An Edge-Based Adaptive Method for Removing High-Density Impulsive Noise from an Image While Preserving Edges

Dong-Ho Lee

This paper presents an algorithm for removing high-density impulsive noise that generates some serious distortions in edge regions of an image. Although many works have been presented to reduce edge distortions, these existing methods cannot sufficiently restore distorted edges in images with large amounts of impulsive noise. To solve this problem, this paper proposes a method using connected lines extracted from a binarized image, which segments an image into uniform and edge regions. For uniform regions, the existing simple adaptive median filter is applied to remove impulsive noise, and, for edge regions, a prediction filter and a line-weighted median filter using the connected lines are proposed. Simulation results show that the proposed method provides much better performance in restoring distorted edges than existing methods provide. When noise content is more than 20 percent, existing algorithms result in severe edge distortions, while the proposed algorithm can reconstruct edge regions similar to those of the original image.

Keywords: Impulsive noise, edge region, uniform region, binarized image, median filter.

Manuscript received June 22, 2011; revised Jan. 8, 2012; accepted Jan. 25, 2012.

This work was supported by the research fund of College of Engineering Science at Hanyang University (HY-2011-N).

Dong-Ho Lee (phone: +82 31 400 5216, dhlee77@hanyang.ac.kr) is with the Department of Electrical and Communication Engineering, Hanyang University ERICA Campus, Ansan, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0111.0392

I. Introduction

As image processing technologies have improved, images have become important for storing and expressing information in modern society. Removing noise in an image is important for improving image quality. Usually, noise originates from taking pictures through a defective sensor or transmitting images through a noisy channel. Such noise can be categorized into many different types by a probability distribution, such types being impulsive noise, Gaussian noise, Rayleigh noise, and Laplacian noise, to name a few [1]-[3]. Because impulsive noise seriously deteriorates picture quality and disrupts image processing, an efficient noise-removal algorithm is a core image processing technology. Impulsive noise is also known as salt-and-pepper noise because of the random dispersion of black and white pixels with abrupt distortions throughout an entire image.

A popular method for removing impulsive noise is a median filter, which is widely used due to its high performance and implementation simplicity [4]. The performance of a median filter depends on the size and shape of a mask. If an image contains very high-density impulsive noise, edge distortions are unavoidable with median filters of any size or shape. The following studies have been conducted to solve these problems.

Filters that adaptively determine the appropriate mask size according to the quantity of noise, including the simple adaptive median (SAM) filter [5] and the efficient decision-based (EDB) algorithm [6], have been proposed. Since these methods merely apply a filter according to noise quantity without considering the edge region, they produce serious distortions in edge regions corrupted by large amounts of noise. Yet, methods that consider edge regions, such as the directional-weighted median (DWM) filter [7] and the adaptive center-weighted median (ACWM) filter [8], also have limitations. Since the DWM filter performs poorly with respect to removing noise, it must be repeatedly applied, and since it also calculates correlations using a small 5×5 area, it is apt to generate some mistakes or errors in determining correlations. The ACWM filter also has performance problems. Since the edge region in a binarized image has already been corrupted by noise, ACWM makes some errors in determining the exact weights. The ACWM filter has a serious shortcoming in that it does not remove noise well from images corrupted by large amounts of noise.

This paper proposes an adaptive method that can remove noise without edge distortions whether or not the edge region has any slope. After segmenting an image into uniform and edge regions and compensating for edge distortions with simple binary image processing, the edge region is represented using connected lines that contain slope and directional information. The noise in uniform regions is then removed using a SAM filter adjusted to the amount of noise present. Next, noise in edge regions is removed without edge distortions by using a new prediction filter along with a line-weighted median filter that uses the detected slope and directional information of the connected lines.

Section II of this paper briefly summarizes the existing algorithms, and section III describes the proposed algorithm. Section IV provides some simulation results, and, finally, section V concludes the paper.

