

Letter

Noise-Aware and Light-Weight VLSI Design of Bilateral Filter for Robust and Fast Image Denoising in Mobile Systems

Sung-Joon Jang ¹ and Youngbae Hwang ^{2,*}

¹ Intelligent Image Processing Research Center, Korea Electronics Technology Institute, Seongnam 13509, Korea; sjjang0626@keti.re.kr

² Department of Electronics Engineering, Chungbuk National University, Cheongju 28644, Korea

* Correspondence: ybhwang@cbnu.ac.kr

Received: 20 July 2020; Accepted: 19 August 2020; Published: 21 August 2020



Abstract: The range kernel of bilateral filter degrades image quality unintentionally in real environments because the pixel intensity varies randomly due to the noise that is generated in image sensors. Furthermore, the range kernel increases the complexity due to the comparisons with neighboring pixels and the multiplications with the corresponding weights. In this paper, we propose a noise-aware range kernel, which estimates noise using an intensity difference-based image noise model and dynamically adjusts weights according to the estimated noise, in order to alleviate the quality degradation of bilateral filters by noise. In addition, to significantly reduce the complexity, an approximation scheme is introduced, which converts the proposed noise-aware range kernel into a binary kernel while using the statistical hypothesis test method. Finally, blue a fully parallelized and pipelined very-large-scale integration (VLSI) architecture of a noise-aware bilateral filter (NABF) that is based on the proposed binary range kernel is presented, which was successfully implemented in field-programmable gate array (FPGA). The experimental results show that the proposed NABF is more robust to noise than the conventional bilateral filter under various noise conditions. Furthermore, the proposed VLSI design of the NABF achieves 10.5 and 95.7 times higher throughput and uses 63.6–97.5% less internal memory than state-of-the-art bilateral filter designs.

Keywords: image denoising; bilateral filter; range kernel; image noise model; very-large-scale integration (VLSI); field-programmable gate array (FPGA)

1. Introduction

Image denoising methods have been rapidly evolved as a critical component of image processing pipeline [1] and high-level vision tasks [2,3]. Recently introduced methods, such as improved block-matching and three-dimensional (3D) filtering (BM3D) [4] and deep learning-based algorithms [5,6], reduce a noise significantly under various environments. However, these methods are infeasible for battery-powered mobile systems, in which the low-power operation is essential, owing to high complexity requiring high-end central processing units (CPUs) or graphics processing units (GPUs). Alternatively, the bilateral filter [7] has been started to adopt for mobile systems [8,9], owing to relatively low complexity and edge-preserving characteristic. Furthermore, very-large-scale integration (VLSI) designs for real-time and low-power bilateral filtering in mobile systems have been recently presented [10–13].

However, the bilateral filter is problematic to use practically owing to image quality degradation by noise. This is because the pixel intensity is changed randomly by the noise that is generated in image sensors [14], but the range kernel cannot distinguish whether this variation is caused by noise or not. Furthermore, a non-optimal parameter defined by a user can degrade the quality of filtered image.

In addition, VLSI designs of the bilateral filter suffer from increased complexity due to the range kernel. The range kernel is typically implemented using a look-up table (LUT) memory, which stores pre-defined weights according to the difference in pixel intensity. Because differences in pixel intensity are random, separate LUT memories are required for the parallel processing of pixels within a pixel window, and this increases the resource use in proportion to the pixel window size.

In this paper, we propose a noise-aware bilateral filter (NABF) to resolve the aforementioned image quality degradation problem, which replaces the conventional range kernel with an intensity difference-based image noise model [15] to estimate the probability that an intensity difference is caused by noise and adjusts the weight dynamically according to the estimated noise probability. A user-defined parameter that affects the quality of the bilateral filter is also eliminated by this replacement. In addition, the proposed range kernel is approximated into a binary kernel using the statistical hypothesis test method to reduce the usage of the LUT memory and the logic considerably in a VLSI design [16]. Finally, a fully parallelized and pipelined VLSI of the NABF based on the proposed binary range kernel is designed, which was successfully implemented in a field-programmable gate array (FPGA). The experimental results demonstrate that the proposed NABF and its approximated one are more robust to noise than the conventional bilateral filter under various noise conditions. Furthermore, the proposed VLSI design achieves 10.5 and 95.7 times higher throughput and uses 63.6–97.5% less internal memory when compared with recent VLSI designs of the bilateral filter.

