

# Realization of a Real-Time Image Denoising System for Dashboard Camera Applications

Chu Yu<sup>✉</sup>, Member, IEEE, and Li-Zhong Hou

**Abstract**—Noise interference during the acquisition of digital images can severely degrade image quality, particularly for images captured under low-light conditions; however, the removal of image noise requires sophisticated digital image processing systems. This study presents a hardware-based solution to real-time image denoising using an existing algorithm designed for the removal of mixed impulse noise (salt-and-pepper and random-valued impulse noise), while preserving image edge details and image borders, without the need for additional computation time or memory capacity. Note that mixed impulse noise is typical of most real-world situations, such as the video noise associated with dashboard cameras. The proposed design was implemented using 180 nm complementary metal-oxide-semiconductor (CMOS) technology, consuming only 21.7 mW when operated at 200 MHz. This operating frequency allows the proposed chip to process noisy video streams with resolution of 1920 × 1080 at 60 frames per second in real time. In terms of image restoration, the proposed algorithm achieved image quality on par with that achieved using software simulation. We also demonstrated the efficacy of the proposed scheme in denoising noisy video images from a dashboard camera.

**Index Terms**—Digital images, impulse noise, dashboard cameras.

## I. INTRODUCTION

DIGITAL images are subject to various forms of noise during signal acquisition, particularly when dealing with low-light environments. Fixed-value impulse noise refers to salt-and-pepper noise, whereas random-value refers to the uniform dispersion of pixels of random values with dynamic range from 0 to 255. This type of noise appears as falling snowflakes of various brightness, which must be removed to preserve the quality of the resulting images and facilitate image processing (e.g., coding and object recognition). This paper presents an efficient filtering scheme for the removal of fixed-valued and random-valued impulse noise, both of which are encountered when using digital video recorders, such as those employed in dashboard cameras. The proposed scheme was then implemented using an application specific integrated circuit (ASIC) with support for real-time image processing.

Manuscript received January 31, 2022; revised April 18, 2022 and May 3, 2022; accepted May 13, 2022. Date of publication May 17, 2022; date of current version June 3, 2022. This work was supported in part by the Taiwan Ministry of Science and Technology under Grant MOST 109-2221-E-197-030. (Corresponding author: Chu Yu.)

The authors are with the Department of Electronic Engineering, National Ilan University, Yilan 260007, Taiwan (e-mail: chu@niu.edu.tw; houliz1202@qq.com).

Digital Object Identifier 10.1109/TCE.2022.3175796

Dashboard cameras are used in many real-world applications, such as scene recognition [1]–[5], driver assistance systems [6]–[26], and the monitoring of traffic incidents [27]–[28]. Note that evidence pertaining to faces, license plates, and other details can be useful in criminal proceedings. Hnoohom and Thanapattherakul [1] proposed a machine learning (ML) prediction model for use in identifying problems associated with the images from dashboard cameras. Wu *et al.* [2] sought to reduce the storage requirements of dashboard cameras, while improving object recognition performance (e.g., cars, traffic signals, and pedestrians). Altun and Celenk [3] employed a dashboard camera in a vision-based driver assistance system aimed at enhancing scene awareness. Okumura *et al.* [4] presented a system that allows the use of in-vehicle cameras to estimate the narrowing of roads due to the piling up of snow. Takase *et al.* [5] presented a system that allows the use of in-vehicle cameras to estimate the surface conditions of expressways in the winter. Yang and Zheng [6] and Wang and Kato [7] employed dashboard cameras to record on-road collisions. Lai *et al.* [8] developed a vehicle detection system that uses cameras in conjunction with deep learning (DL) to prevent vehicle collisions. Ogura and Nagasaki [9] presented a DL-based method by which to improve the visibility of pedestrians to vehicle-mounted cameras at night. Kawashita and Wada [10] proposed an pedestrian-vehicle collision avoidance system capable of detecting multiple pedestrians simultaneously. Madhumitha *et al.* [11] implemented a vision system within a novel heuristic unimodal model to estimate the likelihood of on-road collisions. Kim *et al.* [12] experimented with object detection and localization in on-road environments using a deep neural network. Marković *et al.* [13] developed a post-processing class-dependent method in which semantic segmentation is used to enhance the accuracy and configurability of front-view cameras in automobiles. Mody *et al.* [14] developed an automated driver assistance system using an signal processing pipeline for front-view cameras. Janjatovic *et al.* [15] proposed a universal software development kit for advanced driver assistance systems. Gupta and Choudhary [16] employed the dashboard camera in an unsupervised learning scheme aimed at real-time lane detection (LD). Kim *et al.* [17] employed lane and vehicle detection in an automated collision warning system. Vajak *et al.* [18] proposed a solution to LD based on classical computer vision. Hashimoto [19] conducted experiments to evaluate the use of cameras in identifying 10 types of lane markers under a range of operating conditions. Gojak *et al.* [20]

presented a portable bird's eye view system to facilitate parking and minimize the likelihood of a traffic accident. Jeong and Song [21] proposed a night-time vehicle detection system that uses only gray-scale values from the rear lamps. Yoshizawa and Nakatoh [22] presented a system by which to detect vehicles behind obstacles, based on diffraction of the sounds they emit. Yi and Jung [23] implemented a forward vehicle start alarm function using only a dashboard camera. Gupta and Choudhary [24] developed a real-time framework for the detection and recognition of traffic signs using a dashboard camera. Abadi *et al.* [25] employed a deep neural network to characterize the orientation and movement of the head and body based on images from the dashboard camera. Poon *et al.* [26] developed a DL-based method by which to monitor cabin conditions, including distractions due to in-vehicle occupants (e.g., back seat passengers and pet dogs). Hara *et al.* [27] developed a light-weight model by which to record the movement of pedestrians as evidence in traffic incidents. Yao *et al.* [28] developed an unsupervised approach to traffic accident detection using the dashboard camera.

Note that all of the above applications require high-quality images; i.e., with minimal noise. Unfortunately, dashboard cameras are subject to impulse noise during image acquisition, and particularly in low-light conditions. Good image quality can only be achieved by removing as much noise as possible.

Image denoising methods can be divided into spatial filtering [29]–[34], transform-based filtering [35]–[38], and DL-based filtering [39]–[55]. Spatial filtering can be further subdivided into linear and non-linear filtering. Conventional linear filtering is ill-suited to the removal impulse noise, due to the fact that it removes noise-related pixels as well as pixels that are unaffected by noise. This is particularly true for situations of high noise density and random-value impulse noise. Researchers have developed a wide range of non-linear digital filters using decision-based algorithms [29], edge-preserving algorithms [30], and denoising-based clustering algorithms [31] for the removal of fixed-value impulse noise as well as random-value impulse noise [32]. Thaipanich *et al.* [33] proposed an adaptive non-local means algorithm for image denoising, which is especially advantageous in situations where the noise level is high. Yu *et al.* [34] presented an efficient edge-based bilateral filter for the removal of actual noise. Transform filtering methods based on image priors, including nonlocal self-similarity models [35]–[37] and sparse models [38], have proven highly effective in denoising images containing additive white Gaussian noise (AWGN). However, this approach imposes high computational complexity and requires the manual adjustment of multiple parameters. Most state-of-the-art image denoising methods for the removal of AWGN employ DL to improve image quality. DL-based denoisers have been included in various commercial software packages. DL-based denoisers can be subdivided into paired supervised learning models [39]–[41], unpaired supervision learning models [42]–[46], and self-supervised learning models [47]–[54].

Supervised learning models train a denoising network over an external dataset comprising pairs of noisy and clean (ground truth) images. This provides powerful denoising capability; however, it necessitates the addition of synthetic noise. Furthermore, obtaining a large dataset of clean images is difficult in many situations. The scheme proposed by Lehtinen *et al.* [55] employed a training dataset of identical images respectively corrupted using two types of synthetic noise to eliminate the need for noisy-clean image pairs.

Unpaired supervised learning models eliminate the need for paired images, thereby making the collection of a usable database feasible in most real-world situations.