II. Existing Algorithms

Median filters, in general, show good preservation of edges in situations of small amounts of noise when compared to the effects of applying other commonly used filters. However, if an image has large amounts of noise, the median filter is more likely to fail to restore noisy pixels on object boundaries compared to uniform regions, resulting in small random shifts of the object boundary. To solve this problem, the SAM filter [5] has been proposed to remove noise by adaptively adjusting mask size based on the quantity of noise. The SAM filter provides superior performance both for removing noise and also for preserving edges of an image with small amounts of noise. However, since it does not consider edge regions, the SAM filter generates some serious distortions in edge regions that have large amounts of noise.

To improve noise removal for filters with a small mask size, the EDB algorithm [6] has been proposed. The EDB algorithm improves noise removal by using directional information to orient surrounding pixels. In contrast to a median filter, which sorts pixels by quantity and chooses the median value in a mask, the EDB algorithm removes noise by selecting the central value after sorting pixels in the vertical, horizontal, and diagonal directions. Since the mask used in the EDB algorithm is relatively small, it cannot remove noise from images corrupted with large amounts of noise. If it does not remove noise using the filter, the EDB algorithm replaces a noisy pixel with a pixel from the surrounding area. In general, the EDB algorithm performs better at removing noise and preserving edge region than a conventional median filter with the same mask size. However, when there is a large amount of noise, it generates some distortions, especially in edge regions, because it adopts a pixel in the surrounding area as the output.

A common problem of most conventional filters is that they produce distortions in edge regions. One filter proposed to avoid edge distortion is the DWM filter [7]. The structure of the DWM filter is divided into two functions: noise detection and filter application. The DWM filter detects impulsive noise using the correlation between the central pixel in a 5×5 mask and other pixels in the vertical, horizontal, and diagonal directions. After detecting noise and choosing one direction from the four possible directions, the DWM filter assigns weights to pixels located in the chosen direction of the 5×5 mask and then applies a median filter. Although the DWM filter provides superior performance in recovering the distorted edge region, its shortcoming is that it does not remove noise well. Consequently, the filter is generally applied several times in succession. However, it still may not get correct directional information from the small 5×5 mask size. In addition, the number of directions considered is too small. Because it considers just four directions, the DWM filter cannot detect all directions with low slope.

The ACWM filter [8], which is adaptively applied to edge regions in a binarized image, has been proposed to improve the recovery of edge regions. The binarized image is generated by first applying a Sobel operator to a noisy image and then comparing its output with a threshold value. Since this binarized image is reconstructed from an image corrupted by noise, it can make mistakes by treating some areas around impulsive noise as edge regions. To solve this problem, the ACWM filter detects noise by comparing a value calculated using simple templates with a threshold value. After locating noise positions in a binarized image, the ACWM filter also tries to improve performance by excluding false edge regions adjacent to noise, using simple templates. Then, the ACWM filter improves edge conservation and noise removal by selectively assigning weights based on the center of the binarized edge map. However, it also has a serious problem: when there is a large amount of impulsive noise, it performs poorly due to false error detection.

III. Proposed Algorithm

The algorithm proposed in this paper is shown in its entirety in Fig. 1. After detecting impulsive noise pixels and applying the conventional SAM filter to remove noise, the proposed algorithm generates a binarized image, which describes connected edge regions based on connected lines. Then, an efficient filter using the connected lines in the edge regions, which can remove noise and minimize distortions in a noisy edge region, is proposed.

1. Noise Detection

Noise must be accurately detected before an efficient filter can be designed. To increase the likelihood of detecting impulsive noise while also reducing false positives that recognize an actual pixel as noise, this paper adopts a hybrid method for detecting impulsive noise, using both median and average values. First, the absolute difference between the center pixel value and the median of the 5×5 mask is compared to a threshold value. Then, the absolute difference between the center value and the average of a 3×3 mask is compared to a second threshold. Finally, a pixel exceeding both of these thresholds is considered noisy.

