Journal of King Saud University – Computer and Information Sciences 34 (2022) 782–792

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Improved approximated median filter algorithm for real-time computer vision applications

Obed Appiah *, Michael Asante, James Benjamin Hayfron-Acquah

Kwame Nkrumah University of Science and Technology, University of Energy and Natural Resources, Ghana

ARTICLE INFO

Article history:
Received 24 December 2019
Revised 25 February 2020
Accepted 2 April 2020
Available online 15 April 2020

Keywords:
Image Denoising
Median Filtering
Approximated Median Filter
Fast Median Filter
Real-time Image Processing

ABSTRACT

Median filter is one of the predominant filters that are used to suppress impulse noise. Its simplicity and ability to maintain edges has led to an extensive application in the domain of image processing and computer vision. However, challenges such as moderate to high running time of the standard median filter algorithm and relatively poorer performance when the image is highly corrupted with impulse noise, have led to the design of several variations of the algorithm. One set of variation of the algorithm concentrates on generating quality outputs, while the other set focuses on reducing running time. Among the set targeting the reduction of the running time of the median filter is the DP approximated median filter. However, DP performs poorly when images are corrupted with moderate to high levels of noise. This paper therefore proposes an Improved Approximation Median Filtering Algorithms (IAMFA-I & IAMFA-II) based on DP to generate a better output. The introduction of Mid-Value-Decision-Median in DP reduces the chances of selecting corrupted pixel for denoised image. Experimental results indicate that the IAMFA-II has better running time and equivalent output compared with DP, while IAMFA-I generates better output and has equivalent running time when compared with DP.

1. Introduction

In denoising impulse noise, the rank-order information of pixel's intensity within a kernel or window has been widely used. A class of filters that has extensively explored this concept is the median filters which are robust non-linear filter (Pitas and Venetsanopoulos, 1992). The median filter was proposed by Tukey (1977) as an effective approach for smoothing signals and since then has been successfully used to handle impulse noise. The impulse noise which is sometimes known as "salt and pepper" noise is effectively suppressed by this filter and also able to preserve edges in images after the filtering process is done (Szeliski, 2010). The median filter however has various challenges such as its inability to deal effectively with all kinds of noise introduced into an image (example; the Gaussian noise), relatively high running time, and does not use rigorous statistical operations to effectively denoise an image (Asano et al., 1991). Its moderate to high computational cost (Huber, 1981; Hampel, et al. 1986; Stewart, 1999) has led to various proposed modifications of the median filter to help tackle this problem. The computational cost associated with this algorithm is generally due to its median value selection subtask (sort-based or histogram-based) (Huang et al., 1979; Ko and Lee, 1991; Kasparis, et al., 1992; Chan et al., 2005; Perreault and Hebert, 2007). Again, the Standard Median Filter (SMF) alters pixels even when they are noise free, that is, the algorithm changes the central pixel's values if the value is not the same as the median of a window, which sometimes leads to poor outputs. Variations of the median filter can be seen today with each algorithm attempting to tackle one issue or the other related to the median filter. Weighted Median Filter (WMF) proposed by Arce and Parades (2000) for example did not use the direct rank-order information of pixels' intensities within the filtering window for the filtering task. The Iterative Median Filter (IMF) proposed by Forouzan and Araabi (2003), Weighted Median Filter (WMF) proposed by Liu et al. (2015), Centred Weighted Median Filter (CWMF), Centred Median Filter (CMF), Direction Median Filters (DMF), Adaptive Median Filter (AMF), and Recursive Median Filter (RMF) are various

* Corresponding author.
E-mail addresses: obed.appiah@uenr.edu.gh (O. Appiah), mickasst@yahoo.com (M. Asante), jbhayfron-acquah.cos@knust.edu.gh, jbha@yahoo.com (J.B. Hayfron-Acquah).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2020.04.005
1319-1578/

modified versions of the median filter with the aim of improving outputs or the running time of the standard median filter or both.

The underlining data structure required to select the median value of a given window and the sorting technique used for the estimation of a median value affects the time complexity of the median filter. For example, if a sliding window of size n is used for performing median filtering, then the number of elements in a square window can be considered as n^2. If the quick sort is implemented, then the worst case (big-O) for a given window to be sorted will be $\boldsymbol{O(n^2.n^2) = O(n^4)}$. If this operation is repeated for the entire image, then the worst case scenario for performing median filtering using the quick sort as a subtask can be estimated as $\boldsymbol{O(NM).O(n^4) = O(MNn^4) \approx O(N^2n^4)}$ as N approaches M. This moderate to high running time of the median filter makes it difficult for real-time image processing applications. The running time of the filter which is generally proportional or equivalent to the sorting problem in theory and practice was demonstrated by Suomela (2014). The IMF proposed by Forouzan and Araabi (2003) for example has a high running time due to the repetitive access of image matrix by the algorithm. The need to tackle the running time of the median filter in order to achieve real-time has become important because a lot of these filters are used as pre-processing algorithms for computer vision tasks and if vision tasks are to be accomplished in real-time then all pre-processing must also be delivered in real-time.

2. Comparative analysis of real-time median filters

The need to achieve real-time performance of the median filter has warranted a lot of proposed algorithms over the years. All these algorithms basically tackle the approaches that are used to estimate the median value for the replacement of the central pixel value in a given window. Two (2) main approaches have been observed in literature over the years and they are the histogram based approach for the estimation of the median value and the approximated techniques for median estimation. In this section, we discuss some of the proposed median filtering algorithms based on the categories mentioned in order to achieve real-time performance.

An algorithm proposed by Paeth (1990) successfully reduced the number of comparison required by the sort-based median filter to 20 comparisons when a window of size 3×3 was used. The approach smartly considered sorting a window partially so that the time required to select the median is halved. For real-time processing on CPU today, the number of comparisons for the window is still high and therefore may cause some form of bottleneck in real-time computer vision application.

Work by Singh (2011) acknowledged the high computational time complexity associated with the median filter and therefore proposed an alternative algorithm for 3×3 median filtering of digital images. The algorithm is capable of processing two (2) adjacent pixel values in one step instead of one (1) pixel value as done by most median filtering algorithms. Fig. 1 presents the concept of the alternative algorithm for 3×3 median filter. When a window slides, the sorted values of the overlapping columns in the adjacent windows are copied from the previous operation and used in the next operation. The merge sort is used for combining values in the window and the median estimated for the two (2) pixels value for the denoised image.

Pixels at I(x,y) and I(x + 1,y) medians is computed in one step by taking about 19 comparisons. That is for each location (x, y), an average of 9.5 comparisons is performed at the worst case scenario. Even though the algorithm processes the median of a 3×3 window, a window of size 3×4 is processed at a time and the window is made to slide 2 columns instead of 1 column. The overhead cost of merged sort may result in the algorithm taking much time to complete than expected. The work proposed the used of nested IF statements instead of loops which increase the number of comparisons if not properly implemented.

