noise $I_{min}$ will be '0' and $I_{max}$ will be '255'.

Switching median filters are simple and more effective than median filters [2] [3]. There are different methods for detecting the impulse noise present in the image. They are: neural approach [4], fuzzy approaches [5-7] and boundary based approaches [9][10][12]. Among the three categories boundary based approach [9][10][12] is preferred due to its simplicity compared to computational complexity and system structure of other two categories. The noise removal techniques as proposed in Iterative Average Estimation Filter (IAEF) [8] is capable of removing high density of impulse noise effectively and Iterative Switching Filter for removal of high Density Impulse Noise[11] is capable of removing high density impulse noise but it is more time consuming. The noise removal technique as proposed in Noise Adaptive Weighted Switching Median Filter[12] is using BDND algorithm for detection map construction and the filtering is done using weighted median so it will consumes more time. The proposed method is giving significant improvement in time consumption than previous methods.

## II. CONSTRUCTION OF DETECTION MAP USING BDND ALGORITHM

The basic idea behind the BDND algorithm [9][10] is to observe each and every adjacent pixel from coarse to fine. The BDND algorithm starts by classifying the pixel into three categories- lower intensity pixel, medium intensity pixel and high intensity pixel. If the central pixel is within the medium intensity range, all the pixels within this range is considered as uncorrupted. Otherwise the pixels are considered as corrupted. So inorder to get an accurate intensity range we have to find accurate boundary values.

This algorithm is applied to every pixel within the noisy image, to identify whether the pixel is corrupted or uncorrupted. When this algorithm is applied to the entire image, a two dimensional binary detection map is generated, contains only zeros and ones. The value zeros in the detection map indicates uncorrupted pixels and ones indicates corrupted pixels.

All the pixels centered the current pixel will be grouped into three cluters. For that we require two boundaries. These boundaries $b_1$ and $b_2$ are to be determined. Each pixel is processed. If $0 \leq X_{i,j} \leq b_1$, then the pixel will be assigned to lower intensity, otherwise to the medium intensity cluster for $b_1 < X_{i,j} \leq b_2$, or to the high density cluster for $b_2 < X_{i,j} \leq 255$. If the central pixel falls into the middle cluster, then that pixel is treated as uncorrupted, otherwise it treated as corrupted.

The BDND algorithm [9] consists of two iterations. The second iteration will be invoked conditionally. In the first iteration an a large window size of 21x21 is considered for determining whether the pixel is an uncorrupted one. If the condition is succeeded then there will not be any further iteration, otherwise the second iteration will be invoked for examining the pixel based on a 3X3 window. In summary the steps involved in the BDND algorithm [9] are given below:

Step 1: Impose a 21X 21 window centered around $X_{i,j}$, where $X_{i,j}$ is the current pixel in the image.

Step 2: Sort the pixels within the window in ascending order and store it to $V_0$ and find the median of $V_0$ and store the result to *med.*

Step 3: Calculate the intensity difference of each pair of adjacent pixels within the vector $V_0$ and store the result to the difference vector $V_D$.

Step 4: Find the pixels from $V_0$ that corresponds to the maximum differences in the intervals of [0, *med*] and [*med*, 255].

Set these pixel's intensities as the decision boundaries *b1* and *b2* respectively.

Step 5: If the pixel that we are processing belongs to the middle cluster it is classified as uncorrupted and the classification process stops.

Otherwise we must consider the second iteration, which will be invoked as follows;

Step 6: Impose a 3X3 window, being centered around the concerned pixel and repeat steps 2 to 4.

Step 7: If the current pixel belongs to the middle cluster, it is classified as "uncorrupted" pixel. Otherwise it is corrupted.

Based on this algorithm we are updating the detection map. If the pixel that we are processing is uncorrupted the detection map is updated with "0" otherwise detection map is updated with "1".

## III. IMPULSE NOISE FILTERING STAGE

The noise filtering that we are discussing here is an iterative average estimation model [8]. Around each pixel location (i,j) of the noisy image *X*, select a small window $W_{i,j}$ of size 3X3 and the detection map *D*. Use of smaller size window for the computation may be too efficient and effective. So we use small window size for computation. By applying a small window of size 3X3, we obtain the patch of detection map in $W^d_{i,j}$ and the patch of noisy image in $W^x_{i,j}$. The patches $W^x_{i,j}$ and $W^d_{i,j}$ are shown below :

$$W^x_{i,j} = \begin{bmatrix} x_{i-1,j-1} & x_{i,j-1} & x_{i,j+1} \\ x_{i-1,j} & x_{i,j} & x_{i+1,j} \\ x_{i-1,j+1} & x_{i,j+1} & x_{i+1,j+1} \end{bmatrix},$$

$$W^d_{i,j} = \begin{bmatrix} d_{i-1,j-1} & d_{i,j-1} & d_{i,j+1} \\ d_{i-1,j} & d_{i,j} & d_{i+1,j} \\ d_{i-1,j+1} & d_{i,j+1} & d_{i+1,j+1} \end{bmatrix}$$

We compute the count value after each iteration, count is the number of corrupted pixels in the detection map D. The value of count is denoted by the alphabet K. If K is a positive integer and the central pixel $X_{i,j}$ within current window is corrupted, then we maintain an array *R*. This array is populated with the uncorrupted pixels or noise free pixels.

