An Efficient Denoising Architecture for Removal of Impulse Noise in Images

Chih-Yuan Lien, Chien-Chuan Huang, Pei-Yin Chen, *Member*, *IEEE*, and Yi-Fan Lin

Abstract—Images are often corrupted by impulse noise in the procedures of image acquisition and transmission. In this paper, we propose an efficient denoising scheme and its VLSI architecture for the removal of random-valued impulse noise. To achieve the goal of low cost, a low-complexity VLSI architecture is proposed. We employ a decision-tree-based impulse noise detector to detect the noisy pixels, and an edge-preserving filter to reconstruct the intensity values of noisy pixels. Furthermore, an adaptive technology is used to enhance the effects of removal of impulse noise. Our extensive experimental results demonstrate that the proposed technique can obtain better performances in terms of both quantitative evaluation and visual quality than the previous lower complexity methods. Moreover, the performance can be comparable to the higher complexity methods. The VLSI architecture of our design yields a processing rate of about 200 MHz by using TSMC 0.18 μm technology. Compared with the state-of-the-art techniques, this work can reduce memory storage by more than 99 percent. The design requires only low computational complexity and two line memory buffers. Its hardware cost is low and suitable to be applied to many real-time applications.

Index Terms—Image denoising, impulse noise, impulse detector, architecture

1 Introduction

Image processing is widely used in many fields, such as medical imaging, scanning techniques, printing skills, license plate recognition, face recognition, and so on. In general, images are often corrupted by impulse noise in the procedures of image acquisition and transmission. The noise may seriously affect the performance of image processing techniques. Hence, an efficient denoising technique becomes a very important issue in image processing [1], [2]. According to the distribution of noisy pixel values, impulse noise can be classified into two categories: fixed-valued impulse noise and random-valued impulse noise. The former is also known as salt-and-pepper noise because the pixel value of a noisy pixel is either minimum or maximum value in gray-scale images. The values of noisy pixels corrupted by random-valued impulse noise are uniformly distributed in the range of [0, 255] for gray-scale images. There have been many methods for removing salt-and-pepper noise, and some of them perform very well [3], [4], [5], [6], [7]. The random-valued impulse noise is more difficult to handle due to the random distribution of noisy pixel values. We only focus on removing the random-valued impulse noise from the corrupted image in this paper.

- *C.-Y. Lien is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan, R.O.C. E-mail: cylien@cc.kuas.edu.tw.*
- *C.-C. Huang, P.-Y. Chen, and Y.-F. Lin are with the Department of Computer Science and Information Engineering, National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan 70101, Taiwan, R.O.C. E-mail: chien.chuan.huang@gmail.com, pychen@csie.ncku.edu.tw, ste1028@hotmail.com.*

Manuscript received 7 June 2011; revised 26 Oct. 2011; accepted 20 Dec. 2011; published online 27 Dec. 2011.
Recommended for acceptance by L. Wang.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2011-06-0379.
Digital Object Identifier no. 10.1109/TC.2011.256.

Recently, many image denoising methods have been proposed to carry out impulse noise suppression [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Some of them employ the standard median filter [8] or its modifications [9], [10]. However, these approaches might blur the image since both noisy and noise-free pixels are modified. To avoid the damage on noise-free pixels, an efficient switching strategy has been proposed in the literature [11], [12], [13]. In general, the switching median filter consists of two steps: 1) impulse detection and 2) noise filtering. It locates the noisy pixels with an impulse detector, and then filters them rather than the whole pixels of an image to avoid causing the damage on noise-free pixels. In addition to median filer, there are other methods used to carry out impulse noise. In [14], Luo proposed an alpha-trimmed mean-based method (ATMBM). It used the alpha-trimmed mean in impulse detection and replaced the noisy pixel value by a linear combination of its original value and the median of its local window. A differential rank impulse detector (DRID) was presented in [15]. The impulse detector of DRID is based on a comparison of signal samples within a narrow rank window by both rank and absolute value. In [16], Yu et al. proposed a method using a statistic of rank-ordered relative differences (RORD-WMF) to identify pixels which are likely to be corrupted by impulse noise. A directional weighted median (DWM) method proposed by Dong and Xu was presented in [17]. It is based on the differences between the current pixel and its neighbors aligned with four main directions. In [18], Petrović and Crnojević proposed a method that employed genetic programming for impulse noise filter construction. The method is based on the switching scheme with cascaded detectors and corresponding estimators.