Huang et al. (1979) proposed a fast two-dimensional median filtering algorithm which uses the histogram approach for the estimation of the median value in order to reduce the running time of the median filter. 2r + 1 pixels are added to and subtracted from the kernel's histogram when moving from one pixel to the next. This algorithm's fast performance is based on the fact that when the sliding window moves one column for the next median value to be calculated, the overlapping values of the adjacent windows are maintained for processing. Given a window of size $n \times n$, the r (radius of a window) is estimated as (n-1)/2. The estimated running time of O(r) is considered better than the use of the sort based approach, especially with increasing window sizes, but poorer for smaller sizes of n due to the overhead cost of maintaining the histogram (Marcus and Ward, 2013).

Weiss (2006) propose a logarithmic-time median filter algorithm, scalable to any radius and adaptable to images of any bit-depth. The algorithm uses hierarchy of histograms and executes with running time of $O(\log r)$. A running time better than the approach proposed by Huang et al. (1979). The algorithm was however complicated to apply in practice (Bae and Yoo, 2018). The running times for smaller window sizes are hampered by the overhead cost of maintaining this hierarchy of histograms.

Perreault and Hebert, 2007) propose a fast and simple median filter algorithm which had running time and storage scale in $O(1)$ as the kernel radius varies. The proposed algorithm has a straightforward instruction-level parallelism which makes it ideal for CPU-based as well as custom hardware implementation. The parallelism mechanism make it possible to achieve $O(1)$ even though not every platform can handle the parallelism. The algorithm works best when compared with the Huang et al. (1979) and Weiss (2006) methods. The high performance of the algorithm is largely due to the concept of parallelism, not the underlining median value selection method. In the case where the implementing device cannot run parallel execution, the algorithm will work just like the other histogram based approaches and hence constrains as mentioned.

Zhu and Huang (2012), Zhang et al. (2014) all proposed median filtering algorithms based on histogram generation.

The challenge of maintaining histogram or array comes in during the implementation phase. This is observed especially when the generation of memory addresses requires multiplication operation which is practically expensive to perform.

Directional Median Filter (DMF), sometimes referred to as stick median filter basically operates by splitting 2D filter windows into a couple of 1D components called sticks. Works by Narendra (1981) and Appiah et al. (2016) confirms that partitioning a window and finding median of each partition after which all the median estimated combine to generate the final median can be an effective way of achieving real-time and still generate excellent outputs. These methods are some form of approximating the median value. Czerwinski et al. (1995) proposed the used of Equation (1) for the estimation of an approximated median for performing the Directional Median Filtering task.

$$S(i,j) = max\{Median(k,l) \in W\Theta\{D(i+k,\ j+l)\} \quad (1)$$

In the work, the output intensity is defined as the largest of all the median values estimated from the partitions or sticks. The final selection method for the median affects the output of the algorithm. The Directional approach is a form of approximation that completes the median filtering near $\boldsymbol{O(n)}$.

Lu et al. (2010) used a concept similar to the directional median filter in their "Sort Optimization Algorithm of Median Filtering Based on FPGA". In the work, a 3×3 window was logical divided

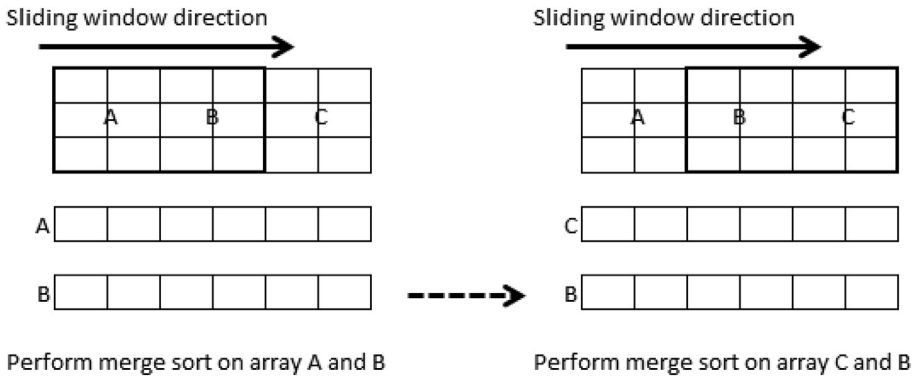

Fig. 1. Illustration of Alternative Algorithms for 3 × 3 (AA3x3) MF for digital images.

into three sticks or columns. The minimum, median, and maximum of the first, second, and third columns were estimated respectively. The median of these values (minimum of 1st Column, median of 2nd Column, maximum of 3rd Column) is estimated to generate the denoised image. Fig. 2 illustrates the concept with columns A, B and C. Equation (2) is used to generate the median value.

$$F_{med} = med(A_{min}, B_{med}, C_{max}) \quad (2)$$

The approach was able to significantly reduce the number of comparisons required by sort based median filtering. However, the use of minimum and maximum could result in the selection of a corrupted pixel for denoised image. Again, the rank-order operations involved with the selection of the minimum and maximum also require some comparisons which may affect the overall running time of the algorithm.

Marcus and Ward (2013) proposed a new non-discrete algorithm (DP) that quickly approximates a median filter. The algorithm combines two principles proposed by Narendra (1981) and Huang et al. (1979) by exploiting data overlap as a window slide during the filtering process. The 3 × 3 kernel implementation of DP partitions a window into 3 columns and the median for each column is calculated. The median of the three (3) medians is also finally estimated. Fig. 3 illustrates how a window is processed as it slides through the image from the left edge to the right of the image. The strength of DP is based on a technique that prevents the resorting of overlapping columns of adjacent sliding windows. That is for a 3 × 3 window or kernel, 6 pixel values overlaps as the window slide each stride. This makes the estimation of the median completed below **O(n)**.

The main challenge of DP is that; the algorithm may replace non-corrupted pixel with corrupted one which leads to poor outputs generated after the denoising operation.