From simulations, we find that the proposed hybrid method provides better performance in detecting impulsive noise than the method using only a median or mean value. Threshold values for the differences between the center pixel and the median and average value are assigned as 25 and 13 from the range of 0 to 255, respectively, based on many experimental simulations. To decide the threshold values, Lena and Pepper images with added impulsive noise of 5%, 10%, 15%, 20%, 25%, and 30% are used for evaluation. After applying all values from 10 to 100 as threshold values and measuring the performance for detecting noise, it is clear that the threshold values provide the highest detection rate.

2. Image Binarization

If image binarization is performed on an image corrupted by large amounts of impulsive noise, significant loss of edge information occurs. In this paper, noise is detected using the method previously explained, and noise is removed by the conventional SAM filter. Then, the Sobel operator, which extracts a binary image segmented into edge regions and uniform regions using a relatively small threshold value, is applied to an image. However, the binarized image extracted from the image that had noise removed using the SAM filter still includes edge distortions generated by severe impulsive noise. Such distortions can be largely corrected by applying simple binary processing. The following adjustments are applied. First, small isolated edge regions in a binarized image, which are just four-connected, are considered to be noise and are detected and excluded from the edge regions. Next, distorted edge regions are smoothed by correcting some binary patterns; six non-smoothing patterns and their corrected results are shown in Fig. 2. If these patterns are detected in a binarized image, since those patterns are not smooth, the patterns are replaced by their corrected patterns. This correction processing is repeatedly applied to the other three different directions by rotating masks by 90º. Finally, the resulting binary image is segmented into smooth connected edge and uniform regions.

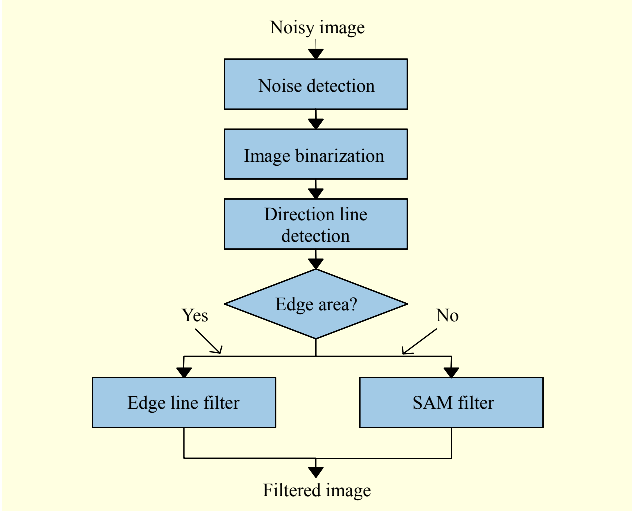

Fig. 1. Flowchart of proposed algorithm.

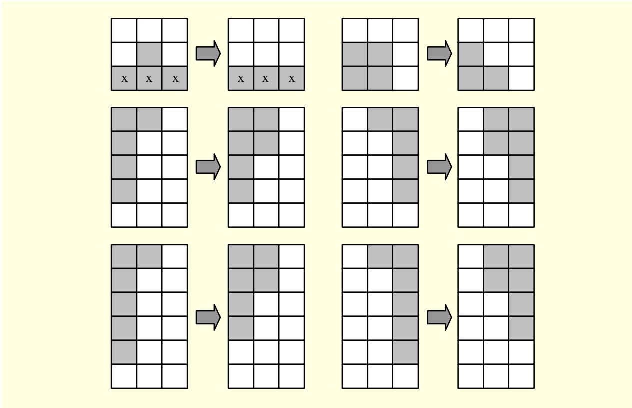

Fig. 2. Non-smoothing binary patterns and their corrected results.