Self-supervised learning models require only noisy images for training (i.e., eliminate the need for clean counterpart images); however, the denoising performance of this approach cannot match that of the supervised learning models.

Similar to transform filtering, DL-based denoisers impose high computational overhead, exceeding that of spatial filtering. DL-based denoisers provide high restoration capability; however, they require hardware components that are too expensive for low-cost commercial devices. Spatial filtering is often used as an alternative to transform-based or DL-based filtering for the removal of impulse noise. Note that most of the noise affecting dashboard cameras is impulse noise, particularly in low-light environments. Our objective in this work was to enable real-time image denoising using consumer-grade dashboard cameras; therefore, we opted for filters with lower computational requirements (i.e., non-linear digital filters).

Advances in non-linear digital filters have prompted the development of various hardware architectures [56]–[62]. The scheme proposed by Dutta *et al.* [56] employed a massively parallel architecture by which to implement a bilateral filter algorithm. Matsubara *et al.* [57] proposed a low-complexity scheme for the removal of random-valued impulse noise. Chen *et al.* [58] devised an efficient VLSI implementation for the removal of salt-and-pepper noise with a mechanism for edge preservation. Lien *et al.* [59] presented an efficient denoising scheme and corresponding VLSI architecture for the removal of random-valued impulse noise. Gabiger-Rose *et al.* [60] proposed a synchronous FPGA implementation of a bilateral filter with edge preservation for real-time image processing. Chandni and Pushpkumari [61] presented a bilateral filter for grayscale and color image denoising. Based on distance-oriented grouping and hardware resource sharing, Lien *et al.* [62] presented a low-cost hardware architecture for real-time image processing using a bilateral filter.

The non-linear filtering architecture proposed in the current study is based on the algorithm described by Hou *et al.* [63], which employs a  $3 \times 3$  mask to determine whether the detected pixels are noisy or noise-free. The decision is based on the distance between the current detected pixel and the median value of the most probable edge. The value of a pixel deemed noise-free is unchanged, whereas the value of those identified as noise are replaced by the median value of the most probable edge.

The main contributions of this study are outlined as follows:

**Algorithm 1** Proposed Algorithm

---

**Input:**  $Y$  /\* a noisy image \*/  
**Output:**  $y$  /\* a denoising image \*/

```

1:  $Y \leftarrow$  read a noisy image;
2: Set  $TH$  /* Threshold value */;
3: Initialize  $Nmax \leftarrow 255$ ,  $Nmin \leftarrow 0$ ,  $Vmax \leftarrow 512$ 
4: for ( $i = 1, i <= N$ ,  $i++$ ) do
5:   for ( $j = 1, j <= M$ ,  $j++$ ) do
6:     Get a  $3 \times 3$  mask centered on ( $i, j$ ) from  $Y$ , as shown Fig. 1;
7:      $WD_1 \leftarrow |r_5 - r_0| + |r_8 - r_3|$ ;
8:      $WD_2 \leftarrow |r_8 - r_1| + |r_7 - r_0|$ ;
9:      $WD_3 \leftarrow |r_7 - r_1| \times 2$ ;
10:     $WD_4 \leftarrow |r_7 - r_2| + |r_6 - r_1|$ ;
11:     $WD_5 \leftarrow |r_6 - r_5| + |r_3 - r_2|$ ;
12:     $WD_6 \leftarrow |r_8 - r_3| \times 2$ ;
13:     $RV_1 \leftarrow$  Find the median value of the most probable edge from
       $WD_1 \sim WD_6$ ;
14:     $ADr_4 \leftarrow |r_4 - RV_1|$ ;
15:     $ADe \leftarrow |r_3 - RV_1|$ ;
16:     $ADf \leftarrow |r_2 - RV_1|$ ;
17:     $ADg \leftarrow |r_1 - RV_1|$ ;
18:     $ADh \leftarrow |r_0 - RV_1|$ ;
19:    if  $r_4 == Nmax$  ||  $r_4 == Nmin$  ||  $|ADr_4| > TH$  then
20:      If one of  $r_0 \sim r_3$  pixels is equal to  $Nmax$  or  $Nmin$  or larger
        than  $TH$ , then the relevant directional differences are disabled
        (set the relevant directional difference  $WD_i$  to  $Vmax$ ).
21:      if  $WD_1 == Vmax \& \& WD_2 == Vmax \& \& r_0 > 0 \& \& r_0 < 255$ 
        then  $WD_7 \leftarrow |a - h| \times 2$ ;
        else  $WD_7 \leftarrow Vmax$ ; /*  $WD_7$  is disabled */
        end if
22:      if  $WD_4 == Vmax \& \& WD_5 == Vmax \& \& r_2 > 0 \& \& r_2 < 255$ 
        then  $WD_8 \leftarrow |c - f| \times 2$ ;
        else  $WD_8 \leftarrow Vmax$ ; /*  $WD_8$  is disabled */
        end if
23:       $RV_2 \leftarrow$  Find the median value of the most probable edge from
       $WD_1 \sim WD_8$ ;
24:       $y(i, j) \leftarrow$  Find a median value of  $(RV_2, r_7, r_5, r_3, r_1)$ ;
25:    end if
26:  end for
27: end for
```

---

- 1) The proposed architecture enables the computationally efficient removal of fixed-value and random-value impulse noise in real time, which makes it ideal for dashboard camera applications.
- 2) The proposed algorithm accounts for edge details in the detection of noisy pixels as well as in compensation, thereby enabling performance superior to a similar method proposed by Lien *et al.* [32].
- 3) We developed a hardware-based scheme capable of automatically symmetrizing pixel extensions in order to prevent border distortion and thereby improve image quality without the need for additional clock cycles or memory space.

The remainder of this paper is organized as follows. Section II presents a review of the image denoising algorithm on which the proposed algorithm is based. Section III presents the proposed architecture. Section IV outlines our realization of the proposed architecture in a practical image denoising system. Section V provides a performance comparison of various image denoising schemes. Section VI concludes the paper.

## II. PROPOSED IMAGE DENOISING SCHEME

The proposed algorithm in Algorithm 1 can be divided into three steps. In the first step, the current detected pixel is designated as the center of a  $3 \times 3$  mask for use in calculating

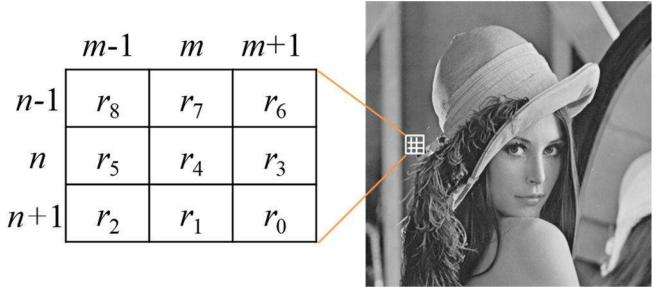


Fig. 1.  $3 \times 3$  mask corresponding to current detected pixel  $r_4$ .

six directional differences based on mask content. In the second step, the median value of the most probable image edge is derived from the six directional differences obtained in the first step. We also obtain five important difference values for use as a reference in subsequent noise detection. In the third step, the current detected pixel is designated as noisy or noise-free in accordance with the results from the previous steps. Noisy pixels are then replaced by the median value of the most probable image edge selected from the eight directional differences. The proposed scheme is detailed as follows:

*Step 1:* Assume that the size of the noisy test image is  $512 \times 512$ . Let  $P_{n,m}$  be a  $3 \times 3$  mask fetched from the noisy test image and  $p_{i,j}$  denote the current detected pixel at coordinate  $(i, j)$ , as follows:

$$P_{n,m} = \{p_{i,j} | n-1 \leq j \leq n+1, m-1 \leq i \leq m+1\}. \quad (1)$$

Fig. 1 presents an example of the above-mentioned  $3 \times 3$  mask  $P_{n,m}$ , in which the nine pixels are denoted  $r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7$ , and  $r_8$ , respectively. Here  $r_4$  refers to the current detection pixel. Implementation of the  $3 \times 3$  mask requires the extension of pixels to boundaries in order to prevent image border distortion. Pixel extension schemes have been developed using symmetrization, zero-padding, or smooth padding. For the sake of image quality, we adopted the symmetrization scheme in dealing with the four image borders.