Generally, the denoising methods can be classified into two categories: lower complexity techniques [8], [9], [10], [11], [12], [13] and higher complexity techniques [14], [15], [16], [17], [18]. The complexity of denoising algorithms

 Published by the IEEE Computer Society

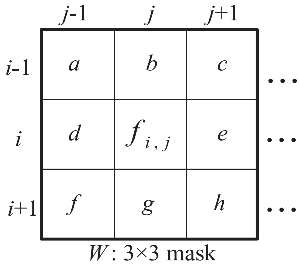


Fig. 1. A 3×3 mask centered on $p_{i,j}$.

depends mainly on the local window size, memory buffer, and iteration times. The lower complexity techniques use a fixed-size local window, require a few line buffers, and perform no iterations. Therefore, the computational complexity is low. However, the reconstructed image quality is not good enough. The higher complexity techniques yield visually pleasing images by using high computational complexity arithmetic operations, enlarging local window size adaptively or doing iterations. The higher complexity approaches require long computational time as well as full frame buffer. Today, in many practical real-time applications, the denoising process is included in end-user equipment, so there appears an increasing need of a good lower-complexity denoising technique, which is simple and suitable for low-cost VLSI implementation. Low cost is a very important consideration in purchasing consumer electronic products. To achieve the goal of low cost, less memory and easier computations are indispensable. In this paper, we focus only on the lower complexity denoising techniques because of its simplicity and easy implementation with the VLSI circuit.

The decision tree is a simple but powerful form of multiple variable analysis [19]. It can break down a complex decision-making process into a collection of simpler decisions, thus provide a solution which is often easier to interpret [20]. There have been several methods using decision tree to deal with salt-and-pepper noise [4], [5], [21], [22], [23], [37] and some of them perform well.

Based on above basic concepts, we present a novel adaptive decision-tree-based denoising method (DTBDM) and its VLSI architecture for removing random-valued impulse noise. To enhance the effects of removal of impulse noise, the results of reconstructed pixels are adaptively written back as a part of input data. The proposed design requires simple computations and two line memory buffers only, so its hardware cost is low. For a 512×512 8-bit gray-scale test image, only two line buffer $(512 \times 2 \times 8 \text{ bits})$ is needed in our design. Most state-of-the-art methods need to buffer a full image $(512 \times 512 \times 8 \text{ bits})$. In our design, 99.6 percent of storage is reduced. Furthermore, only simple arithmetic operations, such as addition and subtraction, are used in DTBDM. Especially, it can remove the noise from corrupted images efficiently and requires no previous training. Our extensive experimental results demonstrate that the proposed technique can obtain better performances in terms of both quantitative evaluation and visual quality than other lower complexity denoising methods [8], [9], [10], [11], [12], [13]. Moreover, the performance can be comparable to the higher complexity methods [14], [15],