3. Directional Line Detection

To determine specific directions after correcting distorted edge regions, the edge regions in a binarized image are

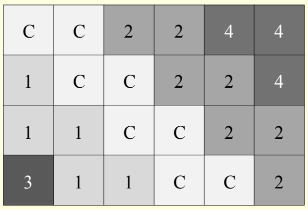

Fig. 3. Example showing some pixels in same edge region. Pixels labeled "C" represent centerline and labels "1," "2," "3," and "4" represent separate lines.

expressed with the connected lines that include directional information. These line descriptions of edge regions are very important in this paper, since they are used as basic information by a prediction filter to restore the edges distorted by impulsive noise and are used to determine the weight values of pixels for a line-weighted median filter.

The connected lines in the edge regions are extracted as follows.

First, the centerline for each edge region, which is similar to a skeleton, is extracted by the thinning process that reduces the edge width to 1- or 2-pixel thickness starting from the outer boundary. This edge reduction process is first applied to edge regions in the vertical direction. The same edge reduction process is then applied to edge regions in the horizontal direction. After the centerline is found, the connected lines in the edge regions are also described by simply expanding to the entire edge region. Each connected line in the edge region is determined by expanding to the neighboring eight-connected pixels from each pixel of a prior line. An example showing an edge region with some connected lines is shown in Fig. 3. In Fig. 3, the label "C" represents a centerline in the edge region; other neighboring lines are labeled by different numbers. Here, the numbers labeled in the figure are just used for representing the separate connected lines in the same edge region.

4. Edge Line Filter

The connected lines in the edge regions contain the directional information that covers any slope and boundary shape. This directional information of the connected lines can be efficiently used for the prediction filter and the line-weighted median filter proposed in this paper to restore the distorted edges.

A. Prediction Filter

When there is a large amount of impulsive noise in edge regions, the filter mask should be large enough to use some pixels located far away from the pixel to apply the filter. Yet, in that case, undesirable results that distort edge regions are likely.

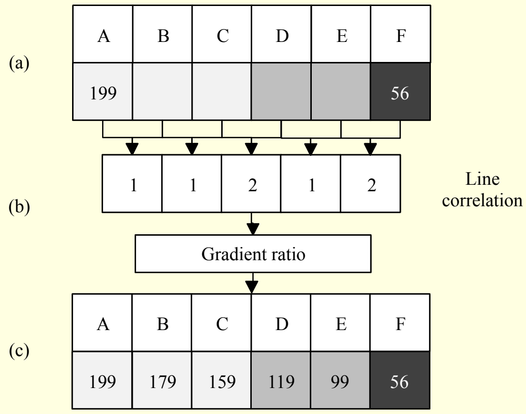

Fig. 4. Example of horizontal prediction filter process: (a) reference and noisy pixels, (b) line correlation and gradient ratio, and (c) predicted results.

This paper proposes a new method that can predict exact pixel values, using the connected lines in the edge regions even when a large amount of noise is distributed in the edge regions. This prediction filter is processed in the following order.

1) First, decide whether to predict horizontally or vertically according to the distribution of the edge region. Using two masks, which are 10×5 and 5×10, the edge region direction is determined by calculating and comparing the degree of distribution of the edge region in each mask. For the case of noisy pixels in the edge region, measure the total number of noisy pixels that are horizontally or vertically connected to one another according to the edge region direction.

2) Decide line correlations for noisy pixels by assigning 1 if two adjacent pixels belong to the same connected line and 2 if two adjacent pixels belong to different lines.

3) Calculate a gradient ratio using (1) to decide gradual change in the edge region. Reference pixels in (1) are non-noisy pixels that are located on both ends of connected noisy pixels. The gradient ratio represents a minimal increase or decrease of the pixel value from the prior pixel:

$$\text{Gradient ratio} = \frac{\text{difference of two reference pixels}}{\text{sum of line correlations}}. \quad (1)$$

4) Finally, predict the values of noisy pixels by adding or subtracting the value obtained by multiplying the gradient ratio found in step 3) by the cumulative value of the line correlations found in step 2).