Based on the processing mask in Fig. 1, eight directional differences ( $WD_1 \sim WD_8$ ) around current detected pixel  $r_4$  can be defined as follows [30]:

$$WD_1 = |r_5 - r_0| + |r_8 - r_3|, \quad WD_2 = |r_8 - r_1| + |r_7 - r_0|, \quad (2)$$

$$WD_3 = |r_7 - r_1| \times 2, \quad WD_4 = |r_7 - r_2| + |r_6 - r_1|, \quad (2)$$

$$WD_5 = |r_6 - r_5| + |r_3 - r_2|, \quad WD_6 = |r_8 - r_3| \times 2, \text{ and} \quad (3)$$

$$WD_7 = |r_8 - r_0| \times 2, \quad WD_8 = |r_6 - r_2| \times 2. \quad (3)$$

Equations (2) and (3) are used to determine the degree of correlation with the eight corresponding directions (image edges). The smallest difference among the eight directions indicates the most probable edge.

*Step 2:* Our objective here is to identify the median value of the most probable edge, denoted as  $WD_{min}$ , from among the eight differences in direction. To simplify the calculation, we take into consideration only six differences in direction, such that  $WD_7$  and  $WD_8$  are disregarded. To determine whether the current pixel is destroyed, a reference value  $RV_1$  [30] is

introduced here, as follows:

$$RV_1 = \begin{cases} \frac{r_8+r_5+r_3+r_0}{4}, & \text{if } WD_{\min} = WD_1 \\ \frac{r_8+r_7+r_1+r_0}{4}, & \text{if } WD_{\min} = WD_2 \\ \frac{r_7+r_1}{4}, & \text{if } WD_{\min} = WD_3 \\ \frac{r_7+r_6+r_2+r_1}{4}, & \text{if } WD_{\min} = WD_4 \\ \frac{r_6+r_5+r_3+r_2}{4}, & \text{if } WD_{\min} = WD_5 \\ \frac{r_5+r_3}{2}, & \text{if } WD_{\min} = WD_6 \end{cases} \quad (4)$$

If current pixel  $r_4$  is larger than reference value  $RV_1$  in Eq. (4), then it is deemed noisy.

We obtain the absolute difference (AD) between current pixel  $r_4$  and the four undetected pixels ( $r_0, r_1, r_2$ , and  $r_3$ ) from reference value  $RV_1$  as follows:

$$ADr_4 = |r_4 - RV_1|,$$

$$ADE = |r_3 - RV_1|, ADf = |r_2 - RV_1|, \quad (5)$$

$$ADg = |r_1 - RV_1|, \text{ and } ADh = |r_0 - RV_1|. \quad (6)$$

The results from Eqs. (5) and (6) are used to determine whether the five pixels ( $ADr_4, ADE, ADf, ADg$ , and  $ADh$ ) are corrupted. If they exceed a given threshold value, they are designated as noisy, and otherwise designated as noise-free.

*Step 3:* The current detected pixel can be identified as noisy based on  $ADr_4$  in (5) in conjunction with the maximum and minimum gray values. If the current detected pixel is not equal to the maximum or minimum values, or if  $ADr_4$  is less than a threshold value, then the pixel is deemed noise-free and therefore remains unchanged. Otherwise, a replacement operation is launched. Note that the proposed scheme also estimates  $ADE, ADf, ADg$ , and  $ADh$ , which are derived using (6). If one of the four values exceeds a threshold value, then the corresponding pixels ( $r_0, r_1, r_2$ , and  $r_3$ ) are likely to be noisy. In subsequent noise detection tasks, the relevant directional differences associated with one of the four undetected pixels are disabled.

In the previous step, pixels  $r_2$  and  $r_0$  were tested. If  $r_0$  is equal to the minimum or maximum values, then directional differences  $WD_1$  and  $WD_2$  are also disabled, thereby introducing a new directional difference  $WD_7$ . If pixel  $r_0$  is deemed noisy, then  $WD_7$  is disabled. Similarly, if pixel  $r_2$  is noise-free, then an additional directional difference  $WD_8$  is established; and otherwise disregarded. The reference value  $RV_2$  associated with  $WD_7$  and  $WD_8$  is estimated as follows:

$$RV_2 = \begin{cases} \frac{r_8+r_0}{2}, & \text{if } WD_{\min} = WD_7 \\ \frac{r_6+r_2}{2}, & \text{if } WD_{\min} = WD_8 \end{cases} \quad (7)$$

If  $r_4$  is noisy, then the eight directional differences  $WD_1 \sim WD_8$  can be used to replace current detected pixel  $r_4$  using the minimum values of  $RV_1$  and  $RV_2$ . This operation is an extension of the method proposed by Chen and Lien [30]. Note that their system enables the removal of only fixed-value impulse noise, whereas our algorithm also enables the removal of random-value impulse noise. Fig. 2 and Fig. 3 respectively present the test images Lena and Baboon with added mixed impulse noise (fixed-value and random-valued impulse noise) and the corresponding images restored using the proposed scheme. Here, we obtained test images with various noise densities (20% to 40%) by randomly adding mixed impulse noise to the original image.

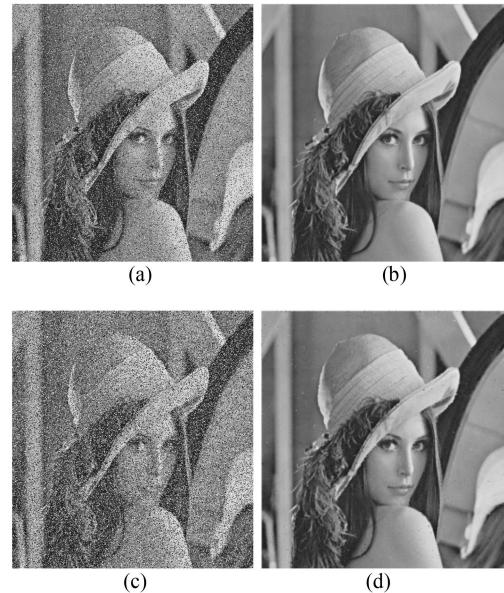


Fig. 2. Noisy images and restorations obtained using the proposed scheme for the original image Lena. (a) Noise of 20%. (b) Restored image of (a). (c) Noise of 40%. (d) Restored image of (c).

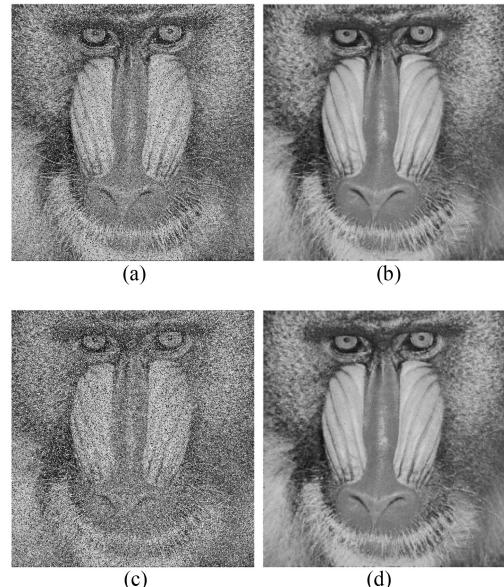


Fig. 3. Noisy images and restorations obtained using the proposed scheme for the original image Baboon. (a) Noise of 20%. (b) Restored image of (a). (c) Noise of 40%. (d) Restored image of (c).

### III. PROPOSED IMAGE DENOISING ARCHITECTURE

Many software implementations of image denoising algorithms are ill-suited to real-time applications, due to excessive computational overhead. These situations require dedicated hardware. Fig. 4 presents a block diagram showing the proposed image denoising architecture based on the algorithm described above. The proposed architecture comprises five modules, including an input buffer with sliding window, a 6-direction difference calculator, a reference value calculator, a noise detector, and a denoising circuit. The functions of the modules are respectively detailed below.