The rest of this paper is organized, as follows. Related studies are presented in Section 2. Section 3 presents the NABF and its approximation method. Section 4 presents the VLSI architecture of the NABF. Experimental results are shown in Section 5. Lastly, Section 6 concludes the paper.

2. Related Works

2.1. Optimal Parameter Selection of Bilateral Filter

Because ground truth images do not exist in the real environment, statistical approaches, such as Stein's unbiased risk estimator (SURE), Poisson unbiased risk estimator (PURE), and Chi-Square unbiased risk estimator (CURE), are adopted for optimal parameter selection of the bilateral filter [17–19]. SURE, PURE, and CURE use Gaussian noise, Poisson, and Chi-Square noise distribution, respectively. Furthermore, the output of bilateral filter is adopted as an estimate of the noiseless ground truth image, and unbiased estimated risk is modeled. Finally, the values of the parameters are selected by minimization of the estimated risk. Despite this, usefulness cannot be adopted for real-time mobile systems owing to high complexity. For example, the CURE requires dozens of minutes of execution time in a PC [20]. Recently, a training method that is based on neural network is presented [21]. Image texture features from training images and their optimal parameter are used for inputs of training model, and peak signal-to-noise ratio (PSNR) is adopted as a loss function. Furthermore, the two-dimensional (2D) wavelet transform and gray-level co-occurrence matrix are adopted to extract image texture features. Finally, a forward path of trained neural network is used as predictive model of optimal parameter for bilateral filtering. However, this method also requires a high-end CPU or GPU, owing to high complexity caused by neural network and feature extraction.

2.2. VLSI Design of Bilateral Filter

A VLSI design based on the integral histogram-based bilateral filter [22] is presented in order to process a large size of window in real-time [10]. However, their design requires many internal memories to store the histogram. Moreover, external memory access, which causes large power consumption and delay in mobile systems, is needed. In [11,12], the equation of bilateral filter is implemented to parallelized VLSI designs without modification. In particular, Gabiger-Rose et al. introduced a resource-efficient approach that divides pixels within the window into multiple groups and assigns each group to the separate pipeline [12]. Although this approach reduces the usage of resources, the processing speed is degraded and the large lookup memory to store weights is required.

In [13], a VLSI that is based on a fast bilateral filtering method using an approximation [23] is designed to support arbitrary size of the window without the increase of resources. However, the speed is low owing to its iterative nature.

3. Proposed Approach

In order to preserve the details of images, the range kernel (f_r) of a bilateral filter controls the degree of smoothing by the spatial kernel (f_s) using the intensity difference between two pixels:

$$I_{BF}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) \cdot f_s \cdot f_r, \quad (1)$$

where $f_s = \exp(-\frac{|x_i - x|^2}{2\sigma_s^2})$, $f_r = \exp(-\frac{|I(x_i) - I(x)|^2}{2\sigma_r^2})$.

Here, $I(x)$ and $I_{BF}(x)$ are the pixel intensity in position x , and its filtered intensity, respectively, and Ω is a pixel window. In addition, $x_i \in \Omega$ are neighboring pixels of x within a pixel window; f_s and f_r are the spatial and range kernel, respectively; σ_s and σ_r are standard deviations of f_s and f_r , respectively, and W_p is a normalization term. However, $I(x)$ and $I(x_i)$ can be changed randomly by the noise generated in image sensors [14], e.g., shot noise, but the range kernel (f_r) cannot recognize whether this change is caused by noise or not. A noise-aware bilateral filter is proposed to resolve this issue.