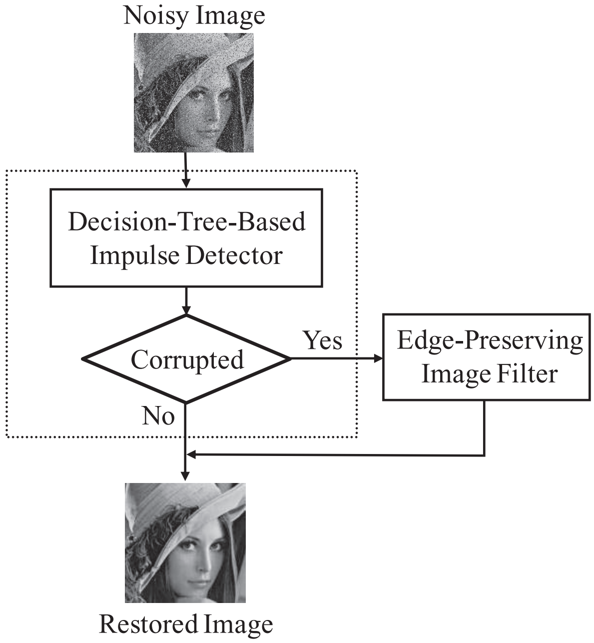


Fig. 2. The dataflow of DTBDM.

[16]. The seven-stage VLSI architecture for the proposed design was implemented and synthesized by using Verilog HDL and Synopsys Design Compiler, respectively. In our simulation, the circuit can achieve 200 MHz with only 21k gate counts by using TSMC $0.18\ \mu\text{m}$ technology.

The rest of this paper is organized as follows. The proposed DTBDM is introduced briefly in Section 2. Section 3 describes the proposed VLSI architecture in detail. Section 4 illustrates the VLSI implementation and comparisons. The conclusion is provided in Section 5.

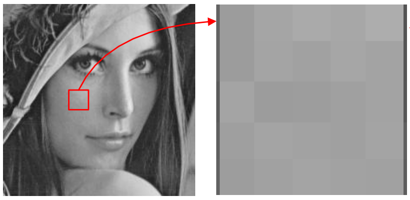
2 The Proposed DTBDM

The noise considered in this paper is random-valued impulse noise with uniform distribution as practiced in [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Here, we adopt a 3×3 mask for image denoising. Assume the pixel to be denoised is located at coordinate (i, j) and denoted as $p_{i,j}$, and its luminance value is named as $f_{i,j}$, as shown in Fig. 1. According to the input sequence of image denoising process, we can divide other eight pixel values into two sets: $W_{TopHalf}$ and $W_{BottomHalf}$. They are given as

$$W_{TopHalf} = \{a, b, c, d\}. \tag{1}$$

$$W_{BottomHalf} = \{e, f, g, h\}. \tag{2}$$

DTBDM consists of two components: decision-tree-based impulse detector and edge-preserving image filter. The detector determines whether $p_{i,j}$ is a noisy pixel by using the decision tree and the correlation between pixel $p_{i,j}$ and its neighboring pixels. If the result is positive, edge-preserving image filter based on direction-oriented filter generates the reconstructed value. Otherwise, the value will be kept unchanged. The design concept of the DTBDM is displayed in Fig. 2.



156	165	170	167	173
156	164	161	166	163
164	156	157	160	163
159	164	162	160	164
157	161	164	162	161

(a) original image (b) Region of red rectangle (c) Gray-scale value

Fig. 3. A smooth region in Lena.

134	136	134	131	133
134	131	119	111	116
120	87	58	55	56
101	65	50	48	49
106	87	70	59	70

(a) original image (b) Region of red rectangle (c) Gray-scale value

Fig. 4. A nonsmooth region in Lena.

2.1 Decision-Tree-Based Impulse Detector

In order to determine whether $p_{i,j}$ is a noisy pixel, the correlations between $p_{i,j}$ and its neighboring pixels are considered [10], [11], [14], [16], [17], [23], [24], [25], [26], [27], [28], [29], [30]. Surveying these methods, we can simply classify them into several ways—observing the degree of isolation at current pixel [10], [11], [14], [16], [17], [23], [24], [25], determining whether the current pixel is on a fringe [16], [17], [26], [27] or comparing the similarity between current pixel and its neighboring pixels [16], [28], [29], [30]. Therefore, in our decision-tree-based impulse detector, we design three modules—isolation module (IM), fringe module (FM), and similarity module (SM). Three concatenating decisions of these modules build a decision tree. The decision tree is a binary tree and can determine the status of $p_{i,j}$ by using the different equations in different modules. First, we use isolation module to decide whether the pixel value is in a smooth region. If the result is negative, we conclude that the current pixel belongs to noisy free. Otherwise, if the result is positive, it means that the current pixel might be a noisy pixel or just situated on an edge. The fringe module is used to confirm the result. If the current pixel is situated on an edge, the result of fringe module will be negative (noisy free); otherwise, the result will be positive. If isolation module and fringe module cannot determine whether current pixel belongs to noisy free, the similarity module is used to decide the result. It compares the similarity between current pixel and its neighboring pixels. If the result is positive, $p_{i,j}$ is a noisy pixel; otherwise, it is noise free. The following sections describe the three modules in detail.