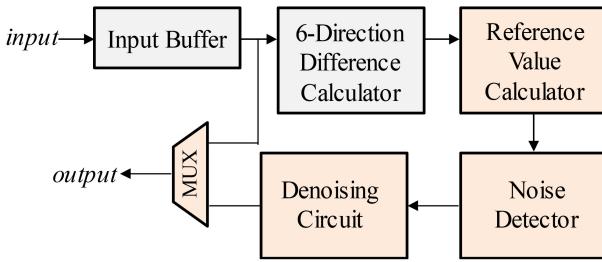
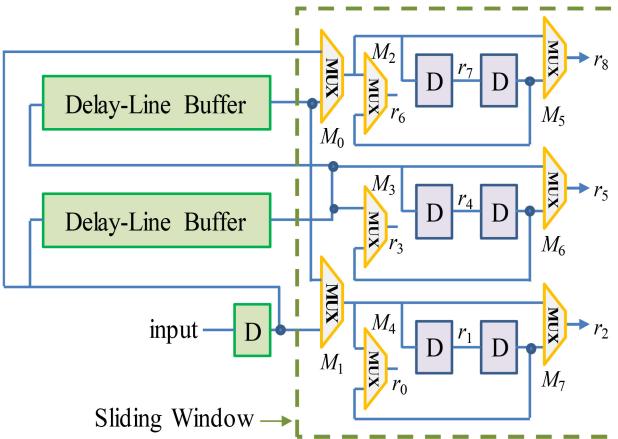


Fig. 4. Block diagram of the proposed image denoising filter.

Fig. 5. Input buffer module with a  $3 \times 3$  mask generator.

### A. Input Buffer Module

In Fig. 5, the input buffer module, which includes a  $3 \times 3$  mask generator (enclosed by the dashed rectangle), is responsible for creating a  $3 \times 3$  mask for the subsequent detection of noise. The input buffer uses two delay-line buffers and a register to maintain the three latest rows of pixels for  $3 \times 3$  filtering. The size of the delay-line buffer is equal to the size of the image measured in rows; however, we used pixel extension based on the symmetrization to improve image quality. The proposed  $3 \times 3$  mask generator employs eight multiplexers to achieve symmetrization. Here, the nine outputs ( $r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7$ , and  $r_8$ ) match the nine pixels in Fig. 1.

The  $3 \times 3$  mask generator with row-wise pixel symmetrization is implemented via switching multiplexers  $M_0$  and  $M_1$ , as shown in Fig. 6. The first and last three rows of pixels are produced by delay-line buffers using shift operations. Denoising every pixel in the first row of an image requires that the proposed input buffer duplicate the entire second row of pixels as the new first row of pixels in order to create a  $3 \times 3$  filter, as shown in Fig. 6(a). Similarly, obtaining the last three rows of the  $3 \times 3$  mask requires that the input buffer copy the previous second row of pixels (i.e., the first row of the  $3 \times 3$  mask) as the third row. Thus, the current input stream becomes the second row of pixels. The aforementioned structure is illustrated in Fig. 6(b). The remaining rows are not subjected to pixel symmetrization. Rather, a first-in-first-out scheme is used to create the three rows of pixels required for  $3 \times 3$  filtering.

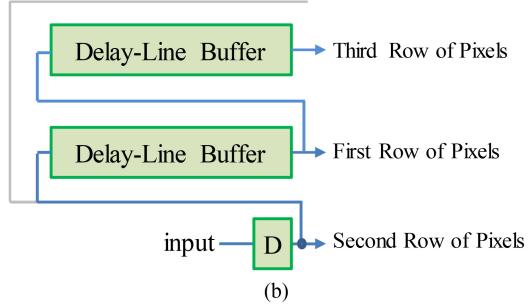
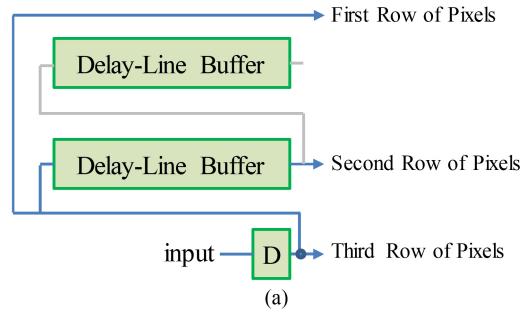
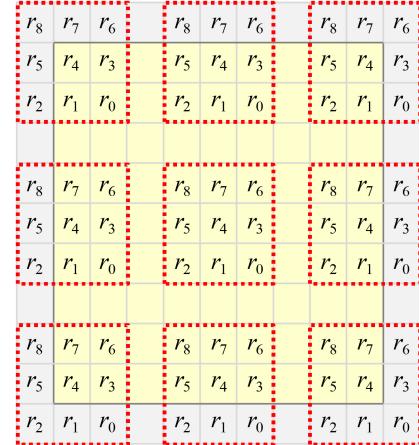


Fig. 6. Row-wise pixel symmetrization. (a) First three rows of pixels. (b) Last three rows of pixels produced by the delay-line buffers.

Fig. 7. Nine operating modes used for the  $3 \times 3$  mask.

The above pixel symmetrization scheme requires a different processing mode for each pixel in the image, as shown in Fig. 7. For example, dealing with the first pixel in the upper left corner of the image requires that we make  $r_2 = r_0$ ,  $r_5 = r_3$ ,  $r_8 = r_6$ ,  $r_7 = r_1$ , and  $r_6 = r_0$ . This means that the proposed  $3 \times 3$  mask scheme employs nine state operations (S01~S09), where S01, S03, S07, and S09 respectively correspond to operations performed in the four corners of the image (see Fig. 8).

The six modes on the left (S01, S04, S07) and right (S03, S06, S09) sides shown in Fig. 8 are used to generate column-wise symmetric pixels. Thus, the proposed input buffer module uses six corresponding multiplexers, as shown in Fig. 9. Fig. 9(a) presents the implementation of the three modes on the left side of Fig. 7, whereas Fig. 9(b) presents the implementation on the right side.

	1	2	3	...	$m-2$	$m-1$	$m$
1	S01	S02	S02	...	S02	S02	S03
2	S04	S05	S05	...	S05	S05	S06
3	S04	S05	S05	...	S05	S05	S06
4	S04	S05	S05	...	S05	S05	S06
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$n-3$	S04	S05	S05	...	S05	S05	S06
$n-2$	S04	S05	S05	...	S05	S05	S06
$n-1$	S04	S05	S05	...	S05	S05	S06
$n$	S07	S08	S08	...	S08	S08	S09

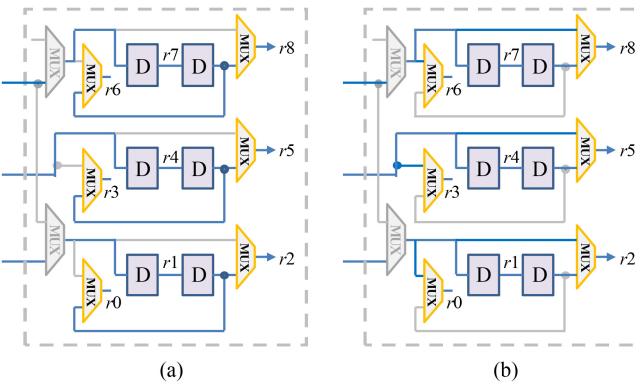
Fig. 8. Nine state operations for all pixels in an  $n \times m$  image.

Fig. 9. Generation circuit used for symmetric pixels. (a) Implementation for states (S01, S04, S07). (b) Implementation for states (S03, S06, S09).

Note that the proposed scheme based on a  $3 \times 3$  mask does not require additional clock cycles or additional memory to process the additional symmetric pixels. Furthermore, the pixel symmetrization mechanism is automated and the input buffer is applicable to images of any size.

