International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10882-10885

Fast Median Filtering by Use of Fast Localization of Median Value

Joungeun Bae and Hoon Yoo*

Department of Computer Science, SangMyung University, Republic of Korea.
Full Professor, Department of Electronics, SangMyung University, Republic of Korea.
**Corresponding author*

Abstract

We propose a fast median filtering algorithm for signal smoothing in image processing. The median filter is the well-known method in image processing because it preserves edges effectively while reducing the impulsive noise. However, it has a problem that execution speed is slow because it uses a sorting algorithm to obtain the median value. So, fast median filtering algorithms have been discussed. In this paper, we analyze and compare several fast median filtering algorithm performances. We propose an algorithm to quickly access median value with a starting point in a histogram based on Huang's algorithm. Experiments were performed to measure and compare the performance of existing algorithms and our algorithm. The results of the experiment show the effective of the proposed algorithm.

Keywords: Median filters, random noise, image processing

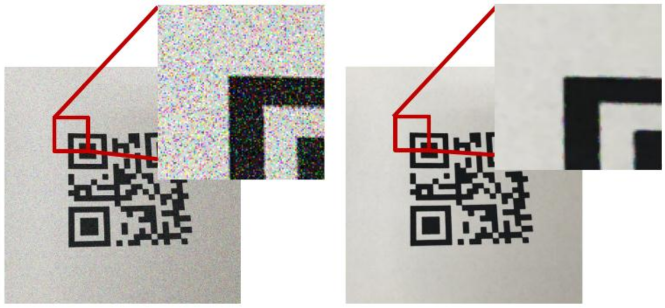
INTRODUCTION

Median filtering is a nonlinear filter used to remove salt or pepper noise of an image or signal. Since the median filter is at strength for eliminating noise while preserving edges, it is a well-known algorithm in image processing. It is used in applications requiring signal smoothing at the preprocessing step such as video edge detection and object segmentation [1-6]. Noise removal at the preprocessing is essential in order to recognize objects or acquire information in the image. For example, it is necessary to eliminate the impulsive noise in order to reduce error rate when recognizing QRcode in Fig.1.

Midian filters are excellent at removing the impulsive noise or sudden noise. Thus, these algorithms have been actively studied to be used in high-level applications. First, a weighted median filter has been proposed to increase performance by adding weight to selected pixels [7-11]. Second, the adaptive median filter reduces the noise density by adjusting the window size to handle high-intensity impulse noise [12-14]. The median filter is a very conceptually simple method using the median value of the list, but it is a nonlinear filter and has a fatal problem that the processing time is proportional to the filter size. Thus, an algorithm that quickly executes the sorting algorithm to obtain the median value is necessary.

Therefore, a discussion on the fast version of the median filter to overcome above mentioned shortcomings has been continuously published [15-17]. The median is the value in the middle of the sorted list of numbers. In a brute-force way, the time complexity per pixel is $O(n^2 \log n)$ when applying a quick sort to a median filter with window size of n. This algorithm is unworkable because the execution time increases very steeply according to the window size. On the other hand, the histogram can be used without the sorting algorithm to obtain the median value. The histogram is made up of 256 bins for 8-bit images, and the frequency is accumulated to obtain the median position. In this case, the time complexity is $O(n^2 + k)$, where k is the maximum value of the input value. A fast median filtering algorithm of $O(n)$ time complexity based on histogram has been proposed [15]. This algorithm is very simple and compact, so it is a hardware friendly method. Based on this algorithm, $O(\log n)$ and $O(1)$ algorithms has been proposed [16,17]. However, this method has a disadvantage that it requires a lot of memory and is complicated to implement in hardware.

In this paper, we propose an enhanced algorithm based on the Huang's algorithm. This algorithm not only updates the histogram like the existing method, but also obtains the starting point in advance and finds out the median value of the histogram quickly using it. This method is effective when the window size of the median filter is not big. Small size of our median filtering algorithm is necessary because smaller window size can better preserve the details such as corner and fine lines, which cause image blur. Therefore, our proposed algorithm is useful and necessary for implementing at high-level applications. In order to evaluate the performance of the median filtering algorithms, we measured the execution time of each median filter. Experimental results show that the proposed method performs faster than the Huang's algorithm.



(a) (b)

Figure. 1. Effect of median filtering. (a) Image before filtering, (b) Image after 3×3 median filtering.