Reflecting the process described above, Fig. 4 presents an example in which the pixels labeled "B" to "E" are noisy pixels and the non-noisy pixel values at the ends are 199 and 56. Pixels located on both ends of the connected noisy pixels are used as reference pixels for calculating the gradient ratio. In the example, the line information for four connected noisy pixels is used to decide the line correlation. In the figure, the pixels labeled "A," "B," and "C" are in the same connected line, and pixels "D" and "E" are in another connected line. The line correlation is set to 1 for pixels in the same connected line and 2 for pixels adjacent to the different connected line. Figure 4(b) shows the line correlation results. Next, the gradient ratio between two reference pixels is calculated. The gradient ratio is calculated as in (1), by dividing the difference between two reference pixels by the total sum of the line correlations. Applying (1) to our example yields a gradient ratio of 20. Using the line correlation and the gradient ratio obtained in the manner described above, the exact values of noisy pixels can be predicted as shown in Fig. 4(c).

Connected noisy pixels are distributed in the horizontal direction in the example shown in Fig. 4. An analogous prediction holds when connected noisy pixels are distributed vertically. Since the suggested prediction filter is applied to the horizontal and vertical directions according to the edge region direction, it has some limitations in predicting pixels if an edge region has a low slope, such as 45º.

B. Line-Weighted Median Filter

The prediction filter is able to forecast a value for a noisy pixel while minimizing edge distortions. However, as mentioned in III.4.A., because the prediction filter can be applied in just two directions, horizontal and vertical, it can make some errors if an edge region has a low slope, such as 45º. Thus, after applying the prediction filter, the proposed line-weighted median filter is applied to all noise pixels in the edge region to enhance edge boundary and address such prediction filter errors. Results of the prediction filter are used as reference pixels for a line-weighted median filter. A 5×5 mask is applied to a noisy pixel and different weights are assigned to some pixels in a 5×5 mask according to the connected line information for the edge region.

C	C	2	2	4
1	C	C	2	2
1	1	C	C	2
3	1	1	C	C
3	3	1	1	C

(a)

1	1	0	0	0
0	3	3	1	0
0	1	5	3	0
0	1	1	3	1
0	0	0	0	1

(b)

Fig. 5. Example of line-weighted median filter: (a) some connected lines in same edge region and (b) assigned weight values.

Figure 5(a) shows an example that includes several different connected lines described by different labels. Figure 5(b) shows weights assigned to pixels in the mask referencing the connected lines in Fig. 5(a). First, for applying a line-weighted median filter, weights are assigned. Weight 5 is assigned to the central noisy pixel in a mask. Weight 3 is assigned to pixels in a 3×3 mask located on the same line as the central pixel. Weight 1 is assigned both to pixels in a 3×3 mask located on the other lines and to pixels outside of a 3×3 mask located on the same line as the central pixel. The remaining pixels are assigned weight 0, meaning that they are excluded from the reference pixels. A median filter using these weights is then applied. Using this line-weighted median filter supplements the prediction filter's errors and more clearly expresses the edge regions.

IV. Simulation Results

To evaluate the performance of the proposed algorithm, we compare its results to those of several existing methods. We use both objective and subjective evaluations and begin with images containing various amounts of impulsive noise. Figure 6 shows the binarized images obtained after applying the SAM filter, the Sobel operator, thresholding, and correction

Fig. 6. Binarized images after edge correction: (a) original image, (b) 5% noise, (c) 10% noise, (d) 15% noise, (e) 20% noise, (f) 25% noise, and (g) 30% noise.