### B. 6-Direction Difference Calculator

The proposed 6-direction difference calculator based on Eq. (2) is detailed in Fig. 10. The absolute operation  $|x|$  requires a non-negative output value; therefore, we adopted the two's complement scheme shown in Fig. 11 to obtain positive values in cases where the input is negative. This eliminates the need for subtractors or multipliers as well as the corresponding hardware cost and power consumption. The operation  $<< n$  indicates a left shift by  $n$  bits, which is implemented by a hardwired barrel left shifter. The resulting 6-direction differences are advanced to the input to identify the minimum value circuit.

Fig. 12 presents a block diagram showing the process by which the minimum value circuit is found based on the bubble sort algorithm, where the input data  $u_0 \sim u_5$  are outputs from the previous stage. The circuit employs five comparators, the output of which is generated by taking the smaller value after comparing the two values. This comparison process is implemented repeatedly to obtain the smallest of the 6 input values.

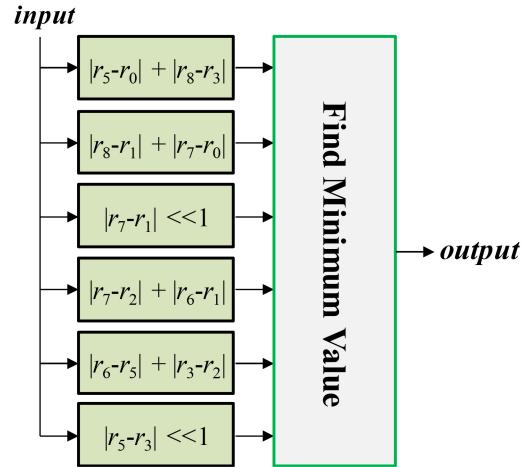


Fig. 10. 6-direction difference calculator.

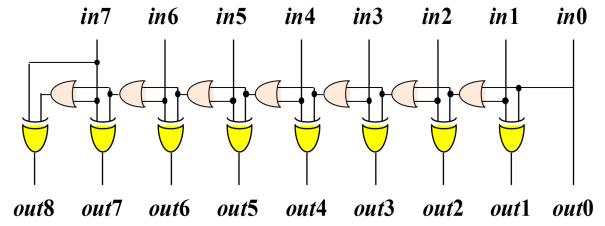


Fig. 11. Two's complement scheme.

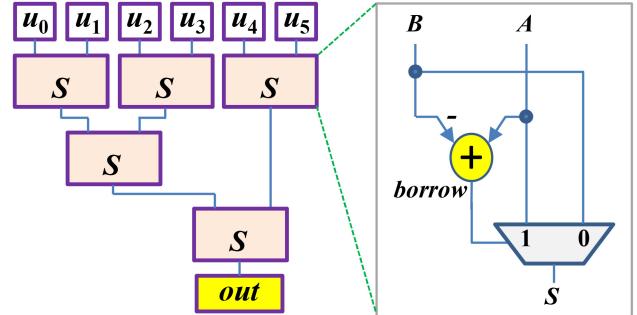


Fig. 12. Identifying the minimum value circuit.

### C. Noise Detection and Denoising Circuit Modules

The operation of the noise detection module is based on Eqs. (5) and (6). The absolute operation also employs the two's complement scheme to change negative values into positive values. The first step in the denoising circuit involves determining whether current pixel  $r_4$  is mixed impulse noise through the use of multiplexers and comparators. Some of the directional differences in Eqs. (4) and (6) are disregarded when one of the undetected pixels ( $r_0, r_1, r_2$ , and  $r_3$ ) is identified as noisy. This part of the circuit also uses multiplexers and comparators. The smallest value among the eight directional differences is identified using the bubble-sort circuit, which is similar to the 6-directional difference calculator.

## IV. REALIZATION OF PROPOSED ARCHITECTURE

The efficacy of the proposed architecture was evaluated by implementing it in an image denoising system. The system makes it possible to visualize image denoising results on

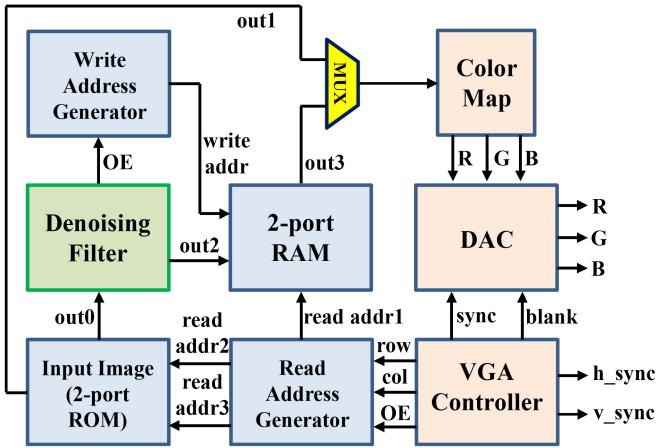


Fig. 13. Block diagram of the image denoising system.

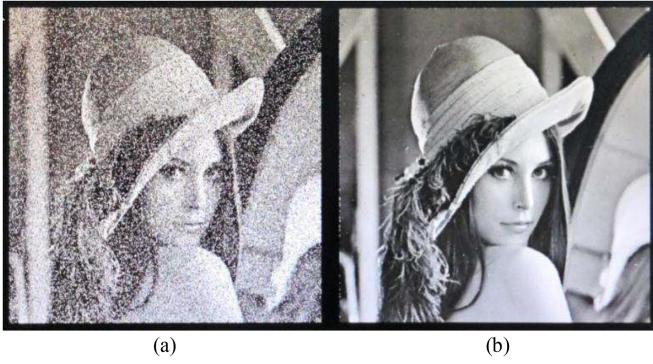


Fig. 14. Screen capture from computer screen showing VGA output from the denoising system. (a) 30% mixed impulse noise. (b) Restoration of image in (a).

screen, as shown in Fig. 13. Here, a  $512 \times 512$  test input image with 30% mixed impulse noise was stored in 2-port read-only memory (ROM). The ROM had two outputs, one of which was used as an input for the proposed system, and the other was forwarded to a video graphics array (VGA) controller to enable an on-screen comparison of results. The proposed system sent the denoised image to a 2-port random access memory (RAM) module, which outputs the image to the color map memory. Note that a read address generator is responsible for obtaining the output of the test ROM and a write address generator is responsible for maintaining the data required to store the output in the resulting RAM space. The proposed system also included a video signal output comprising color map memory, a digital to analog converter (DAC), and a VGA controller. The input for the monitor was from the VGA output. Screen resolution was  $1152 \times 864$  pixels, based on a phase-locked loop (PLL) running at 108 MHz. Fig. 14 presents a screen capture of the of a computer screen showing the results of the proposed image denoising scheme.

The application target in the current study was the dashboard camera used in cars. Most dashboard cameras are able to produce high-quality video images under good illumination conditions; however, the video images produced under low-light conditions are prone to corruption from impulse noise. Fig. 15 presents the proposed method used for the removal of noise from dashboard cameras. Fig. 15(a) and

TABLE I  
COMPARISON OF THREE ALGORITHMS APPLIED  
TO LENA IMAGE IN PSNR (DB)

Noise Density	Lien et al. [59]		Lien et al. [32]		This Work	
	SSIM	MSSSIM	SSIM	MSSSIM	SSIM	MSSSIM
10 %	0.97	0.99	0.97	0.99	0.97	0.99
20 %	0.94	0.98	0.94	0.98	0.96	0.98
30 %	0.89	0.96	0.90	0.97	0.94	0.97
40 %	0.82	0.93	0.88	0.96	0.91	0.96
50 %	0.72	0.87	0.83	0.94	0.88	0.93
60 %	0.9	0.78	0.79	0.90	0.82	0.90
70 %	0.46	0.65	0.75	0.84	0.76	0.84
80 %	0.33	0.52	0.69	0.76	0.67	0.75
90 %	0.24	0.39	0.60	0.66	0.59	0.67