Algorithm 1: DP – Fast Median Filter Approximation

```
Let I represent the image to be filtered
Let H be the HEIGHT of the Image
Let W be the WIDTH of the Image
for i = 2 to H − 1
   col1 ← median(I(i-1, 1), I(i, 1), I(i + 1, 1))
   col2 ← median(I(i-1, 2), I(i, 2), I(i + 1, 2))
   for j = 3 to W −1
      col3 ← median(I(i-1, j), I(i, j), I(i + 1, j))
      I'(i, j) ← median([col1; col2; col3])
      col1 ← col2
      col2 ← col3
   end for
end for
```

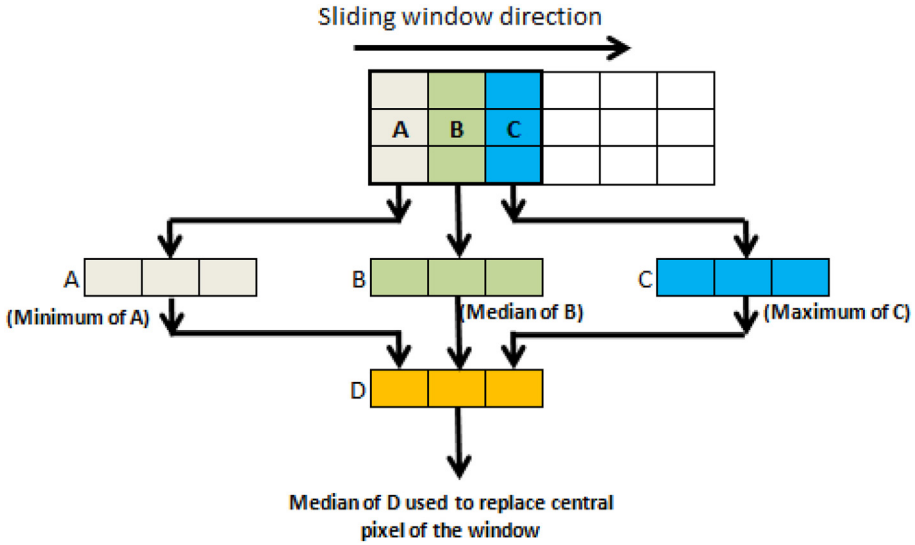

Fig. 2. Illustration of Sort Optimization Algorithm of Median Filtering (SOAMF) Based on FPGA.

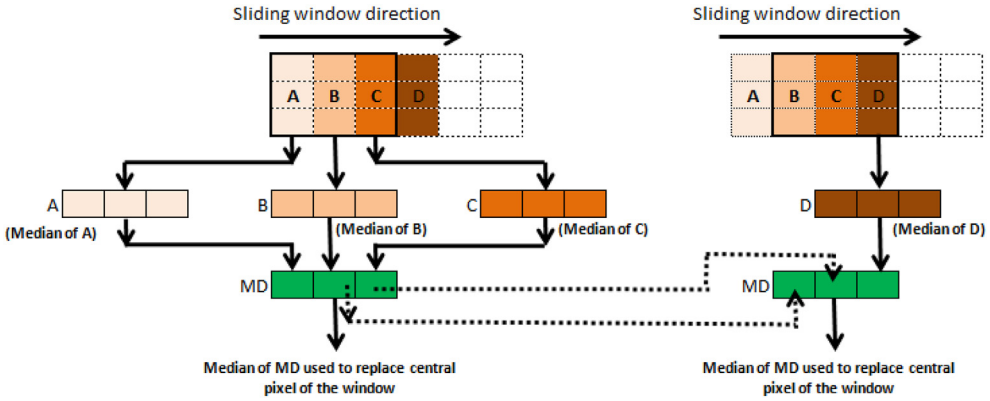

Fig. 3. An illustration of DP: Fast Median Filtering Algorithm.

Marcus and Ward's paper compared the running time of DP with a couple of median filtering algorithms. Fig. 4: illustrates results of the relative running times performances of DP with respect to other median filtering algorithms in the paper.

DP's running time was superior to the algorithms used for the experiment conducted by Marcus and Ward (2013). However, DP's strength in terms of output of denoised image was not discussed fully. The algorithm may be very fast, but may fail to deliver excellent output after the denoised task. This may be generally due to the fact that the method approximates the median value and also no indication of how it can handle images with high impulse noise density. This work therefore proposes an improvement of this approximated median filtering algorithm in order to deliver high speed noise suppressing and excellent output for real-time computer vision applications.

3. Proposed algorithms

3.1. Improved approximated median filtering algorithms (IAMFA-I & IAMFA-II)

These algorithms are designed to improve the running time as well as the output of the DP algorithm. The number of comparisons required to complete the filtering process is redesigned in order to complete faster than the DP. Two modified versions of the DP are presented in this work. The IAMFA-I improving the output generated by DP as the impulse noise density of an image increases while the IAMFA-II targets reducing the running time of the DP with equivalent outputs.

The IAMFA implements a technique called (Mid-Value-Decision-Median) that minimizes the chances of selecting a corrupted value to replace the central pixel value of a window. The technique selects the median value of a column if and only if it is considered not to be a noisy pixel. Let's assume that P stores a partition or a column extracted from a window. When P is sorted in ascending order, the arrangement can be defined as $P = \{P_1 \leq P_2 \leq P_3\}$. The value to be selected for replacement of central value of a window will depend on the current value of P2. The technique has a higher chance of selecting a non-impulse pixel for filtering as compared to the original method used in DP.

$$Mid - Value - Decision\,Median = \begin{cases} P_1\,P_2 = 255 \\ P_2\,0 < P_2 < 255 \\ P_3 P_2 = 0 \end{cases}$$

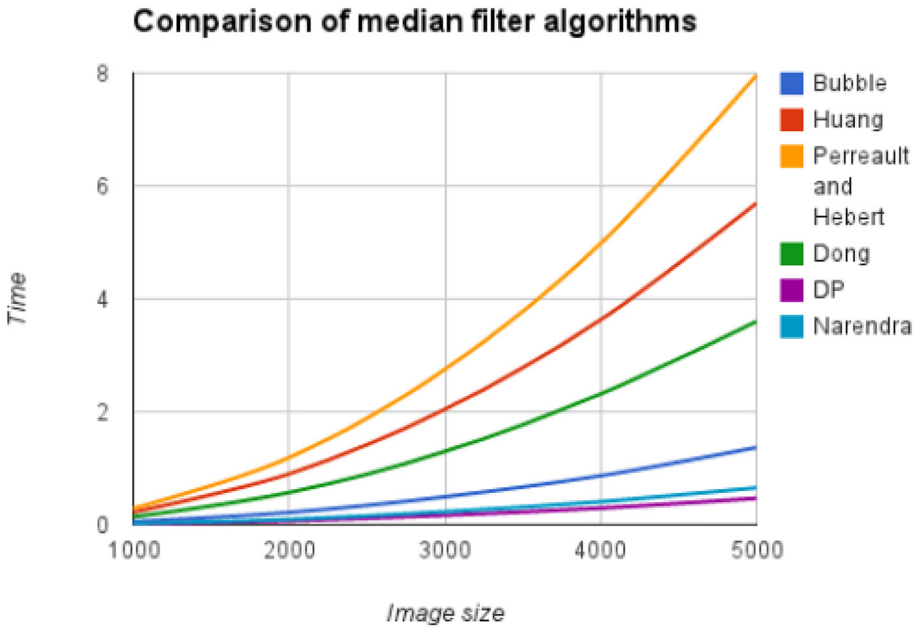

Fig. 4. Running time comparison of median filer algorithms.