2.1.1 Isolation Module

The pixel values in a smooth region should be close or locally slightly varying, as shown in Fig. 3. The differences between its neighboring pixel values are small. If there are noisy values, edges, or blocks in this region, the distribution of the values is different, as shown in Fig. 4. Therefore, we determine whether current pixel is an isolation point by observing the smoothness of its surrounding pixels. Fig. 5 shows an example of noisy image. The pixels with shadow suffering from noise have low similarity with the neighboring pixels and the so-called isolation point. The difference between it and its neighboring pixel value is large. According to the above concepts, we first detect the maximum and minimum luminance values in $W_{TopHalf}$, named as *TopHalf_max*, *TopHalf_min*, and calculate the difference between them, named as *TopHalf_diff*. For $W_{BottomHalf}$, we apply the same idea to obtain *BottomHalf_diff*. The two difference values are compared with a threshold Th_IM_a to decide whether the surrounding region belongs to a smooth area. The equations are as

$$TopHalf\_diff = TopHalf\_max - TopHalf\_min. \tag{3}$$

$$BottomHalf\_diff = BottomHalf\_max - BottomHalf\_min. \tag{4}$$

$$DecisionI = \begin{cases} true, & \text{if } (TopHalf\_diff \geq Th\_IM_a) \\ & \text{or } (BottomHalf\_diff \geq Th\_IM_a) \\ false, & \text{otherwise.} \end{cases} \tag{5}$$

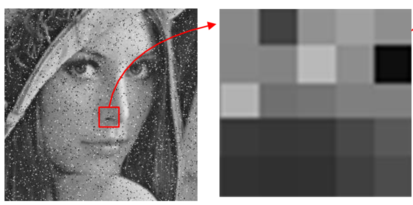
Next, we take $p_{i,j}$ into consideration. Two values must be calculated first. One is the difference between $f_{i,j}$ and *TopHalf_max*; the other is the difference between $f_{i,j}$ and *TopHalf_min*. After the subtraction, a threshold Th_IM_b is used to compare these two differences. The same method as in the case of $W_{BottomHalf}$ is applied. The equations are as

$$IM\_TopHalf = \begin{cases} true, & \text{if } (|f_{i,j} - TopHalf\_max| \geq Th\_IM_b) \\ & or\ (|f_{i,j} - TopHalf\_min| \geq Th\_IM_b) \\ false, & \text{otherwise.} \end{cases} \tag{6}$$

$$IM\_BottomHalf = \begin{cases} true, & \text{if } (|f_{i,j} - BottomHalf\_max| \geq Th\_IM_b) \\ & or\ (|f_{i,j} - BottomHalf\_min| \geq Th\_IM_b) \\ false, & \text{otherwise.} \end{cases} \tag{7}$$

$$DecisionII = \begin{cases} true, & \text{if } (IM\_TopHalf = true) \\ & \text{or } (IM\_BottomHalf = true) \\ false, & \text{otherwise.} \end{cases} \tag{8}$$

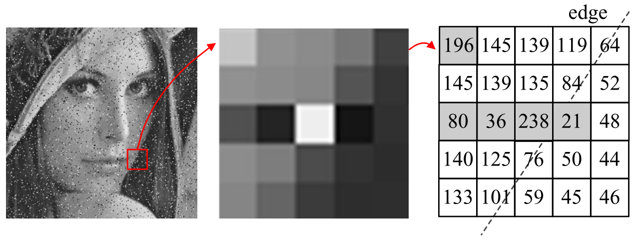
Finally, we can make a temporary decision whether $p_{i,j}$ belongs to a suspected noisy pixel or is noisy free.