TABLE II  
COMPARISON OF THREE ALGORITHMS APPLIED  
TO BABOON IMAGE IN PSNR(DB)

Noise Density	Lien et al. [59]		Lien et al. [32]		This Work	
	SSIM	MSSSIM	SSIM	MSSSIM	SSIM	MSSSIM
10 %	0.84	0.91	0.74	0.91	0.84	0.92
20 %	0.81	0.89	0.71	0.88	0.81	0.90
30 %	0.76	0.87	0.67	0.86	0.77	0.88
40 %	0.70	0.83	0.63	0.82	0.73	0.85
50 %	0.62	0.78	0.59	0.78	0.68	0.81
60 %	0.53	0.71	0.54	0.74	0.62	0.76
70 %	0.43	0.60	0.50	0.67	0.56	0.70
80 %	0.34	0.49	0.45	0.60	0.49	0.63
90 %	0.24	0.37	0.39	0.53	0.42	0.55

Fig. 15(c) respectively present two original (i.e., noisy) images captured from a video of the dashboard camera. Fig. 15(b) and Fig. 15(d) present the denoising results obtained using the proposed method for color image files rendered the YCbCr color space. Individual luminance processing is the same as the gray-scale image denoising mentioned earlier, but with an additional transformation of the RGB color space to YCbCr. The denoising results above demonstrate the efficacy of the proposed method in removing real impulse noise using an algorithm of low complexity.

## V. RESULTS AND PERFORMANCE COMPARISON

To ensure a fair comparison, our performance evaluation included only non-linear digital filters that employ similar methods (i.e., excluding bilateral filters). We adopted two well-known methods by which to objectively evaluate image quality, including the structural similarity index measure (SSIM) [64] and multiscale structural similarity index measure (MSSSIM) [65]. Table I and Table II respectively compare three denoising algorithms when applied to the Lena and Baboon images. Clearly, the proposed scheme outperformed the previous works in terms of image restoration, particularly when removing noise from highly detailed images, (e.g., the Baboon image). The outstanding performance of the proposed system when dealing with high noise densities can be attributed to the detection of mixed impulse noise for the current pixel based on the most probable edge (see Algorithm 1). Nonetheless, this approach was shown to affect restoration performance when dealing with images that lack detail.



Fig. 15. Noise removal from real-world noisy images captured using a dashboard camera. (a) Original noisy image (I). (b) Restoration of color image in (a). (c) Original noisy image (II). (d) Restoration of color image in (c).

TABLE III  
COMPARISON OF THREE DESIGNS

Parameters	Matsubara et al. [57]	Lien et al. [59]	This Work
Mask Size	3×3	3×3	3×3
FPGA/ASIC	FPGA	180 nm	FPGA 180 nm
Gate Counts/ logic cells	13	21 K	1709 29 K
Memory Cells	8297	N/A	- N/A
Clock Rate	130 MHz	200 MHz	140 MHz 200 MHz
Power	12.31 mW	17.12 mW	- 21.7 mW
Denoising Type	Random-Valued Impulse Noise	Fixed-Valued and Random-Valued Impulse Noise	Fixed-Valued and Random-Valued Impulse Noise
Pixel Extension	NO	NO	YES

N/A: Not Applicable -: no data

The design of an application specific integrated circuit (ASIC) for the proposed denoising system was implemented using Verilog hardware description language and synthesized using 0.18  $\mu\text{m}$  CMOS process technology. Here, the delay-line buffer used in the proposed system employed a two-port RAM module to simulate the shift function and thereby reduce area and power overhead.

Table III presents a comparison of three designs in terms of area and power consumption. As shown in Table III, the gate count of the proposed scheme using an ASIC exceeded that of the scheme presented by Lien *et al.* [59]; however, the image restoration performance of our scheme was superior. This can be attributed to the fact that our scheme is able to remove mixed impulse noise, which is a crucial factor when dealing with real impulse noise. The proposed design also provides automated pixel extension, which results in image quality superior to that of other similar schemes. Note that Table III lists only dynamic and static power consumption.

## VI. CONCLUSION

This paper presents a novel hardware approach to the removal of impulse noise from digital images captured using consumer-grade dashboard cameras. The proposed scheme proved superior to previous designs in terms of image restoration performance. The proposed image denoising scheme implemented using 180 nm CMOS technology with roughly 29 K gates consumed only 21.7 mW when operated at 200 MHz (5 ns). At this operating frequency, the proposed chip can process noisy video streams with 1920×1080 resolution at 60 frames per second in real time. The efficacy and feasibility of the proposed scheme was verified in experiments using a prototype system. Our image denoising system proved highly effective in removing real impulse noise from video images captured using a dashboard camera. The proposed scheme is also well suited to object

recognition tasks. Unlike conventional non-linear denoising architectures, the proposed system is able to remove mixed impulse noise, while preserving image borders to facilitate subsequent image processing. Finally, the low complexity of the system enables real-time operations using low-cost consumer-grade electronics (e.g., digital video recorders) with minimal memory capacity and low energy consumption.

### ACKNOWLEDGMENT

The authors would like to thank the Taiwan Semiconductor Research Institute, National Applied Research Laboratories for EDA tool support.