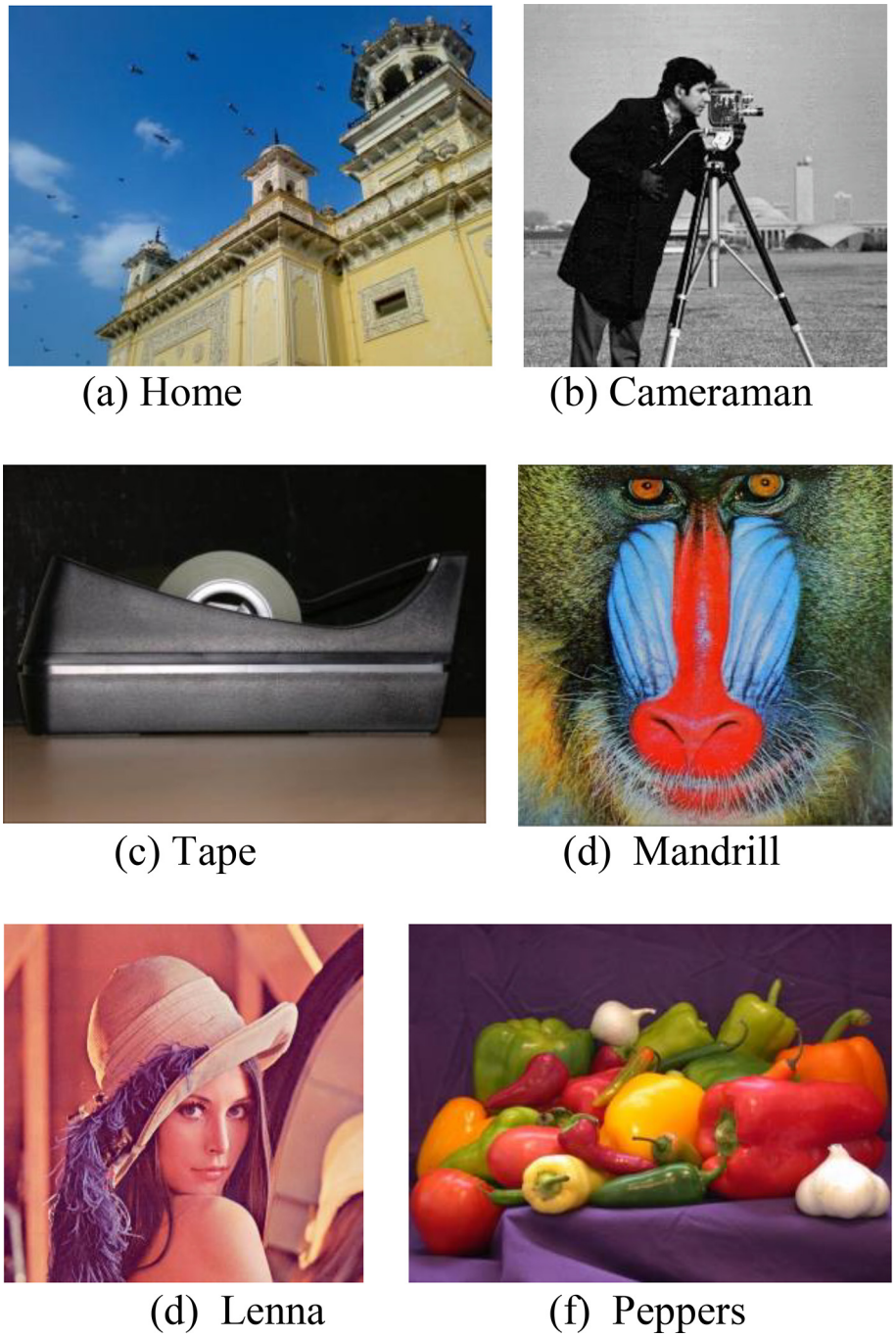

Fig. 5. Sample images used for testing algorithms.

Table 1
CPU Running times (in seconds) of Approximated Median Algorithms.

Noise Density	SMF	AMFA	DP	SOAMF	IAMFA-I	IAMFA-II
0.1	1.561	0.092	0.069	0.085	0.065	0.055
0.2	1.505	0.096	0.069	0.088	0.064	0.057
0.3	1.504	0.094	0.082	0.102	0.079	0.054
0.4	1.498	0.096	0.068	0.087	0.066	0.057
0.5	1.468	0.095	0.070	0.087	0.065	0.057
0.6	1.450	0.097	0.073	0.102	0.068	0.055
0.7	1.418	0.098	0.067	0.087	0.067	0.058
0.8	1.394	0.094	0.068	0.085	0.068	0.059
0.9	1.357	0.095	0.066	0.086	0.071	0.059
1	1.306	0.103	0.064	0.088	0.064	0.054

The Mid-Value-Decision-Median introduced by this algorithm makes it effective in handling windows that contain a lot of impulse noise. This can lower the probability of selecting impulse noise for the denoise image and thereby improving performance.

3.2. Proposed improved approximated median filtering algorithm (IAMFA-I)

The proposed IAMFA-I algorithm targets improving the output DP method generates. The algorithm is the same as the DP with the introduction of the Mid-Value-Decision Median. IAMFA-I is presented in Algorithm 2.

Algorithm 2: Proposed Improved Approximated Algorithm – IAMFA-I

Let I represent the image to be filtered
Let H be the HEIGHT of the Image
Let W be the WIDTH of the Image
for i = 2 to H − 1
*col1 ← **Mid-Value-Decision Median** (I(i-1, 1), I(i, 1), I(i + 1, 1))*
*col2 ← **Mid-Value-Decision Median** (I(i-1, 2), I(i, 2), I(i + 1, 2))*
for j = 3 to W −1
*col3 ← **Mid-Value-Decision Median** (I(i-1, j), I(i, j), I(i + 1, j))*
*I'(i, j) ← **Mid-Value-Decision Median** ([col1; col2; col3])*
col1 ← col2
col2 ← col3
end for
end for

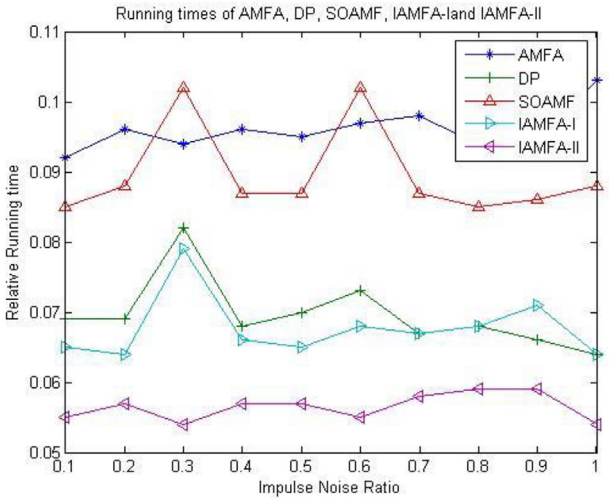

Fig. 6. Relative CPU running times of approximated algorithms.