136	66	145	160	142
134	131	185	141	14
178	111	116	127	127
58	55	56	72	88
50	48	49	63	76

(a) original image (b) Region of red rectangle (c) Gray-scale value

Fig. 5. The difference between noisy and neighboring pixels in Lena.



(a) original image (b) Region of red rectangle (c) Gray-scale value

Fig. 6. The edge region in Lena.

2.1.2 Fringe Module

If $p_{i,j}$ has a great difference with neighboring pixels, it might be a noisy pixel (discussed in Section 2.1.1 IM) or just situated on an edge, as shown in Fig. 6. How to conclude that a pixel is noisy or situated on an edge is difficult. In order to deal with this case, we define four directions, from E_1 to E_4, as shown in Fig. 7. We take direction E_1 for example. By calculating the absolute difference between $f_{i,j}$ and the other two pixel values along the same direction, respectively, we can determine whether there is an edge or not. The detailed equations are as

$$FM\_E_1 = \begin{cases} false, & \text{if } (|a - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|h - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|a - h| \geq Th\_FM_b) \\ true, & \text{otherwise.} \end{cases} \tag{9}$$

$$FM\_E_2 = \begin{cases} false, & \text{if } (|c - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|f - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|c - f| \geq Th\_FM_b) \\ true, & \text{otherwise.} \end{cases} \tag{10}$$

$$FM\_E_3 = \begin{cases} false, & \text{if } (|b - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|g - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|b - g| \geq Th\_FM_b) \\ true, & \text{otherwise.} \end{cases} \tag{11}$$

$$FM\_E_4 = \begin{cases} false, & \text{if } (|d - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|e - f_{i,j}| \geq Th\_FM_a) \\ & \text{or } (|d - e| \geq Th\_FM_b) \\ true, & \text{otherwise.} \end{cases} \tag{12}$$

$$Decision\ III = \begin{cases} false, & \text{if } (FM\_E_1) \text{ or } (FM\_E_2) \\ & \text{or } (FM\_E_3) \text{ or } (FM\_E_4) \\ \text{true}, & \text{otherwise.} \end{cases} \tag{13}$$

2.1.3 Similarity Module

The last module is similarity module. The luminance values in mask W located in a noisy-free area might be close. The median is always located in the center of the variational series, while the impulse is usually located near one of its ends. Hence, if there are extreme big or small values, that implies the possibility of noisy signals. According to this concept, we sort nine values in ascending order and obtain the fourth, fifth, and sixth values which are close to the median in mask W. The fourth, fifth, and sixth values are represented as $4_{th}inW_{i,j}$, $MedianInW_{i,j}$, and $6_{th}inW_{i,j}$. We define $Max_{i,j}$ and $Min_{i,j}$ as

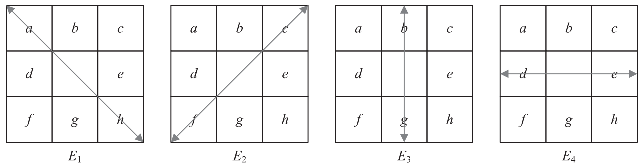


Fig. 7. Four directions in DTBDM.

$$\begin{aligned} Max_{i,j} &= 6_{th}inW_{i,j} + Th\_SM_a, \\ Min_{i,j} &= 4_{th}inW_{i,j} - Th\_SM_a. \end{aligned} \tag{14}$$