3.1. Noise-Aware Bilateral Filter (NABF)

An image noise model is adapted for estimating the noise at each pixel. In particular, the image noise model [15] that is based on the intensity difference between two pixels is adopted among various image noise models to determine whether the intensity difference, which is the input of the range kernel, is caused by noise or not:

$$p(k; \mu) = \exp^{-2\mu} \cdot B_k \cdot 2\mu. \quad (2)$$

Here, k is the intensity difference between two pixels; $p(k)$ and μ are the noise probability function and its parameter, respectively; and, $B_k(\cdot)$ is the modified Bessel function of the first kind as:

$$B_k(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(M + k + 1)} \left(\frac{x}{2}\right)^{2m+k}. \quad (3)$$

The probability that an intensity difference is caused by noise is modeled, as shown in Figure 1a. This noise probability function is dependent on μ , as shown in Figure 1b. Furthermore, μ varies proportionally with the pixel intensity:

$$\mu = f(I) = c_0 \cdot I + c_1, \quad (4)$$

where I is a pixel intensity, and c_0 and c_1 are constants due to a camera setting such as a camera gain. By capturing the colorchecker board, there is the linear relationship between intensity and noise parameters for each homogeneous color patch [15]. From the line, the values of c_0 and c_1 can be determined by the fitting process. For details that are related to equations and their derivation, refer to the literatures [15].

The range kernel of (1) with the above difference-based image noise model is replaced, as expressed in (5), and the degree of smoothing by the spatial kernel is controlled according to the noise probability of intensity difference:

$$f_r^{NABF} = \exp^{-2\mu} \cdot B_{|I(x_i) - I(x)|} \cdot 2\mu, \quad (5)$$

$$\text{where } \mu = c_0 \cdot I(x) + c_1.$$

Here, f_r^{NABF} is the noise-aware range kernel, and $I(x)$ and $I(x_i)$ are the intensities of a target pixel and a neighboring pixel, respectively. When the noise probability of intensity difference is small, the proposed range kernel judges that the difference is caused by a real scene change, and the neighboring pixel is excluded from the smoothing by assigning a smaller weight.

All of the noise probabilities according to $|I(x_i) - I(x)|$ and μ are obtained by calibration [15] and they are stored in LUTs in advance. Because the proposed range kernel refers to the noise probability simply from the LUTs using $|I(x_i) - I(x)|$ during operation without extra computations, the speed is not degraded in comparison with the conventional bilateral filter.

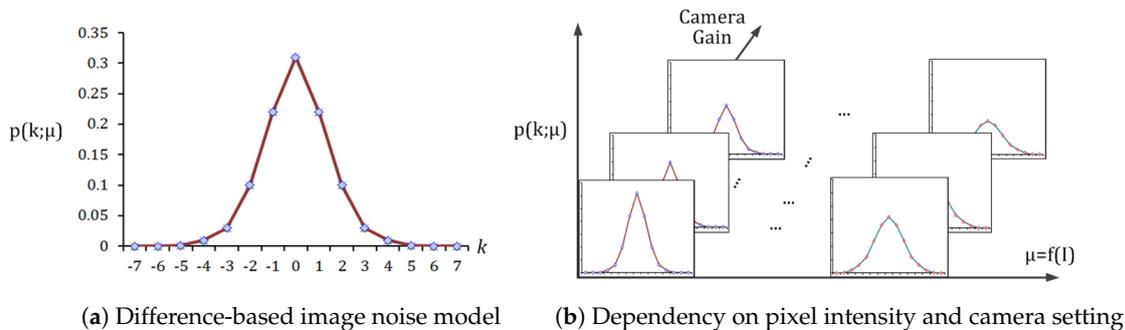


Figure 1. Intensity difference-based image noise model [15]. Here, I and k denote a pixel intensity and the difference in intensity between two random pixels, respectively; and, $p(k)$ and μ are the noise probability function and its parameter, respectively.

3.2. Binary Noise-Aware Bilateral Filter (B-NABF)

The quality degradation due to noise can be mitigated substantially by the proposed NABF. However, the NABF increases the LUT usage considerably owing to storing noise probability values corresponding to all intensity differences, μ s, and camera gains. A binary NABF (B-NABF) which approximates (5) using the statistical hypothesis test is introduced to significantly reduce the computational complexity of the bilateral filter as well as the LUT usage[16]:

$$f_r^{B-NABF} = \begin{cases} 1 & \text{if } \|I(x_i) - I(x)\| \leq K_C(x), \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

$$\text{where } K_C(x) = \arg \max A(k) \text{ s.t. } A(k) \leq 1 - \alpha, \quad A(k) = F(k; \mu) - F(-k; \mu).$$

Here, f_r^{B-NABF} is the binary noise-aware range kernel. $I(x)$ and $I(x_i)$ are the intensity of a target pixel and that of a neighboring pixel, respectively; $K_C(x)$ is the critical value corresponding to the $I(x)$; F is the cumulative distribution function of the f_r^{NABF} ; and α is the significance level, which is commonly set from 0.01 to 0.1 [16].