Table 2
Average PSNR for Approximated Median Algorithms.

Noise Density	SMF	AMFA	DP	SOAMF	IAMFA-I	IAMFA-II
0.1	29.89	29.64	29.41	24.66	29.83	28.55
0.2	26.75	25.23	25.15	20.01	28.04	24.13
0.3	22.51	20.67	20.64	16.70	26.27	20.24
0.4	18.35	16.74	16.72	14.13	23.88	16.85
0.5	14.85	13.69	13.68	12.15	20.76	14.10
0.6	12.03	11.23	11.23	10.51	17.25	11.76
0.7	9.66	9.16	9.16	9.07	13.86	9.74
0.8	7.85	7.57	7.57	7.89	10.91	8.05
0.9	6.36	6.25	6.25	6.86	8.41	6.58
1	5.16	5.16	5.16	5.96	6.48	5.31

3.3. Proposed improved approximated median filtering algorithm (IAMFA-II)

The proposed IAMFA-II algorithm improves on the running time of the DP algorithm by avoiding the sorting of the content of the non-overlapping column as the window slides during the filtering process all the time. The method either uses a statistical rank order to select median value or assign the central value in the new column as the median for the final estimation of the median of medians. This alternating approach of selecting a median for the new column as the window slides helps reduces the number of comparisons that are done in order to complete a row. IAMFA is presented in Algorithm 3.

Algorithm 3: Proposed Improved Approximated Algorithm – IAMFA-II

Let Ind be an indicator for deciding whether to perform rank order or assignment.
Outer Loop slides the window from top to bottom of the image
Inner Loop slides the window from left to right of the image
Ind ← 0;
For i ← 2 to M−1
For j ← 2 to N-2
Identify a window with I(i,j) as its central pixel location
If the window is the first on the row then
*Split it into 3 columns and select the **Mid-Value-Decision Median** for each column*
Else
If Ind is equal to 1 then
*Estimate the **Mid-Value-Decision Median** of the last column of the window using rank order*
Ind ← 0;
Else
Select the middle value of the last column as its median
Ind ← 1;
End if
End if
Save Values in the Second and Third columns for the next operation
Find the median of the medians of the three (3) columns and use it to replace the central pixel
End for
End For

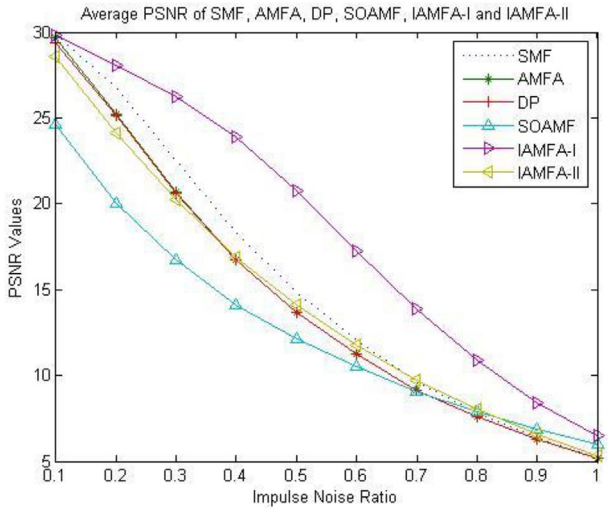

Fig. 7. Relative PSNR performance of approximated median filters.

Table 3
Average SSIM.

Noise Density	SMF	AMFA	DP	SOAMF	IAMFA-I	IAMFA-II
0.10	0.87	0.87	0.87	0.71	0.87	0.86
0.20	0.82	0.79	0.79	0.48	0.86	0.74
0.30	0.68	0.60	0.60	0.30	0.83	0.53
0.40	0.44	0.35	0.35	0.18	0.74	0.30
0.50	0.23	0.17	0.17	0.11	0.57	0.16
0.60	0.11	0.09	0.09	0.07	0.36	0.08
0.70	0.05	0.04	0.04	0.04	0.19	0.04
0.80	0.02	0.02	0.02	0.02	0.09	0.02
0.90	0.01	0.01	0.01	0.01	0.03	0.01
1.00	0.00	0.00	0.00	0.00	0.01	0.00

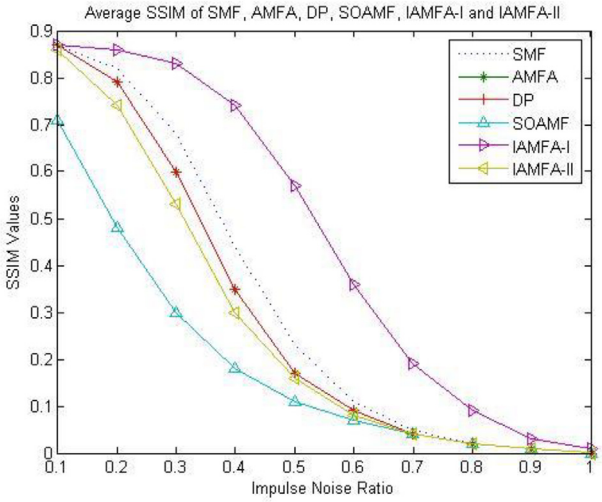

Fig. 8. Relative SSIM performance of approximated median filters.

4. Test of IAMFA-I & IAMFA-II

The Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Metric (SSIM) and CPU running time are used to test the proposed algorithms (IAMFA-I & II) against other approximated median filtering algorithms. The algorithms were implemented with Matlab and the tic/toc function used to measure CPU running time. Home, Cameraman, Mandrill, Lenna, Tape and Peppers images are used for the experiment. The images are injected with different ratios of "salt & pepper" noise and the algorithms made to filter each instance. Approximated Median Filtering Algorithm (AMFA – Directional Median filter), DP (Marcus & Ward), and SOAMF (Lu

Fig. 9. Outputs of SMF, DP, IAMFA-I and IAMFA-II when Lenna was injected with 30% impulse noise.

et al) are also tested alongside the proposed method (IAMFA-I & II). The five (5) algorithms are made to run several times on a particular computing environment and their average running times records. The SSIM and PSNR were used to measure the quality of the restored or denoised image. Fig. 5 presents the six (6) images used to evaluate the algorithms

5. Results and discussions

Table 1 presents the average CPU running times in seconds after the conduct of the experiment. For noise density of 10% on the table, the IAMFA-II recorded 0.055 as against 0.069 recorded by the DP. The other algorithms (AMFA, SOAMF and IAMFA-I) recorded values that were above the running time of 0.055. The IAMFA-II recorded the least running time as compared to the other algorithms. Similar running times were recorded for the rest of noise densities on the table. IAMFA-II recorded the least average running time because the implementation of the alternating selection of median and middle values of columns helps reduce the running time of DP.