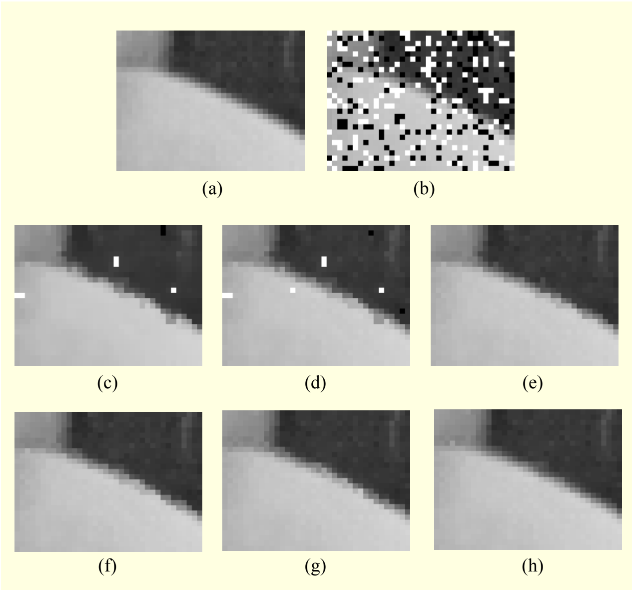

Fig. 7. Simulation outputs of Lena image with 30% impulsive noise added: (a) original image, (b) noisy image, (c) 3×3 median filter, (d) ACWM [8], (e) EDB [6], (f) DWM [7], (g) SAM [5], and (h) proposed algorithm.

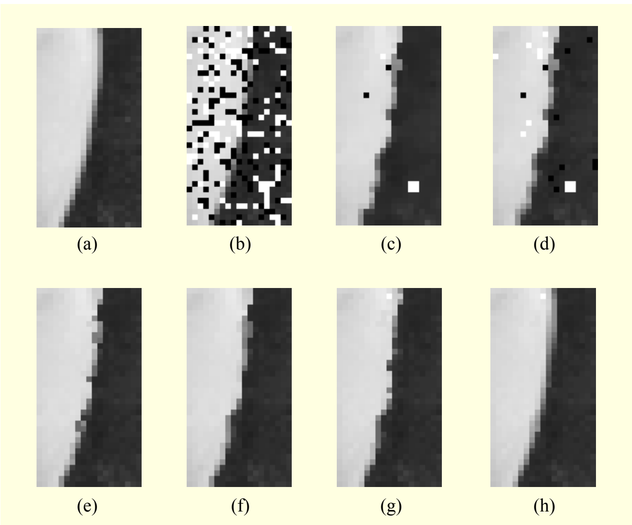

Fig. 8. Simulation outputs with Lena image with 30% impulsive noise added: (a) original image, (b) noisy image, (c) 3×3 median filter, (d) ACWM [8], (e) EDB [6], (f) DWM [7], (g) SAM [5], and (h) proposed algorithm.

work to the edge regions of the Lena images with added impulsive noise of 5%, 10%, 15%, 20%, 25%, and 30%. From the figures, it is evident that, regardless of noise quantity, these binarized images do not have serious edge distortions.

In the subjective performance evaluation, the proposed algorithm is compared with simulation results from the 3×3 median filter, EDB algorithm, SAM algorithm, DWM filter, and ACWM filter. The Lena and Pepper images are used as

Fig. 9. Simulation outputs with Pepper image with 30% impulsive noise added: (a) original image, (b) noisy image, (c) ACWM [8], (d) DWM [7], (e) SAM [5], and (f) proposed algorithm.

test pictures, and output images from conventional algorithms and the proposed algorithm are magnified and compared. Figures 7 and 8 show magnified images of simulation outputs applied to the Lena image with impulsive noise of 30%, showing that the conventional methods generate some distortions in the edge region. However, the output from the proposed algorithm is similar to the original image and does not have edge distortions. Figure 9 compares the full-size images of the original Pepper image, a noisy image with 30% impulsive noise added, and the simulation results of the six algorithms. It is found that the proposed method and SAM algorithm fail to correct a spike in the uniform region at the bottom center of the image; this is from the SAM algorithm's failure to correct. However, the proposed algorithm provides much better performance, especially on edge regions without any distortions, than the existing algorithms. Thus, simulations show that the performance of the proposed algorithm is

Table 1. PSNRs for Lena images with different amounts of noise (unit: dB).