FAST MEDIAN FILTERING ALGORITHM

Huang proposed a fast median algorithm using histogram, which is a time complexity of $O(n)$ [15]. This algorithm reuses the histogram of the filter used in the previous pixel in the next filter as shown in Fig 2. According to Algorithm 1, n

subtraction and n addition are needed to update the histogram per pixel. It has an advantage that it can be easily implemented in the hardware stage since this algorithm is very simple and compact. Advanced algorithm based on this algorithm has been discussed using hierarchy of histogram which is $O(\log n)$ [16]. This approach has been enhanced in terms of reducing execution time compared the existing method, but it is very complicated to apply in practical.

A constant median filtering algorithm based on Huang 's algorithm has been addressed [17]. There is a process to update the histogram of the existing algorithm, but it is executed through two steps as shown as Fig 3. According to algorithm 2, two histograms are upgraded: one is a column histogram and the other is an entire histogram. First, execute at in the rightmost column histogram: subtract the pixels of the existing row and add the pixels of the new row. Second, execute at the entire histogram: subtract the leftmost column histogram and add the newly updated column histogram. This process requires one addition, one subtraction in the first step and 128 subtractions and 128 additions in the second step. This paper proposes $O(1)$ time complexity. In addition, vectorization, cache friendliness and multilevel histograms are applied to improve efficiency. Multilevel histograms update the histogram by dividing by coarse (24bins) and fine (segment corresponding to each coarse) level in order to find the median value faster.

Algorithm 1 Huang's O(n) median filtering algorithm [15]

Input: Image X of size M×N, filter size n
Output: Image Y of the same size as X
  Initialize histogram H
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
      for k = $-n/2$ to $n/2$ **do**
        Remove $X_{i+k,\, j-n/2-1}$ from H
        Add $X_{i+k,\, j+n/2}$ to H
      end for
      $Y_{i,j} \leftarrow \text{median}(H)$
    end for
  end for

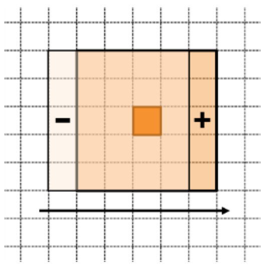


Figure. 2. Histogram update in Huang's algorithm (In this figure, n=5)

Improved algorithms based on Huang's algorithm operate at fast execution speeds [16-17]. However, these algorithms are memory-intensive when initializing several histograms initially. In addition, the hardware implementation is not feasible because it is too complicated to update multiple histograms and to locate the median in the histogram at multiple levels in hardware.

PROPOSED ALGORITHM

In this paper, we propose an algorithm to find median value by applying starting point to histogram based on Huang's algorithm. The focus of the median filtering algorithm is on how quickly the location of the median values is found in the histogram. The proposed algorithm is the same up to Huang 's histogram update method, but it uses the starting point that was previously obtained to find the median value more quickly. According to Algorithm 3, when the histogram is updated, at the same time, the position at which the value of the histogram starts is also updated, which is called the starting point. The median value is searched from the prepared starting point position unlike the existing algorithm of finding the median value sequentially from 0 as shown in Fig 4. (k is the starting point since it is the first position with a positive integer in the histogram of Fig 4)

If the filter size is not big, finding the median value position in the histogram using the starting point is faster than the Huang's algorithm. When the median value is found from the histogram in the Huang's algorithm, the worst-case case occurs with 255 additions (8bit image). At this time, using the starting point can reduce the number of additions. The execution time of $O(n^2)$ is added when updating the starting point, but it is better than Huang's algorithm when the window size is less than 15.

Algorithm 2 O(1) median filtering algorithm [17]

Input: Image X of size M×N, filter size n
Output: Image Y of the same size as X
  Initialize histogram H and column histograms $h_{1\dots n}$
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
        Remove $X_{i-n/2-1,\, j+n/2}$ from $h_{j+n/2}$
        Add $X_{i+n/2,\, j+n/2}$ from $h_{j+n/2}$
        $H \leftarrow H + h_{j+n/2} - h_{j-n/2-1}$
        $Y_{i,j} \leftarrow \text{median}(H)$
    end for
  end for

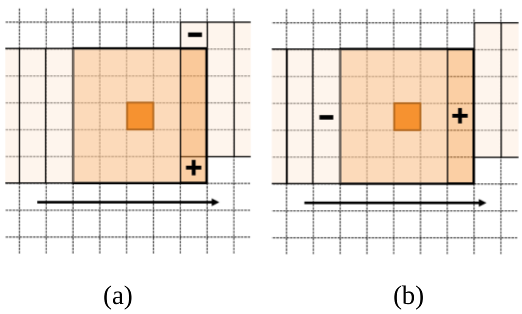


Figure. 3. Histograms update of the constant algorithm. (a) Update processing of the column histogram h_j. (b) Update processing of the kernel histogram H.