The $K_C(x)$ value is determined to be a k , which is the cumulative value of f_r^{NABF} between two symmetric k s ($A[k]$) that is closest to $1-\alpha$. A weight is assigned to 1 or 0 by comparing the intensity difference with $K_C(x)$, as shown in Figure 2. Consequently, the use of LUT is reduced because the B-NABF only requires K_C s instead of noise probability values at all intensity differences. Furthermore, the computational complexity is reduced owing to a binarization of weights.

To further reduce the LUT usage, only the K_C s of partial camera gains are stored in the LUTs instead of storing the K_C values corresponding to all camera gains, and an interpolated K_C for a new camera gain is generated, as follows:

$$K_C^i = R \left(\frac{K_C^{i\downarrow} \times (G_{i\uparrow} - G_i) + K_C^{i\uparrow} \times (G_i - G_{i\downarrow})}{G_{i\uparrow} - G_{i\downarrow}} \right), \quad (7)$$

where $R(\cdot)$ is a round operation; K_C^i and G_i are the critical value and camera gain of an index i , respectively; and, $G_{i\downarrow}$ and $G_{i\uparrow}$ are the closest lower and upper gains with i , respectively.

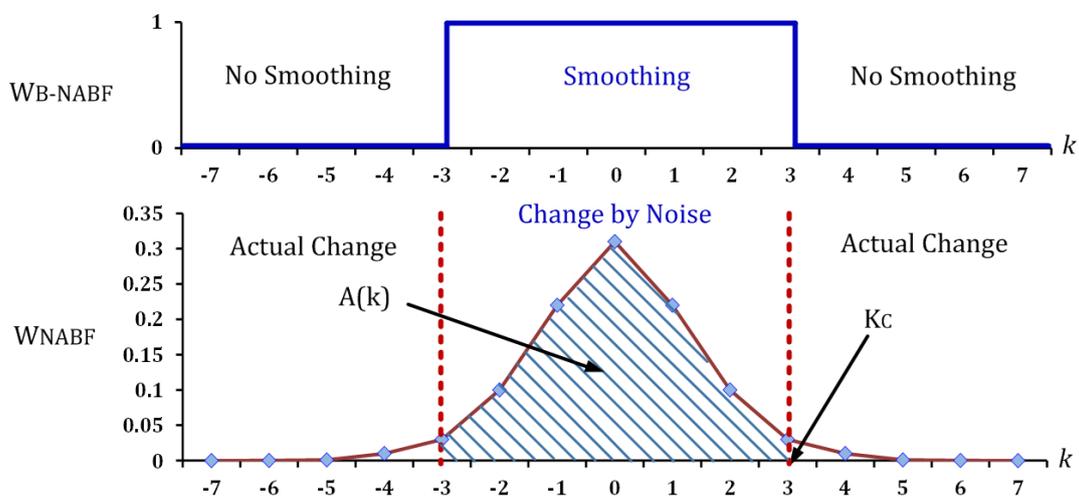


Figure 2. Binary noise-aware range kernel. Here, $A[k]$ is the cumulative value of probability between two symmetric k s; K_C denotes the critical value for a binary range kernel; and, W_{NABF} and W_{B-NABF} are weights by a noise-aware range kernel and a binary noise-aware range kernel, respectively.

4. VLSI Design

A VLSI is designed to accelerate the proposed binary noise-aware bilateral filter. To process a 5×5 -sized pixel window per clock, the hardware design is fully parallelized and pipelined. It mainly consists of the main controller, a binary range kernel unit to compute (6), a spatial kernel unit to perform f_s of (1), a K_C memory to store critical values, and a K_C interpolation unit to calculate (7), as shown in Figure 3. The operation of each unit is described in detail, as follows.