Noise	Noisy image	3×3 MF	ACWM	EDB	DWMF	SAM	Proposed algorithm
5%	18.50	34.74	39.45	38.84	37.50	34.34	36.02
10%	15.48	33.74	36.60	37.37	35.97	33.57	35.34
15%	13.73	32.02	33.57	35.90	34.50	32.57	34.61
20%	12.46	29.62	30.37	34.69	33.40	31.91	34.13
25%	11.47	26.71	27.04	33.34	32.44	31.15	33.36
30%	10.49	23.84	24.02	32.55	31.45	30.29	32.50

Table 2. PSNRs for Pepper images with different amounts of noise (unit: dB).

Noise	Noisy image	3×3 MF	ACWM	EDB	DWMF	SAM	Proposed algorithm
5%	18.26	34.47	39.04	39.28	37.45	36.00	37.34
10%	15.33	33.54	36.33	37.42	36.01	34.83	36.28
15%	13.54	31.89	33.38	35.78	34.78	33.73	35.49
20%	12.27	29.93	30.73	34.29	33.51	32.66	34.74
25%	11.34	26.80	27.15	33.25	32.36	31.73	33.80
30%	10.52	23.60	23.76	32.31	31.64	31.13	33.24

Table 3. Computation times for Lena images with different amounts of noise (unit: ms).

Noise	3×3 MF	ACWM	EDB	DWMF	SAM	Proposed algorithm
5%	0.109	0.281	0.515	0.390	0.563	0.765
10%	0.109	0.203	0.515	0.406	0.594	0.812
15%	0.109	0.140	0.531	0.437	0.641	0.859
20%	0.109	0.125	0.531	0.468	0.703	0.921
25%	0.109	0.109	0.531	0.500	0.828	1.047
30%	0.109	0.11	0.531	0.531	1.047	1.281

Table 4. Computation times for Pepper images with different amounts of noise (unit: ms).

Noise	3×3 MF	ACWM	EDB	DWMF	SAM	Proposed algorithm
5%	0.109	0.281	0.531	0.390	0.562	0.781
10%	0.109	0.188	0.531	0.406	0.578	0.812
15%	0.109	0.156	0.531	0.438	0.625	0.859
20%	0.109	0.125	0.531	0.468	0.671	0.921
25%	0.109	0.109	0.531	0.515	0.750	0.968
30%	0.109	0.109	0.531	0.547	0.875	1.109

superior to existing algorithms for images that are more severely corrupted by large amounts of impulsive noise.

The objective performance evaluation compares operating times and peak signal-to-noise ratios (PSNRs) of the output images. The PSNR is equivalent to the mean square error between the original and the filtered image and is obtained using (2):

$$PSNR = 10\log\left(\frac{255^2}{MSE}\right). \tag{2}$$

Tables 1 and 2 show that, for the Lena and Pepper images, the PSNRs of the proposed algorithm are higher than those of the existing algorithms when the amount of impulsive noise in the test images is more than 20%. Specifically, as the noise amount increases, the proposed algorithm provides much higher PSNRs than the existing algorithms. In Tables 3 and 4, the operating times of all six algorithms are compared and show that the SAM filter requires the longest operating time among the existing algorithms. To determine the proper mask size, the SAM filter counts the impulsive noise in a mask. Since the proposed algorithm adopts the SAM filter for preprocessing, its operating time is slightly longer than that of the SAM filter.

V. Conclusion

This paper presented an efficient algorithm that removes impulsive noise from an image severely damaged by impulsive noise, without adding edge distortions. First, edge regions were expressed in a binarized image, and simple adjustments were applied to this binarized image to correct edge distortions. Then, direction information for edge regions was displayed using connected lines; these lines in the edge region were used as important criteria in designing a filter. For uniform regions, the existing SAM filter was applied to remove impulsive noise, and, for edge regions, a prediction filter and a line-weighted median filter using connected lines were proposed. The prediction filter forecasted the value for a noisy pixel using line information and surrounding pixels when there was connected impulse noise in the horizontal or vertical directions. However, because the prediction filter was applied in both horizontal and vertical directions, it made some mistakes with edge regions having a low slope such as 45º. To correct these possible errors, a line-weighted median filter using the line information was applied to the result of the prediction filter. Simulation results showed that the performance of the proposed algorithm is similar to that of existing algorithms when an image is corrupted with a small amount of impulsive noise but is much better when an image is severely corrupted with large amounts of impulsive noise. Specifically, when the noise content is more than 20%, existing algorithms make severe edge distortions while the proposed algorithm can reconstruct edge regions similar to those of the original image.