To estimate the value of noisy-pixels within the window, we emphasize a constraint of minimum three noise-free pixels ($M_f$=3) and noise free pixels in the array *R*. If the above condition is satisfied, then place the central noisy pixel with the estimated value, i.e.,

$$g_{i,j} = \begin{cases} e_s \text{ ,if } d_{i,j} = 1 \vee L(R) \geq M_f \\ x_{i,j} \text{ , Otherwise} \end{cases} \quad (2)$$

where $e_s$ is the estimated value for the noisy pixel, and L(R) =length(R) is the length of the array R. The value of $e_s$ is computed as:

$$e_s=\frac{1}{L(R)}\sum_{t=1}^{L(R)} Ri \qquad (3)$$

### IV. UPDATE NOISY PIXELS IN DETECTION MAP

If the corrupted pixel(noisy pixel) within the image is replaced with the estimated value $e_s$, then the corresponding detection map, D is also updated by the new value. The updation of detection map is done by changing the corresponding entries to zero. It is shown below:-

$$d(i,j)=\begin{cases} 0, if\ d(i,j) = 1 \ \vee \ L(R) \geq Mf \\ d(i.j), Otherwise \end{cases} \qquad (4)$$

Each iteration produces a refined image G. At the end of each iteration the detection map is also updated. After some iterations, all the entries in the detection map become zeros. At that point the updation and the iteration processes are terminated. We obtain an image G, which is clear from salt-and-pepper noise.

### V. PROPOSED ALGORITHM

The proposed iterative average estimation filter using BDND algorithm will create a detection map, which is a binary matrix of zeros and ones, where the noisy pixels in the input image are represented as ones and noise free as zeros. Depending on the number of uncorrupted pixel average of the noise free pixel is found and replaces the noisy pixel.

Step 1: Obtain noisy image as input.
Step 2: Construct the detection map using BDND algorithm.
Step 3: Check the detection map to find if there are any noisy pixels. If so do a)-f)

a) Consider each noisy pixel $P_{i,j.}$
b) Select a 3X3 window with central pixel $P_{i,j}$ as processing pixel.
c) If $P_{i,j}$ is noisy goto d), Otherwise goto a).
d) Check whether the number of noisy pixel is less than 3. If it is true goto e), otherwise goto a).
e) Compute the estimate value and replace $P_{i,j}$.
f) Update the detection map and also the image.

Step 4: Display the new image.

### VI. EXPERIMENTAL RESULTS

The execution time of the proposed algorithm has been evaluated using the tic-toc command. The command tic starts a stop watcher timer for measuring the performance of the proposed algorithm. The function toc records the internal time at execution of the tic command. The elapsed time will be displayed by the toc function. It uses another value called timeVal. This is the value of the internal timer at the execution of the tic command. This value will be used as an input argument for the subsequent call of the function, toc.

The test images that are used for evaluation are Lena image of size 512x512, Boat 512x512. The execution time of this algorithm is compared with that of Weighted Median Filter (WMF) [11] and a simple Median Filter.

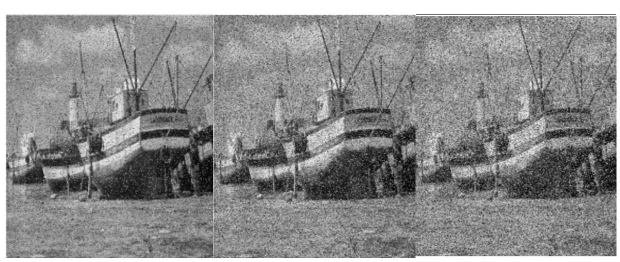
(a) (b)

Fig 1. Standard test image of (a) LENA, (b) BOAT

(a) (b) (c)

(d) (e) (f)

Fig 2. Images corrupted with salt-and-pepper Noise (a) with 10%, (b) with 20%, (c) with30%, (d)-(f) Results after the Proposed filtering for the respective noisy images



(a) (b) (c)

(d) (e) (f)