Fig. 6 presents the relative running times of the algorithms used for the experiment. The figure was generated from Table 1. The IAMFA-II performed best among all the algorithms used for the experiment.

Table 2 presents the average PSNR values recoded after the restorations of all the six (6) images used for the experiment. The SMF, AMFA, DP, and IAMFA-I registered PSNR values of about 29.00 when images were injected with noise densities of 10%. The performances however dropped with increasing noise density.

Fig. 7 illustrates the relative PSNR values registered for the denoised images. From the results, the IAMFA-I performed best with increasing noise densities. Even though the algorithm is an approximation of the median filtering method, the PSNR values registered by this algorithm out-performed the standard median filter (SMF) algorithm.

Table 3 also presents the average SSIM values registered by the various algorithms used for the experiment. Again the SMF, AMFA, DP and IAMFA-I recorded 0.87 for noise density of 10%, and values dropped with increasing noise ratio.

Fig. 8 illustrates the relative SSIM values registered for the denoised images. The IAMFA-I again performed best with increasing noise densities. Again, IAMFA-I out-performed the standard median filter (SMF) algorithm.

Figs. 9–14 present the SMF, DP, IAMFA-I and IAMFA-II outputs for the images injected with 30% impulse noise density.

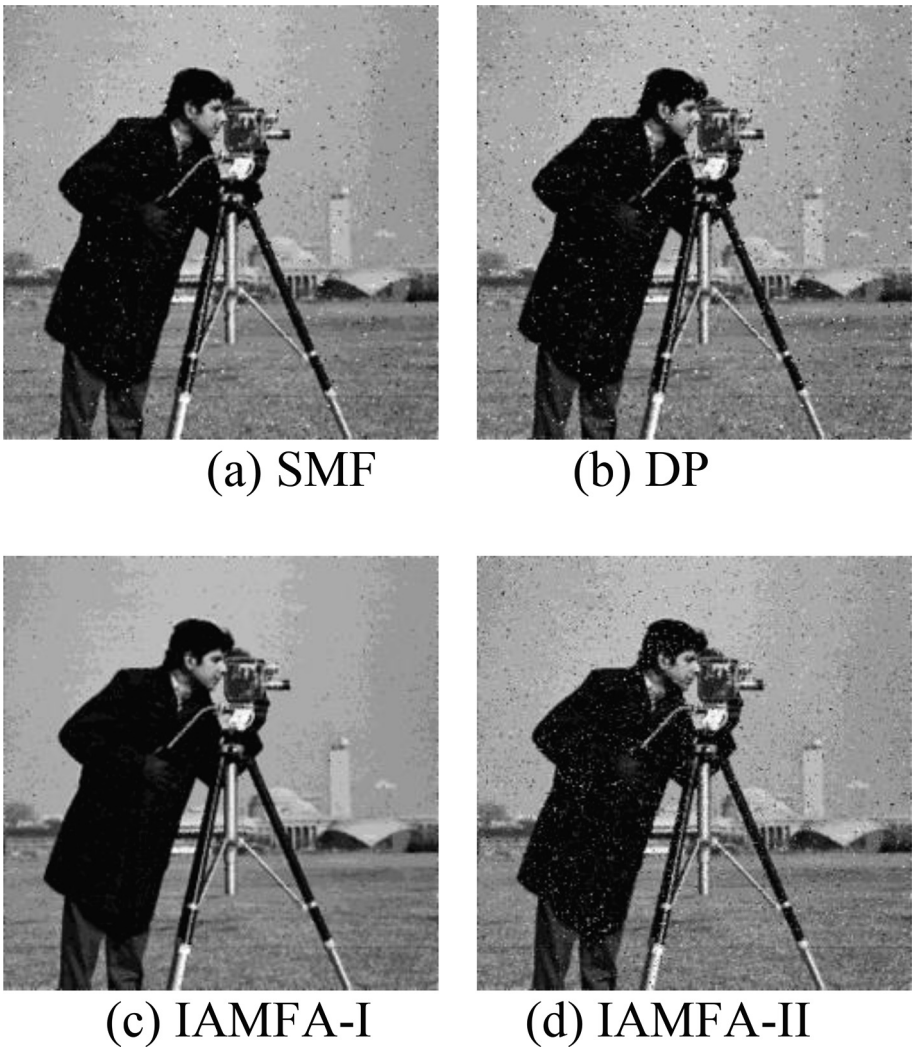

Fig. 10. Outputs of SMF, DP, IAMFA-I and IAMFA-II when Cameraman was injected with 30% impulse noise.

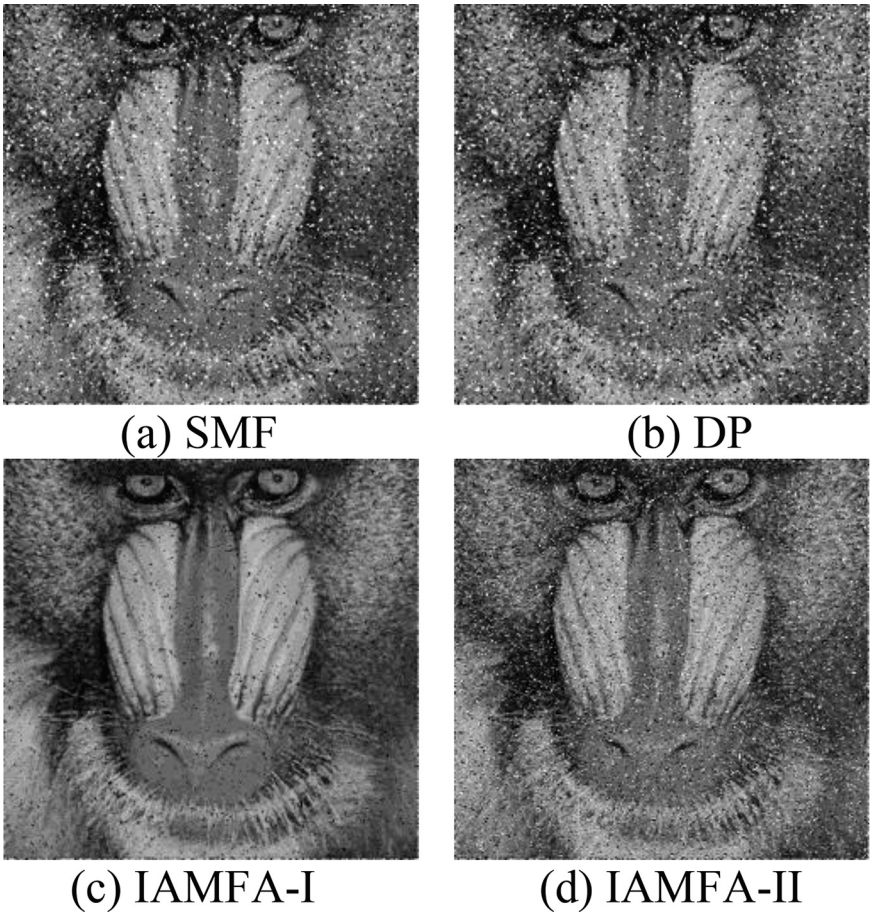

Fig. 11. Outputs of SMF, DP, IAMFA-I and IAMFA-II when Mandrill was injected with 30% impulse noise.