$Max_{i,j}$ and $Min_{i,j}$ are used to determine the status of pixel $p_{i,j}$. However, in order to make the decision more precisely, we do some modifications as

$$N_{\max} = \begin{cases} Max_{i,j}, & \text{if } (Max_{i,j} \leq MedianInW_{i,j} \\ & \quad + Th\_SM_b) \\ MedianInW_{i,j} & \\ \quad + Th\_SM_b, & \text{otherwise.} \end{cases} \tag{15}$$

$$N_{\min} = \begin{cases} Min_{i,j}, & \text{if } (Min_{i,j} \geq MedianInW_{i,j} \\ & \quad - Th\_SM_b) \\ MedianInW_{i,j} & \\ \quad - Th\_SM_b, & \text{otherwise.} \end{cases} \tag{16}$$

Finally, if $f_{i,j}$ is not between N_{max} and N_{min}, we conclude that $p_{i,j}$ is a noise pixel. Edge-preserving image filter will be used to build the reconstructed value. Otherwise, the original value $f_{i,j}$ will be the output. The equation is as

$$Decision\ IV = \begin{cases} true, & \text{if } (f_{i.j} \geq N_{\max}) \text{ or } (f_{i.j} \leq N_{\min}) \\ false, & \text{otherwise.} \end{cases} \tag{17}$$

Obviously, the threshold affects the quality of denoised images of the proposed method. A more appropriate threshold contributes to achieve a better detection result. However, it is not easy to derive an optimal threshold through analytic formulation. The fixed values of thresholds make our algorithm simple and suitable for hardware implementation. According to our extensive experimental results, the thresholds Th_IM_a, TH_IM_b, Th_FM_a, Th_FM_b, Th_SM_a, and Th_SM_b are all predefined values and set as 20, 25, 40, 80, 15, and 60, respectively.

2.2 Edge-Preserving Image Filter

To locate the edge existing in the current W, a simple edge-preserving technique which can be realized easily with VLSI circuit is adopted. The dataflow and the pseudocode of our edge-preserving image filter are shown in Figs. 8 and 9, respectively. Here, we consider eight directional differences, from D_1 to D_8, to reconstruct the noisy pixel value, as shown in Fig. 10 and (18). Only those composed of noise-free pixels are taken into account to avoid possible misdetection. Directions passing through the suspected pixels are discarded to reduce misdetection. Therefore, we use $Max_{i,j}$ and $Min_{i,j}$, defined in similarity module, to determine whether the values of d, e, f, g, and h are likely corrupted, respectively. If the pixel is likely being corrupted by noise, we don't consider the direction including the suspected pixel. In the second block, if d, e, f, g, and h are all suspected to be noisy pixels, and no edge can be processed, so $\hat{f}_{i,j}$ (the