Fig 3. Images corrupted with salt-and-pepper Noise (a) with 10%, (b) with 20%, (c) with30%, (d)-(f) Results after the Proposed filtering for the respective noisy images

Table I. Execution time for the LENA image corrupted with various Noise Densities.

| Noise (%) | WMF (sec) | SMF (sec) | Proposed (sec) |
|---|---|---|---|
| 10 | 26.254329 | 27.557449 | 14.490290 |
| 20 | 30.153692 | 28.954053 | 14.729608 |
| 30 | 31.834289 | 29.213969 | 15.184809 |

Table II. Execution time for the BOAT image corrupted with various Noise Densities

| Noise (%) | WMF (sec) | SMF (sec) | Proposed (sec) |
|---|---|---|---|
| 10 | 32.143284 | 29.206402 | 13.654468 |
| 20 | 36.517433 | 29.766694 | 15.586077 |
| 30 | 41.178144 | 30.201843 | 17.417527 |

From Table I and Table II we can conclude that the proposed algorithm consumes less execution time compared to the other filters. Hence when time is considered as a major factor, the proposed method gives a better result than others. The most interesting factor is that as the time consumption is reduced, the performance of the system is not much affected. But if the noise intensity present in the image is large, the quality of the image is inferior. So in smart phone applications, where time is a major factor and the image quality is not so important, the proposed algorithm promises better performance.

## VII. FUTURE ENHANCEMENT

The proposed filtering technique is capable of filtering gray scale images. However it cannot be used for JPEG images, because it contains three components R,G and B. JPEG images can be filtered by separating the three components and filtering them separately. Since the proposed method is less time consuming, it will give a better result for JPEG images also.

## REFERENCES


[1] N. I. Petrovic, and V. Crnojevic, "Universal Impulse Noise Filter Basedon Genetic Programming," *IEEE Transactions on Image Processing*,vol. 17, pp. 1109-1120, 2008.

[2] Civicioglu, P.: Using Uncorrupted Neighborhoods of the Pixels for Impulsive Noise Suppression With ANFIS. IEEE Transactions on Image Processing 16(3) (March 2007)

[3] Nair, M.S., Revathy, K., Tatavarti, R.: An Improved Decision-Based Algorithm for Impulse Noise Removal. In: International Congress on Image and Signal Processing - CISP 2008, vol. 1, pp. 426–431. IEEE Computer Society Press, LoAlamitos(2008), doi:10.1109/CISP.2008.21

[4] Nair, M.S., Raju, G.: A new fuzzy-based decision algorithm for high-density impulse noise removal. Signal Image and Video Processing, doi:10.1007/s11760-010-0186-4

[5] Luo, W.: Efficient removal of impulse noise from digital images. IEEE Transaction on Consumer Electronics 52(2), 523–527 (2006) Impulse Noise Detection and Reduction Method. IEEE Transactions on Image Processing 15(5),1153–1162 (2006)

[6] Schulte, S., Nachtegael, M., Witte, V.D., Weken, D.V., Kerre, E.E.: A Fuzzy Impulse Noise Detection and Reduction Method. IEEE Transactions on Image Processing 15(5), 1153–1162 (2006)

[7] Schulte, S., Witte, V.D., Nachtegael, M., Weken, D.V., Kerre, E.E.: Fuzzy Two-Step Filter for Impulse Noise Reduction From Color Images. IEEE Transactions on Image Processing 15(11), 3567–3578 (2006)

[8] Abdul Majid, Muhammad Tariq Mahmood:A Novel technique for removal of high density impulse noise from digital images.In 2010 $6^{th}$ International Conference on Emerging Technologies(ICET)

[9] Ng, P.-E., Ma, K.-K.: A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images. IEEE Transactions on Image Processing 15(6), 1506–1516 (2006)

[10] Iyad F.jafar,Rami A.AlNa'mneh,and Khalid A.Darabkh: Efficient Improvements on the BDND Filtering Algorithm for the Removal of High-Density Impulse Noise. IEEE Transactions on Image Processing, vol.22,No. 3, (1223-1232).March 2013

[11] Jisha John, Ann Mary Jacob, Mekha Prasannan, Priyanka Suja Pradeep, Sruthi Ignatious: Iterative Switching Filter for high density impulse noise. International Journal for Science and Engineering Research (IJSER) Vol.3 Issue 7, July 2012.

[12] Madhu S Nair and P.M. Ameera Mol: Noise Adaptive Weighted Switching Median Filter for removing High Density Impulse Noise. A. Abraham et al. (Eds.): ACC 2011, Part III, CCIS 192, pp. 193–204, 2011.Springer-Verlag Berlin Heidelberg 2011