References

[1] F. Van der Heijden, *Image Based Measurement System*, Hoboken, NJ: John Wiley & Sons, Inc., 1994.

[2] G. Giakos et al., "Noninvasive Imaging for the New Century," *IEEE Instrum. Meas. Mag.*, vol. 2, no. 2, June 1999, pp. 32-49.

[3] I. Pitas, *Digital Image Processing Algorithms and Applications*, Hoboken, NJ: John Wiley & Sons, Inc., 2000.

[4] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd ed., Upper Saddle River, NJ: Prentice Hall, 2008.

[5] H. Ibrahim, N. Kong, and T. Ng; "Simple Adaptive Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," *IEEE Trans. Consum. Electron.*, vol. 54, no. 4, Nov. 2008, pp. 1920-1927.

[6] K. Srinivasan and D. Ebenezer, "A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noise," *IEEE Signal Process. Lett.,* vol. 14, no. 3, Mar. 2007, pp. 189-192.

[7] Y. Dong and S. Xu, "A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise," *IEEE Signal Process. Lett.*, vol. 14, no. 3, Mar. 2007, pp. 193-196.

[8] B. Jeon, G. Chae, and C. Jeong, "ACWM Filter Design for Removing Impulsive Noise," *J. Korean Institute Inf. Scientists Engineers*, Oct. 1997, pp. 142-145.

[9] S. Zhang and M.A. Karim, "A New Impulse Detector for Switching Median Filters," *IEEE Signal Process. Lett.*, vol. 9, no. 11, Nov. 2002, pp. 360-363.

[10] D. Brownrigg, "The Weighted Median Filter," *Commun. Assoc. Comput.*, Oct. 1984, pp. 807-818.

[11] L.G. Shapiro and G.C. Stockman, *Computer Vision*, Prentence Hall, 2001, pp. 137-150.

[12] R. Chan, C. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail Preserving Regularization," *IEEE Trans. Image Process.*, vol. 14, no. 4, Oct. 2005, pp. 1479-1485.

[13] H. Hwang and R. Haddad, "Adaptive Median Filters: New Algorithms and Results," *IEEE Trans. Image Process.*, vol. 14, no. 4, Apr. 1995, pp. 499-502.

[14] Z. Wang and D. Zhang, "Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Image," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, Jan. 1999, pp. 78-80.

[15] W. Luo, "Efficient Removal of Impulse Noise from Digital Images," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, May 2006, pp. 523-527.

[16] T. Chen and H.R. Wu "Adaptive Impulse Detection Using Center-Weighted Medina Filters," *IEEE Signal Process. Lett.* vol. 8, no. 1, Jan. 2001, pp. 1-3.

[17] N.I. Petrovic and V. Crnojevic, "Universal Impulse Noise Filter Based on Genetic Programming," *IEEE Trans. Image Process.*, vol. 17, no. 7, July 2008, pp. 1109-1120.

Dong-Ho Lee received his BS in electronic engineering from Hanyang University, Seoul, Korea, and his MS and PhD in electrical and computer engineering from the University of Texas at Austin, Austin, Texas, USA, in 1986, 1988, and 1991, respectively. From 1989 to 1990, he served as an officer in the Korean Army. From June 1991 to February 1994, he was a senior engineer at LG Electronics, where he was involved in the development of the DTV system and its implementation. Since 1994, he has been a professor in the Department of Electronics and Communications Engineering at Hanyang University. His current research interests include digital image processing and digital TV applications.