REFERENCES

[1] R.C. Gonzalez and R.E. Woods, *Digital Image Processing.* Pearson Education, 2007.
[2] W.K. Pratt, *Digital Image Processing.* Wiley-Interscience, 1991.
[3] H. Hwang and R.A. Haddad, "Adaptive Median Filters: New Algorithms and Results," *IEEE Trans. Image Processing,* vol. 4, no. 4, pp. 499-502, Apr. 1995.
[4] S. Zhang and M.A. Karim, "A New Impulse Detector for Switching Median Filter," *IEEE Signal Processing Letters,* vol. 9, no. 11, pp. 360-363, Nov. 2002.
[5] R.H. Chan, C.W. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization," *IEEE Trans. Image Processing,* vol. 14, no. 10, pp. 1479-1485, Oct. 2005.
[6] P.E. Ng and K.K. Ma, "A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images," *IEEE Trans. Image Processing,* vol. 15, no. 6, pp. 1506-1516, June 2006.
[7] P.-Y. Chen and C.-Y. Lien, "An Efficient Edge-Preserving Algorithm for Removal of Salt-and-Pepper Noise," *IEEE Signal Processing Letters,* vol. 15, pp. 833-836, Dec. 2008.
[8] T. Nodes and N. Gallagher, "Median Filters: Some Modifications and Their Properties," *IEEE Trans. Acoustics, Speech, Signal Processing,* vol. ASSP-30, no. 5, pp. 739-746, Oct. 1982.
[9] S.-J. Ko and Y.-H. Lee, "Center Weighted Median Filters and Their Applications to Image Enhancement," *IEEE Trans. Circuits Systems,* vol. 38, no. 9, pp. 984-993, Sept. 1991.
[10] T. Sun and Y. Neuvo, "Detail-Preserving Median Based Filters in Image Processing," *Pattern Recognition Letters,* vol. 15, pp. 341-347, Apr. 1994.
[11] E. Abreu, M. Lightstone, S.K. Mitra, and K. Arakawa, "A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images," *IEEE Trans. Image Processing,* vol. 5, no. 6, pp. 1012-1025, June 1996.
[12] T. Chen and H.R. Wu, "Adaptive Impulse Detection Using Center-Weighted Median Filters," *IEEE Signal Processing Letters,* vol. 8, no. 1, pp. 1-3, Jan. 2001.
[13] T. Chen and H.R. Wu, "Space Variant Median Filters for the Restoration of Impulse Noise Corrupted Images," *IEEE Trans. Circuits Systems II, Analog Digital Signal Processing,* vol. 48, no. 8, pp. 784-789, Aug. 2001.
[14] W. Luo, "An Efficient Detail-Preserving Approach for Removing Impulse Noise in Images," *IEEE Signal Processing Letters,* vol. 13, no. 7, pp. 413-416, July 2006.
[15] I. Aizenberg and C. Butakoff, "Effective Impulse Detector Based on Rank-Order Criteria," *IEEE Signal Processing Letters,* vol. 11, no. 3, pp. 363-366, Mar. 2004.
[16] H. Yu, L. Zhao, and H. Wang, "An Efficient Procedure for Removing Random-Valued Impulse Noise in Images," *IEEE Signal Processing Letters,* vol. 15, pp. 922-925, 2008.
[17] Y. Dong and S. Xu, "A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise," *IEEE Signal Processing Letters,* vol. 14, no. 3, pp. 193-196, Mar. 2007.
[18] N.I. Petrovic and V. Crnojevic, "Universal Impulse Noise Filter Based on Genetic Programming," *IEEE Trans. Image Processing,* vol. 17, no. 7, pp. 1109-1120, July 2008.
[19] B. De Ville, *Decision Trees for Business Intelligence and Data Mining.* SAS Publishing, 2007.
[20] S. Rasoul Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," *IEEE Trans. Systems Man, Cybernetics,* vol. 21, no. 3, pp 660-674, May 1991.
[21] H.-H. Tsai, X.-P. Lin, and B.-M. Chang, "An Image Filter with a Hybrid Impulse Detector Based on Decision Tree and Particle Swarm Optimization," *Proc. IEEE Int'l Conf. Machine Learning and Cybernetics,* July 2009.
[22] H.-L. Eng and K.-K. Ma, "Noise Adaptive Soft-Switching Median Filter," *IEEE Trans. Image Processing,* vol. 10, no. 2, pp. 242-251, Feb. 2001.
[23] G. Pok, J. Liu, and A.S. Nair, "Selective Removal of Impulse Noise Based on Homogeneity Level Information," *IEEE Trans. Image Processing,* vol. 12, no. 1, pp. 85-92, Jan. 2003.
[24] Z. Wang and D. Zhang, "Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," *IEEE Trans. Circuits Systems II, Analog Digital Signal Processing,* vol. 46, no. 1, pp. 78-80, Jan. 1999.
[25] A.S. Awad and H. Man, "High Performance Detection Filter for Impulse Noise Removal in Images," *IEEE Electronic Letters,* vol. 44, no. 3, pp. 192-194, Jan. 2008.
[26] X. Zhang and Y. Xiong, "Impulse Noise Removal Using Directional Difference Based Noise Detector and Adaptive Weighted Mean Filter," *IEEE Signal Processing Letters,* vol. 16, no. 4, pp. 295-298, Apr. 2009.
[27] Z. Xu, H.R. Wu, B. Qiu, and X. Yu, "Geometric Features-Based Filtering for Suppression of Impulse Noise in Color Images," *IEEE Trans. Image Processing,* vol. 18, no. 8, pp. 1742-1759, Aug. 2009.
[28] F.J. Gallegos-Funes, V.I. Ponomaryov, S. Sadovnychiy, and L. Nino-de-Rivera, "Median M-Type K-Nearest Neighbour (MM-KNN) Filter to Remove Impulse Noise from Corrupted Images," *IEEE Electronics Letters,* vol. 38, no. 15, pp. 786-787, July 2002.
[29] I. Aizenberg, C. Butakoff, and D. Paliy, "Impulsive Noise Removal Using Threshold Boolean Filtering Based on the Impulse Detecting Functions," *IEEE Signal Processing Letters,* vol. 12, no. 1, pp. 63-66, Jan. 2005.
[30] Z. Wang and D. Zhang, "Restoration of Impulse Noise Corrupted Images Using Long-Range Correlation," *IEEE Signal Processing Letters,* vol. 5, no. 1, pp. 4-7, Jan. 1998.
[31] 0.18 μm TSMC/Artisan Memory Compiler, http://www.artisan.com, 2012.
[32] Synopsys Inc., "DesignWare Building Block IP," http://www.synopsys.com, 2012.
[33] S.-C. Hsia, "Parallel VLSI Design for a Real-Time Video-Impulse Noise-Reduction Processor," *IEEE Trans. Very Large Scale Integration Systems,* vol. 11, no. 4, pp. 651-658, Aug. 2003.
[34] I. Andreadis and G. Louverdis, "Real-Time Adaptive Image Impulse Noise Suppression," *IEEE Trans. Instrumentation and Measurement,* vol. 53, no. 3, pp. 798-806, June 2004.
[35] V. Fischer, R. Lukac, and K. Martin, "Cost-Effective Video Filtering Solution for Real-Time Vision Systems," *EURASIP J. Applied Signal Processing,* vol. 13, pp. 2026-2042, 2005.
[36] T. Matsubara, V.G. Moshnyaga, and K. Hashimoto, "A FPGA Implementation of Low-Complexity Noise Removal," *Proc. 17th IEEE Int'l Conf. Electronics, Circuits, and Systems (ICECS '10),* pp. 255-258, Dec. 2010.
[37] P.-Y. Chen, C.-Y. Lien, and H.-M. Chuang, "A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise," *IEEE Trans. Very Large Scale Integration Systems,* vol. 18, no. 3, pp. 473-481, Mar. 2010.