### REFERENCES

- [1] N. Hnoohom and T. Thanapattherakul, "Image problem classification for dashboard cameras," in *Proc. 12th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, Nov. 2016, pp. 673–678, doi: [10.1109/SITIS.2016.112](https://doi.org/10.1109/SITIS.2016.112).
- [2] J.-W. Wu, C.-C. Wu, W.-S. Cen, S.-A. Chao, and J.-T. Weng, "Integrated compressed sensing and YOLOv4 for application in image-storage and object-recognition of dashboard camera," in *Proc. Aust. New Zealand Control Conf.*, Nov. 2021, pp. 7–11, doi: [10.1109/ANZCC53563.2021.9628221](https://doi.org/10.1109/ANZCC53563.2021.9628221).
- [3] M. Altun and M. Celenk, "Road scene content analysis for driver assistance and autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3398–3407, Dec. 2017, doi: [10.1109/TITS.2017.2688352](https://doi.org/10.1109/TITS.2017.2688352).
- [4] K. Okumura, S. Takahashi, and T. Hagiwara, "An estimation method of road narrowing condition in in-vehicle camera," in *Proc. IEEE 9th Global Conf. Consum. Electron. (GCCE)*, Kobe, Japan, 2020, pp. 733–734, doi: [10.1109/GCCE50665.2020.9291982](https://doi.org/10.1109/GCCE50665.2020.9291982).
- [5] T. Takase *et al.*, "An estimation method of road surface condition on winter expressway via mobile nets using in-vehicle camera images," in *Proc. IEEE 9th Global Conf. Consum. Electron. (GCCE)*, Kyoto, Japan, Oct. 2021, pp. 109–110, doi: [10.1109/GCCE53005.2021.9621866](https://doi.org/10.1109/GCCE53005.2021.9621866).
- [6] M.-T. Yang and J.-Y. Zheng, "On-road collision warning based on multiple FOE segmentation using a dashboard camera," *IEEE Trans. Veh. Technol.*, vol. 64, no. 11, pp. 4974–4984, Nov. 2015, doi: [10.1109/TVT.2014.2378373](https://doi.org/10.1109/TVT.2014.2378373).
- [7] Y. Wang and J. Kato, "Collision risk rating of traffic scene from dashboard cameras," in *Proc. Int. Conf. Digit. Image Comput. Techn. Appl.*, Nov. 2017, pp. 1–6, doi: [10.1109/DICTA.2017.8227417](https://doi.org/10.1109/DICTA.2017.8227417).
- [8] Y.-K. Lai, Y.-H. Huang, and T. Schumann, "Intelligent vehicle collision warning system based on a deep learning approach," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE-TW)*, Taichung, Taiwan, May 2018, pp. 1–2, doi: [10.1109/ICCE-China.2018.8448556](https://doi.org/10.1109/ICCE-China.2018.8448556).
- [9] R. Ogura and T. Nagasaki, "Improve visibility of nighttime images for pedestrian recognition by in-vehicle camera," in *Proc. IEEE 9th Global Conf. Consum. Electron. (GCCE)*, Nara, Japan, Oct. 2018, pp. 51–54, doi: [10.1109/GCCE.2018.8574702](https://doi.org/10.1109/GCCE.2018.8574702).
- [10] Y. Kawashita and T. Wada, "Pedestrian-vehicle collision avoidance support system corresponding to multiple pedestrians," in *Proc. IEEE 10th Global Conf. Consum. Electron. (GCCE)*, Kyoto, Japan, Oct. 2021, pp. 312–313, doi: [10.1109/GCCE53005.2021.9621800](https://doi.org/10.1109/GCCE53005.2021.9621800).
- [11] G. Madhumitha, R. Senthilnathan, K. M. Ayaz, J. Vignesh, and K. Madhu, "Estimation of collision priority on traffic videos using deep learning," in *Proc. IEEE Int. Conf. Mach. Learn. Appl. Netw. Techn.*, Dec. 2020, pp. 1–6, doi: [10.1109/ICMLANT50963.2020.9355992](https://doi.org/10.1109/ICMLANT50963.2020.9355992).
- [12] H. Kim, Y. Lee, B. Yim, E. Park, and H. Kim, "On-road object detection using deep neural network," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE-Asia)*, Oct. 2016, pp. 1–4, doi: [10.1109/ICCE-Asia.2016.7804765](https://doi.org/10.1109/ICCE-Asia.2016.7804765).
- [13] M. Marković, M. Milošević, and N. Jovanov, "Using class-dependent post-processing to improve ai for automotive front view camera," in *Proc. IEEE 5th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Berlin, Germany, Sep. 2019, pp. 211–214, doi: [10.1109/ICCE-Berlin47944.2019.8966233](https://doi.org/10.1109/ICCE-Berlin47944.2019.8966233).
- [14] M. Mody *et al.*, "Image signal processing for front camera based automated driver assistance system," in *Proc. IEEE 5th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Berlin, Germany, Sep. 2015, pp. 158–159, doi: [10.1109/ICCE-Berlin.2015.7391221](https://doi.org/10.1109/ICCE-Berlin.2015.7391221).
- [15] J. Janjatovic, R. Ostojic, G. Velikic, and T. Maruna, "Algorithm optimization framework for advanced driver assistance systems," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2017, pp. 275–276, doi: [10.1109/ICCE.2017.7889316](https://doi.org/10.1109/ICCE.2017.7889316).
- [16] A. Gupta and A. Choudhary, "Real-time lane detection using spatio-temporal incremental clustering," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, Oct. 2017, pp. 1–6, doi: [10.1109/ITSC.2017.8317827](https://doi.org/10.1109/ITSC.2017.8317827).
- [17] H. Kim, Y. Lee, T. Woo, and H. Kim, "Integration of vehicle and lane detection for forward collision warning system," in *Proc. IEEE 11th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Berlin, Germany, Sep. 2016, pp. 5–8, doi: [10.1109/ICCE-Berlin.2016.7684703](https://doi.org/10.1109/ICCE-Berlin.2016.7684703).
- [18] D. Vajak, M. Vranješ, R. Grbić, and N. Teslić, "A rethinking of real-time computer vision-based lane detection," in *Proc. IEEE 11th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Berlin, Germany, Nov. 2021, pp. 1–6, doi: [10.1109/ICCE-Berlin53567.2021.9720012](https://doi.org/10.1109/ICCE-Berlin53567.2021.9720012).
- [19] N. Hashimoto, "Evaluation experiments of lane markers for lane detection systems under several conditions: Differences between all-weather lane marker and ordinary lane markers under several weather conditions and several times of day," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2021, pp. 1–6, doi: [10.1109/ICCE50685.2021.9427647](https://doi.org/10.1109/ICCE50685.2021.9427647).
- [20] V. Gojak, J. Janjatovic, N. Vukota, M. Milosevic, and M. Z. Bjelica, "Informational bird's eye view system for parking assistance," in *Proc. IEEE 7th Int. Conf. Consum. Electron. (ICCE)*, Berlin, Germany, Sep. 2017, pp. 103–104, doi: [10.1109/ICCE-Berlin.2017.8210604](https://doi.org/10.1109/ICCE-Berlin.2017.8210604).
- [21] K. M. Jeong and B. C. Song, "Night time vehicle detection using rear-lamp intensity," in *Proc. Int. Conf. Consum. Electron. (ICCE-Asia)*, Oct. 2016, pp. 1–3, doi: [10.1109/ICCE-Asia.2016.7804770](https://doi.org/10.1109/ICCE-Asia.2016.7804770).
- [22] S. Yoshizawa and Y. Nakatoh, "Blind vehicle's sound detecting technique for advanced safety-driving system," in *Int. Conf. Consum. Electron. Dig. Technical Papers*, Jan. 2009, pp. 1–2, doi: [10.1109/ICCE.2009.5012169](https://doi.org/10.1109/ICCE.2009.5012169).
- [23] K. Yi and K. H. Jung, "A fast and accurate forward vehicle start alarm by tracking moving edges obtained from dashboard camera," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveillance*, Nov. 2018, pp. 1–6, doi: [10.1109/AVSS.2018.8639162](https://doi.org/10.1109/AVSS.2018.8639162).
- [24] A. Gupta and A. Choudhary, "A framework for real-time traffic sign detection and recognition using Grassmann manifolds," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, Nov. 2018, pp. 274–279, doi: [10.1109/ITSC.2018.8569556](https://doi.org/10.1109/ITSC.2018.8569556).
- [25] A. D. Abadi, Y. Gu, I. Goncharenko, and S. Kamijo, "Detection of cyclists' crossing intentions for autonomous vehicles," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2022, pp. 1–6, doi: [10.1109/ICCE53296.2022.9730559](https://doi.org/10.1109/ICCE53296.2022.9730559).
- [26] Y.-S. Poon, C.-C. Lin, Y.-H. Liu, and C.-P. Fan, "YOLO-based deep learning design for in-cabin monitoring system with fisheye-lens camera," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2022, pp. 1–4, doi: [10.1109/ICCE53296.2022.9730235](https://doi.org/10.1109/ICCE53296.2022.9730235).
- [27] Y. Hara, A. Uchiyama, T. Umeda, and T. Higashino, "Sidewalk-level people flow estimation using dashboard cameras based on deep learning," in *Proc. 11th Int. Conf. Mobile Comput. Ubiquit. Netw.*, Oct. 2018, pp. 1–6, doi: [10.23919/ICMU.2018.8653595](https://doi.org/10.23919/ICMU.2018.8653595).
- [28] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2019, pp. 273–280, doi: [10.1109/IROS40897.2019.8967556](https://doi.org/10.1109/IROS40897.2019.8967556).
- [29] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision based algorithm for removal of high-density impulse noises," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 189–192, Mar. 2007, doi: [10.1109/LSP.2006.884018](https://doi.org/10.1109/LSP.2006.884018).
- [30] P.-Y. Chen and C.-Y. Lien, "An efficient edge-preserving algorithm for removal of salt-and-pepper noise," *IEEE Signal Process. Lett.*, vol. 15, pp. 833–836, Dec. 2008, doi: [10.1109/LSP.2008.2005047](https://doi.org/10.1109/LSP.2008.2005047).
- [31] S. N. Sulaiman and N. A. Mat Isa, "Denoising-based clustering algorithms for segmentation of low level salt-and-pepper noise-corrupted images," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2702–2710, Nov. 2010, doi: [10.1109/TCE.2010.5681159](https://doi.org/10.1109/TCE.2010.5681159).
- [32] C. C. Lien, C. C. Huang, P. Y. Chen, and H. Y. Yang, "An efficient denoising approach for random-valued impulse noise in digital images," in *Proc. Int. Conf. Innov. Bio-Inspired Comput. Appl.*, Dec. 2011, pp. 13–16, doi: [10.1109/IBICA.2011.8](https://doi.org/10.1109/IBICA.2011.8).
- [33] T. Thaipanich, B. T. Oh, P. Wu, D. Xu, and C.-J. Kuo, "Improved image denoising with adaptive nonlocal means (ANL-means) algorithm," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2623–2630, Nov. 2010, doi: [10.1109/TCE.2010.5681149](https://doi.org/10.1109/TCE.2010.5681149).