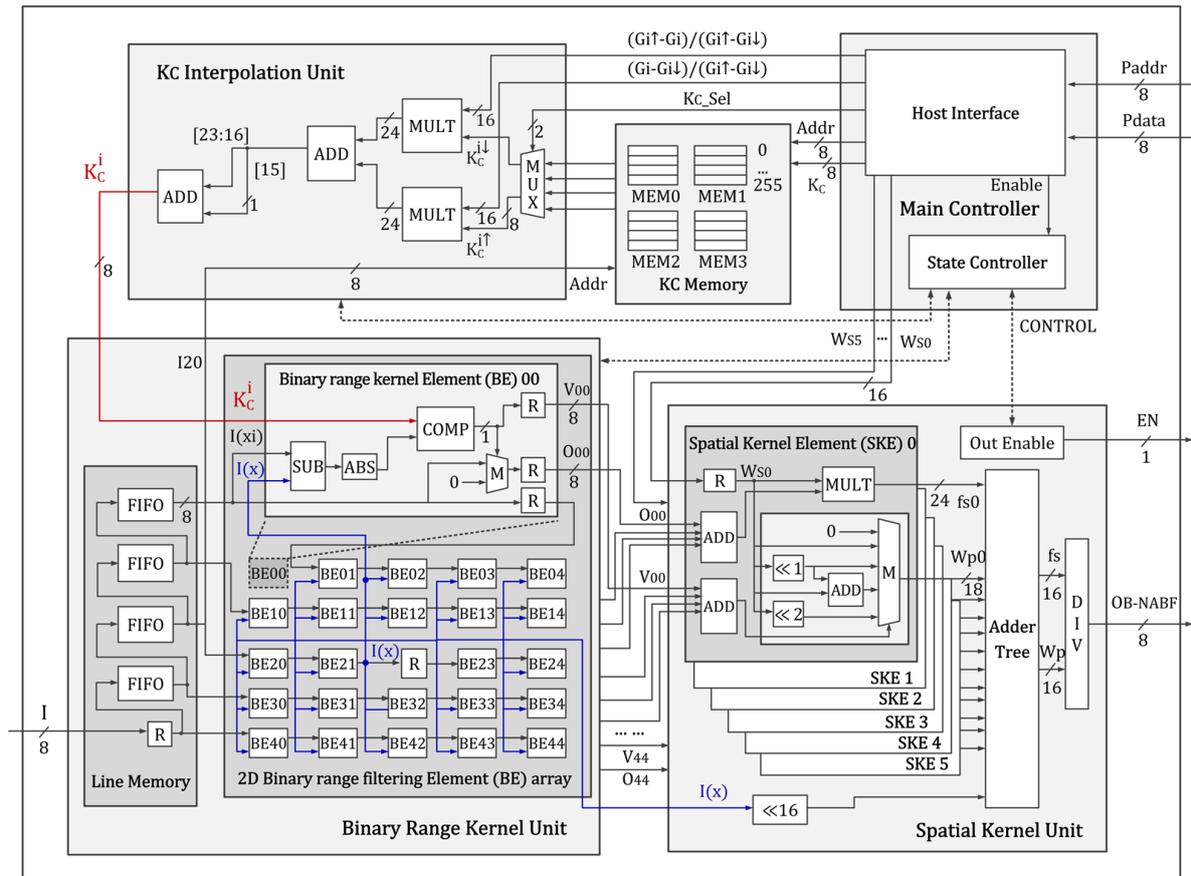


Figure 3. Proposed VLSI architecture of noise-aware bilateral filter. Here, R denotes registers; ADD and SUB denote adders and subtractors, respectively; MULT and DIV are multipliers and the divider, respectively; M and MUX are multiplexers; and ABS and COMP are the logic to conduct absolute operation and comparison, respectively.