Of course, the purpose of the fast median algorithm with a large filter size is to easily remove noise even in high-resolution images. However, there is a critical problem of eliminating a small structure, which is a part of the image, when it is applied in applications. This is a huge problem in areas such as image segment and image recognition where median filters are used. Therefore, it is correct to apply the median filter of a small size. At this time, the proposed algorithm has a merit that it is sufficiently fast, useful and simple.

EXPERIMENTAL RESULTS

To evaluate the performance of the fast median filtering algorithm, their processing times were compared in this section. The input image is an image including QR code, the impulsive noise exists and the size is 1024 × 1024 as illustrated in Fig 5(a). This experiment was performed on a computer with a CPU performance of 3.30 GHz. The run time measurement of the algorithm was averaged over ten iterative experiments. The number of algorithms applied to the experiment is 4 and these are as follows: Quick sort algorithm, simple histogram algorithm, Huang's algorithm [15], and proposed algorithm.

The graph for performance comparison of the runtime of the four algorithms is depicted in Fig 6. First, the algorithm using the quick sort, a brute-forth approach method, shows an unworkable measure, as evidenced by $O(n^2 \log n)$. The algorithm with simple histogram which is $O(n^2+k)$ also shows the result that is not useful. Next, the result of the Huang's algorithm is shown as $O(n)$. Finally, with regard to the performance of the proposed algorithm, Huang's algorithm with starting point, is improved over that algorithm.

Algorithm 3 The proposed median filtering algorithm.

```
Input: Image X of size M×N, filter size n
Output: Image Y of the same size as X
  Initialize histogram H and starting point k
  for i = 1  to M do
    for j = 1 to N do
      for k = -n/2 to n/2 do
         Remove X_{i+k, j-n/2-1} from H
         Add X_{i+k, j+n/2} to H
         Update k
      end for
         Y_{i,j} ← median(H, k)
    end for
  end for
```

	0	0	0	0	....	2	2	1	0	8	1	....
idx =	0	1	2	3	....	k	k+1	k+2	k+3	k+4	k+5	....

↑
starting point

Figure. 4. Find the median at the histogram with starting point

When the median filtering algorithm is applied to the image, the small structure of image is lost while noise disappears as shown in Fig. 5(b-d). This indicates that this method can not be used in high-level applications because the larger the filter size, the worse the loss of a portion of the QR code. Fast median filtering algorithm has been developed in order to apply a large size of the median filter to a high resolution image. However, the large size of the median filter is not workable because of the above reason. Therefore, the filter used becomes a small-sized filter. The algorithm proposed in this paper is faster than the Huang's algorithm when the filter size is less than 15. Thus, the proposed algorithm improves the performance by using fast location median value.

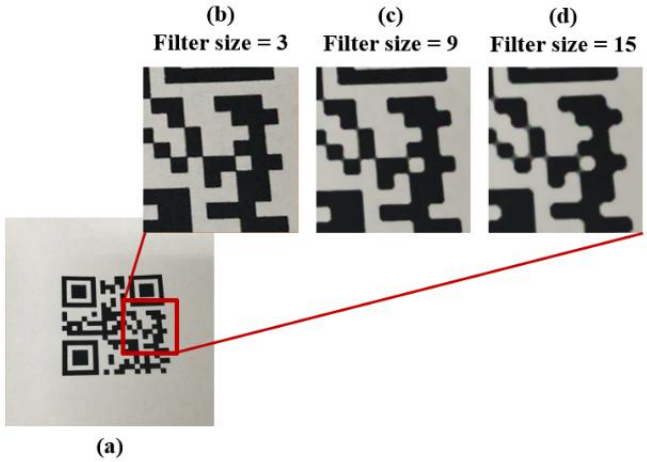


Figure. 5. Effect of median filtering. (a) Original RGB image, (b) The part of the image after applying the 3×3 median filtering (c) The part of the image after applying the 9×9 median filtering (d) The part of the image after applying the 15×15 median filtering.