- [34] H. Yu, L. Zhao, and H. Wang, "An efficient edge-based bilateral filter for restoring real noisy image," *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 682–686, May 2011, doi: [10.1109/TCE.2011.5955208](#).
- [35] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, pp. 2080–2095, 2007, doi: [10.1109/TIP.2007.901238](#).
- [36] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2007, pp. 313–316, doi: [10.1109/ICIP.2007.4378954](#).
- [37] R. D. da Silva, R. Minetto, W. R. Schwartz, and H. Pedrini, "Adaptive edge preserving image denoising using wavelet transforms," *Pattern Anal. Appl.*, vol. 16, no. 4, pp. 567–580, Nov. 2013.
- [38] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, pp. 1620–1630, 2013, doi: [10.1109/TIP.2012.2235847](#).
- [39] J. Xu, L. Zhang, and D. Zhang, "External prior guided internal prior learning for real-world noisy image denoising," *IEEE Trans. Image Process.*, vol. 27, pp. 2996–3010, 2018, doi: [10.1109/TIP.2018.2811546](#).
- [40] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, pp. 3142–3155, 2017, doi: [10.1109/TIP.2017.2662206](#).
- [41] Y. Kim, J. W. Soh, G. Y. Park, and N. I. Cho, "Transfer learning from synthetic to real-noise denoising with adaptive instance normalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 3479–3489, doi: [10.1109/CVPR42600.2020.00354](#).
- [42] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164, doi: [10.1109/CVPR.2018.00333](#).
- [43] H. S. Park, J. Baek, S. K. You, J. K. Choi, and J. K. Seo, "Unpaired image denoising using a generative adversarial network in X-ray CT," *IEEE Access*, vol. 7, pp. 110414–110425, 2019, doi: [10.1109/ACCESS.2019.2934178](#).
- [44] N. Moran, D. Schmidt, Y. Zhong, and P. Coady, "Noisier2Noise: Learning to denoise from unpaired noisy data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12061–12069, doi: [10.1109/CVPR42600.2020.01208](#).
- [45] X. Wu, M. Liu, Y. Cao, D. Ren, and W. Zuo, "Unpaired learning of deep image denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 352–368.
- [46] K. Lee and W.-K. Jeong, "ISCL: Interdependent self-cooperative learning for unpaired image denoising," *IEEE Trans. Med. Imag.*, vol. 40, no. 11, pp. 3238–3248, Nov. 2021, doi: [10.1109/TMI.2021.3096142](#).
- [47] S. Soltanayev and S. Y. Chun, "Training deep learning based denoisers without ground truth data," 2018, *arXiv:1803.01314*.
- [48] J. Batson and L. Royer, "Noise2Self: Blind denoising by self-supervision," in *Proc. Int. Conf. Mach. Learn.*, Mar. 2019, pp. 524–533.
- [49] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2Void-learning denoising from single noisy images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2129–2137, doi: [10.1109/CVPR.2019.00223](#).
- [50] C. Niu, F. Fan, Q. Lyu, and G. Wang, "Noise2Sim-similarity-based self-learning for image denoising," 2020, *arXiv:2011.03384*.
- [51] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2Self with dropout: Learning self-supervised denoising from single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1887–1895, doi: [10.1109/CVPR42600.2020.00196](#).
- [52] J. Xu *et al.*, "Noisy-as-Clean: Learning self-supervised denoising from corrupted image," *IEEE Trans. Image Process.*, vol. 29, pp. 9316–9329, 2020, doi: [10.1109/TIP.2020.3026622](#).
- [53] K. Lee and W.-K. Jeong, "Noise2Kernel: Adaptive self-supervised blind denoising using a dilated convolutional kernel architecture," 2020, *arXiv:2012.03623*.
- [54] V. Dewil, J. Anger, A. Davy, T. Ehret, G. Facciolo, and P. Arias, "Self-supervised training for blind multi-frame video denoising," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 2723–2733, doi: [10.1109/WACV48630.2021.00277](#).
- [55] J. Lehtinen *et al.*, "Noise2Noise: Learning image restoration without clean data," 2018, *arXiv:1803.04189v3*.
- [56] H. Dutta, F. Hannig, J. Teich, B. Heigl, and H. Hornegger, "A design methodology for hardware acceleration of adaptive filter algorithms in image processing," in *Proc. IEEE 17th Int. Conf. Appl.-Specif. Syst. Archit. Processors (ASAP)*, Sep. 2006, pp. 331–340, doi: [10.1109/ASAP.2006.4](#).
- [57] T. Matsubara, V. G. Moshnyaga, and K. Hashimoto, "A FPGA implementation of low-complexity noise removal," in *Proc. 17th IEEE Int. Conf. Electron. Circuits Syst.*, pp. 255–258, Dec. 2010, doi: [10.1109/ICECS.2010.5724502](#).
- [58] 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 Integr. (VLSI) Syst.*, vol. 18, no. 3, pp. 473–481, Mar. 2010, doi: [10.1109/TVLSI.2008.2012263](#).
- [59] C.-Y. Lien, C.-C. Huang, P.-Y. Chen, and Y.-F. Lin, "An efficient denoising architecture for removal of impulse noise in images," *IEEE Trans. Comput.*, vol. 62, no. 4, pp. 631–643, Apr. 2013, doi: [10.1109/TC.2011.256](#).
- [60] A. Gabiger-Rose, M. Kube, R. Weigel, and R. Rose, "An FPGA-based fully synchronized design of a bilateral filter for real-time image denoising," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4093–4104, Aug. 2014, doi: [10.1109/TIE.2013.2284133](#).
- [61] C. S. Chandni and R. Pushpakumari, "Reduced hardware architecture of bilateral filter for real time image denoising," in *Proc. Int. Conf. Intell. Comput. Instrum. Control Technol.*, Jul. 2017, pp. 769–774, doi: [10.1109/ICICICT1.2017.8342661](#).
- [62] C.-Y. Lien, C.-H. Tang, P.-Y. Chen, Y.-T. Kuo, and Y.-L. Deng, "A low-cost VLSI architecture of the bilateral filter for real-time image denoising," *IEEE Access*, vol. 8, pp. 64278–64283, 2020, doi: [10.1109/ACCESS.2020.2984688](#).
- [63] L.-Z. Hou, C.-F. Kuo, Z.-H. Lin, and C. Yu, "Efficient image denoising scheme for removal of impulse noise," in *Proc. IEEE Global Conf. Consum. Electron. (GCCE)*, Osaka, Japan, Oct. 2015, pp. 298–299, doi: [10.1109/GCCE.2015.7398612](#).
- [64] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, 2004, doi: [10.1109/TIP.2003.819861](#).
- [65] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals Syst. Comput.*, vol. 2, Nov. 2003, pp. 1398–1402, doi: [10.1109/ACSSC.2003.1292216](#).



**Chu Yu** (Member, IEEE) received the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999. Since 2000, he has been a member of the Faculty with the Department of Electronic Engineering, National Ilan University (NIU), where he is a Full Professor. He served as the Dean of the Office of Research and Development, NIU from 2017 to 2020. He was a General Chair of 2019 IEEE International Conference on Consumer Electronics—Taiwan. He is currently the Dean of the College of Electrical Engineering and Computer Science. His research interests include IC design for digital communications and digital signal processing, embedded systems, and Internet of Things. He is a member of the IEEE Consumer Electronics Society and the IEEE Circuits and Systems Society.



**Li-Zhong Hou** received the B.S. degree in microelectronics from Xidian University, Xi'an, China, in 2013, and the M.S. degree in electronic engineering from National Ilan University, Yilan, Taiwan, in 2015. His research interests include digital signal and image processing.