Chih-Yuan Lien received the BS and MS degrees in computer science and information engineering from National Taiwan University, Taiwan (R.O.C.), in 1996 and 1998, respectively, and the PhD degree in computer science and information engineering from National Cheng Kung University, Taiwan (R.O.C.), in 2009. He is currently an assistant professor in the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan (R.O.C.). His research interests include image processing, VLSI chip design, and video coding system.

Chien-Chuan Huang received the BS and MS degree in computer science and information engineering from National Cheng Kung University, Taiwan (R.O.C.), in 2005 and 2008, respectively. Since September 2008, he has been working toward the PhD degree in computer science and information engineering from National Cheng Kung University, Taiwan (R.O.C.). His research interests include image processing, VLSI chip design, and data compression.

Pei-Yin Chen received the BS degree in electrical engineering from National Cheng Kung University, Taiwan (R.O.C.), in 1986, the MS degree in electrical engineering from Penn Sate University, Pennsylvania, in 1990, and the PhD degree in electrical engineering from National Cheng Kung University, Taiwan (R.O.C.), in 1999. He is currently a professor in the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan (R.O.C.). His research interests include VLSI chip design, video compression, fuzzy logic control, and gray prediction. He is a member of the IEEE.

Yi-Fan Lin received the BS degree in computer science from National Sun Yat-sen University, Taiwan (R.O.C.), in 2006, and the MS degree in computer science and information engineering from National Cheng Kung University, Taiwan (R.O.C.), in 2009. His research interests include VLSI chip design, data compression, and image processing.