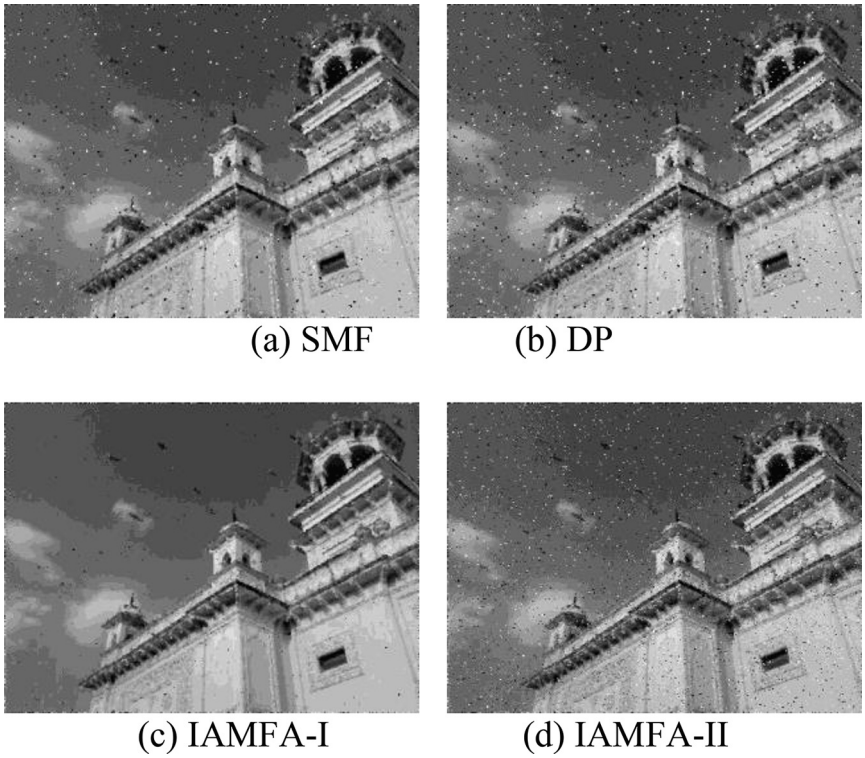

Fig. 12. Outputs of SMF, DP, IAMFA-I and IAMFA-II when Home was injected with 30% impulse noise.

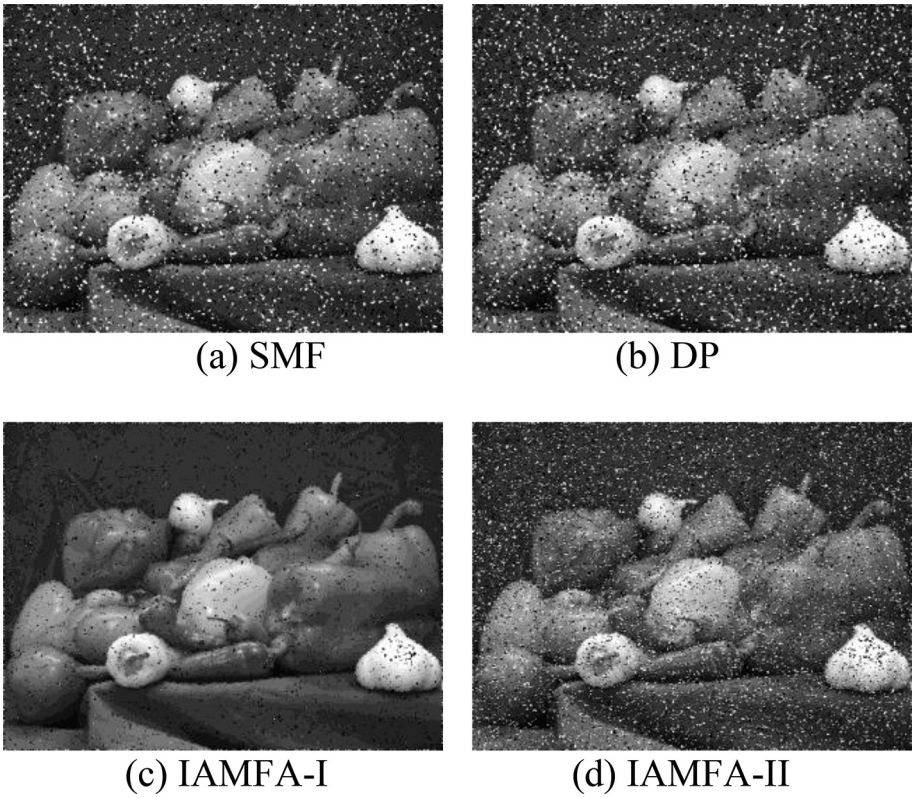

Fig. 13. Outputs of SMF, DP, IAMFA-I and IAMFA-II when Pepper was injected with 30% impulse noise.