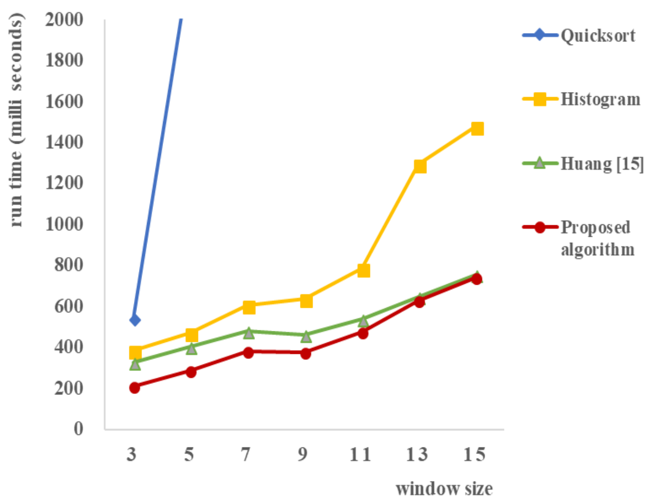


Figure. 6. Timing of median filtering algorithms and the proposed algorithm.

CONCLUSION

In this paper, we analyzed the fast median filtering algorithms used to remove pepper and salt noise in image. We calculated and compared the time complexity of existing median filtering algorithms for analysis. An algorithm was proposed to find the position of the median value more quickly by applying starting point to a very simple Huang's algorithm. Comparison of runtime by filter size was executed in experiments in order to compare the performance of the fast median filtering algorithms. The experimental results show that the proposed algorithm is more effective than Huang 's algorithm when the window size of the filter is less than 15.

REFERENCES

[1] Y. Zhang, S. Li, and S. Wang, "Revealing the traces of median filtering using high-order local ternary patterns." *IEEE Signal Processing Letters*, vol.21, no. 3, pp.275-279, Mar. 2014.

[2] K. M. Sundaram, D. Sasikala, and P. A. Rani, "A study on preprocessing a mammogram image using Adaptive Median Filter." *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 3, pp.10333-10337, Mar. 2014.

[3] G. M. Rao, and Ch. Satyanarayana, "Object Tracking System Using Approximate Median Filter, Kalman Filter and Dynamic Template Matching." *International Journal of Intelligent Systems and Applications, vol.* 6, no. 5, pp.83-89, Apr. 2014.

[4] V. Gupta, V. Chaurasia, and M. Shandilya, "Random-valued impulse noise removal using adaptive dual threshold median filter." Elsevier *Journal of visual communication and image representation, vol.*26, pp. 296-304, Oct. 2014.

[5] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks." *IEEE Signal Processing Letters, vol.* 22, no. 11, pp.1849-1853, Nov. 2015.

[6] S. Gan, S. Wang, Y. Chen, X. Chen, and K. Wiang, "Separation of simultaneous sources using a structural-oriented median filter in the flattened dimension." Elsevier Journal of *Computers & Geosciences, vol.* 86, pp: 46-54m Jan. 2016.

[7] G. T. Arce, and J. L. Paredes, "Recursive weighted median filters admitting negative weights and their optimization." *IEEE Transactions on Signal processing, vol.* 48, no. 3, pp. 768-779, Mar. 2000.

[8] S. M. Rahman, and M. K. Hasan, "Wavelet-domain iterative center weighted median filter for image denoising." *Signal Processing*, vol. 83, no. 5, pp. 1001-1012, 2003.

[9] C. Y. Wang, L. L. Li, f. P. Yang, and H. Gong, "A new kind of adaptive weighted median filter algorithm." *Computer Application and System Modeling (ICCASM), 2010 International Conference on IEEE*, Vol. 11, pp.667-671, Oct. 2010.

[10] Z. Ma, K. He, Y. Wei, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond." *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, p.49-56, 2013.

[11] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)." *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, pp. 2830-2837, 2014.

[12] H. Hwang, and R. A. Haddad, "Adaptive median filters: new algorithms and results." *IEEE Transactions on image processing*, vol. 4, no. 4, pp. 499-502, Apr. 1995.

[13] S. Q. Yuan, and Y. H. Tan, "The solutions of equation-based noise detector for an adaptive median filter." Elsevier journal of *Pattern recognition,* vol. 39, no. 11, pp. 2252-2257, Nov. 2006.

[14] S. Shrestha, "Image denoising using new adaptive based median filters." arXiv preprint arXiv:1410.2175, 2014.

[15] T. Huang, G. J. T. G. Y. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13-18, Feb. 1979.

[16] B. Weiss, "Fast median and bilateral filtering." *ACM Transactions on Graphics (TOG)*, vol. 25, No. 3, pp.519-526, Jul. 2006.

[17] S. Perreault, and H. Patrick, "Median filtering in constant time." *IEEE transactions on image processing*, vol. 16, no. 9, pp.2389-2394, Aug. 2007.