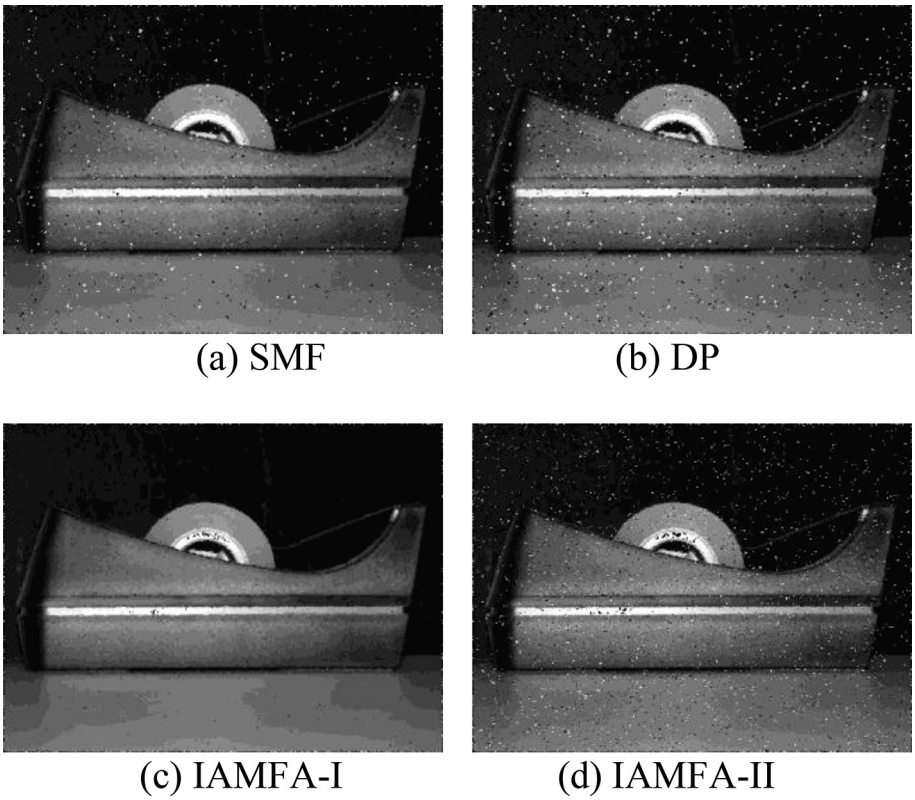

Fig. 14. Outputs of SMF, DP, IAMFA-I and IAMFA-II when Tape was injected with 30% impulse noise.

6. Conclusion

IAMFA-I and IAMFA-II algorithms are proposed in this paper to help improve the output as well as running time of DP median filter respectively. The introduction of the "mid-value-decision-median" technique for the selection of the value to replace the central pixel for the filtering task in IAMFA-I & IAMFA-II reduces the chances of selecting impulse values (0 or 255) for the denoising task. Experimental results indicate that IAMFA-I runs with equivalent running time when compared with DP, but generate better output. Again, IAMFA-II is able to generate output equivalent to that of DP, but runs with better speed. For real-time image processing and computer vision task, the proposed IAMFA-I & II can be used effectively to achieve excellent outputs when compared with the DP fast median filter as well as the five (5) other median filters compared with DP median filter in Marcus and Ward (2013).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Appiah, O., Asante, M., HayfronAcquah, J.B., 2016. Adaptive approximated median filtering algorithm for impulse noise reduction. Asian J. Math. Comput. Res. 12, (2) 134144.

Arce, G.R., Paredes, J.L., 2000. Recursive weighted median filters admitting negative weights and their optimization. G. R. Arce, J. L. Paredes, "Recursive weighted median filters admitting IEEE Transactionson Signal Processing, vol. 48, no. 3, , pp. 768–779.

Asano, A., Itoh, K., Ichioka, Y., 1991. Optimization of the weighted median filter by learning. Opt. Express 16 (3), 168–170.

Bae, J., Yoo, H., 2018. Fast Median Filtering by Use of Fast Localization of Median Value, 13(12), 10882–10885.

Chan, R.H., Ho, C., Nikolova, M., October 2005. Salt and pepper noise removal by median type noise detectors and detail preserving regularization. IEEE Trans. Image Processing 14 (10), 14791485.

Czerwinski, R.N., Jones, D.L., O'Brien, W.D., 1995. Ultrasound speckle reduction by directional median filtering. doi: 10.1109/icip.1995.529720.

Forouzan, A.R., Araabi, B.N., 2003. Iterative median filtering for restoration of images with impulsive noise. Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, 1(l), 232–235. doi: 10.1109/ICECS.2003.1302019.

Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A., 1986. One-Dimensional Estimators. In Robust Statistics: The Approach Based on Influence Functions. doi: 10.1002/9781118186435.ch2.

Huang, T.S., Yang, G.J., Tang, G.Y., 1979. A fast two dimensional median filtering algorithm. IEEE Trans. Acoustics Speech Signal Process., 27(1), 13–18. doi: 10.1109/TASSP.1979.1163188.

Huber, P.J., 1981. Robust Statistics. Wiley, New York.

Kasparis, T., Tzannes, N.S., Chen, Q., 1992. Detail preserving adaptive conditional median filters. J. Electron. Imag. 1 (14), 358–364.

Ko, J., Lee, Y.H., 1991. Center weighted median filters and their applications to image enhancement. IEEE Trans. Circuits Syst. 38 (9), 984–993.

Liu, L., Chen, C.L.P., Zhou, Y., You, X., 2015. A new weighted mean filter with a two-phase detector for removing impulse noise. Inf. Sci. Information Sci. 315 (2015), 1–16.

Lu, Y., Jiang, L., Dai, M., Li, S., 2010. Sort optimization algorithm of median filtering based on FPGA. In: 2010 International Conference on Machine Vision and HumanMachine Interface, MVHI 2010. doi: 10.1109/MVHI.2010.145.

Marcus, R., Ward, W., 2013. DP : a Fast Median Filter Approximation, 1–11.

Narendra, P.M., 1981. A separable median filter for image noise smoothing. Pattern Analysis and Machine Intelligence, IEEE Trans. PAMI-3(1):20{29, 1981.

Paeth, A.W., 1990. Median finding on a 3 × 3 Grid. Graphics Gems. https://doi.org/10.1016/b9780080507538.500449.

Perreault, S., Hebert, P., 2007. Median filtering in constant time. IEEE Trans. Image Process. 16 (9), 2389–2394. https://doi.org/10.1109/TIP.2007.902329.

Pitas, I., Venetsanopoulos, A.N., Dec. 1992. Order statistics in digital image processing. Proc. IEEE, vol. 80, no. 12, (pp. pp. 1893–1921).

Singh, S., 2011. An alternate algorithm for (3x3) median filtering of digital images. Int. J. Comput. Technol., 1(1), 7–9. doi: 10.24297/ijct.v1i1.6732.

Stewart, C.V., 1999. Robust parameter estimation in computer vision. SIAM Rev. 41 (3), 513–537. https://doi.org/10.1137/s0036144598345802.

Suomela, J., 2014. Median Filtering is Equivalent to Sorting. Retrieved from http://arxiv.org/abs/1406.1717.

Szeliski, R., 2010. Introduction, 1–25. https://doi.org/10.1007/978-1-84882-935-0_1.

Tukey, J.W., 1977. Exploratory Data Analysis. Addison.

Weiss, B., 2006. Fast median and bilateral filtering. ACM Trans. Graphics. https://doi.org/10.1145/1141911.1141918.

Zhang, Q., Xu, L., Jia, J., 2014. 100+ times faster weighted median filter (WMF). Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2830–2837. doi: 10.1109/CVPR.2014.362.

Zhu, Y., Huang, C., 2012. An improved median filtering algorithm for image noise reduction. Phys. Procedia 25, 609–616. https://doi.org/10.1016/j.phpro.2012.03.133.