4.1. Main Controller

The main controller comprises a state controller for handling the operational sequence of all units which are pipelined pixel by pixel, as shown in Figure 4, and a host interface module to communicate with the host central processing unit (CPU). First, the state controller activates the line memory in the binary range kernel unit for the line buffering of an image. The pixel intensity (I) of each line is sent to the two dimensional (2D) binary range kernel element (BE) array module sequentially to produce the 2D pixel window. When two lines of the line memory are full and three I s of the third line are stored, the state controller starts the operation of the 2D BE array module in order to compute (6). Subsequently, the spatial kernel unit is enabled after one clock. The K_C interpolation unit is activated before three clock cycles of the start of the 2D BE array module to provide K_C^i on time. As a result, the final output of the B-NABF is generated per clock.

The data, such as K_Cs , α , and the spatial kernel weight (f_s in (1)), are sent by the host CPU. The host interface module stores them in registers, and the data are sent to each unit.

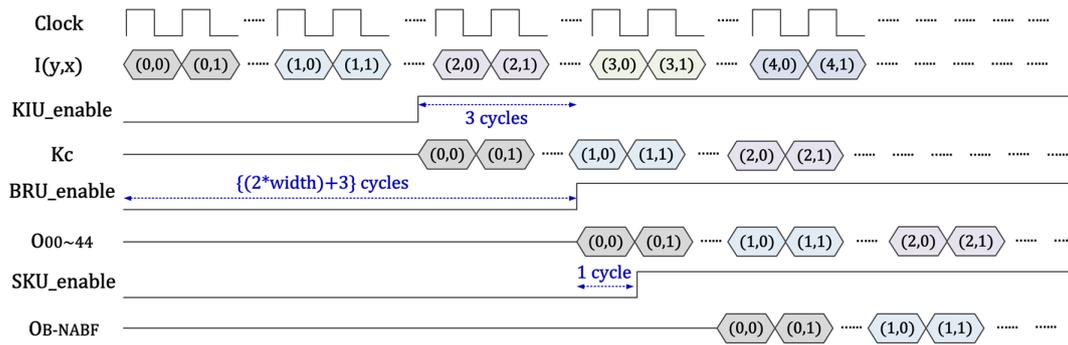


Figure 4. Control and timing diagram for pipelined operation. Here, I denotes the pixel intensity of an input image; x and y denote horizontal and vertical position in an image, respectively; KIU, BRU, and SKU are K_C interpolation unit, binary range kernel unit, and spatial kernel unit, respectively; K_C , $O_{00} \sim O_{44}$, and O_{B-NABF} are outputs of KIU, BRU, and SKU, respectively.

4.2. Binary Range Kernel Unit

The binary range kernel unit computes (6) in parallel for all pixels within the window. To scale the size of the pixel window easily, a unit module, called the binary range kernel element (BE), is designed for computing (6) in one pixel, and the BE is duplicated and it is connected to process the pixel window in parallel, as shown in Figure 3. A subtractor calculates $I(x_i) - I(x)$ in the BE, and the absolute value is computed using the sign of its most significant bit (MSB) and a two's complement. Furthermore, a comparator is employed to compare $|I(x_i) - I(x)|$ with K_C^i , and a multiplexer is connected to select the input intensity or 0 as an output (O_{xi}) according to the compared result. The compared result, which is 1 or 0, is sent to the spatial kernel unit as a valid signal (V_{xi}) with O_{xi} .

The line memory is designed using first-in first-out (FIFO) memories connected serially and a shift register in the BE in order to provide the intensities of the neighboring pixels to each BE in parallel. The output ports of each FIFO memory are connected to the input port of the BEs in the first column in the 2D BE array module, and intensities are shifted to the right BE while using the shift register connected horizontally, as shown in Figure 3. The output of the shift register in BE21 is used as the center pixel of the window ($I(x)$) and it is connected with all of the BEs.

In addition, the output of the third FIFO memory (I_{20}) is sent to the K_C interpolation unit in advance to generate the K_C^i corresponding to $I(x)$ and synchronize it with $|I(x_i) - I(x)|$.

4.3. K_C Memory & K_C Interpolation Unit

A memory with data width of eight bits and a depth of 256 is used for the K_C memory, because the intensity and its difference are between 0 and 255. Four memories are used to store K_C s of four camera gains. Furthermore, a logic in the host interface module is added to update the K_C memory by the host CPU at the initialization.

The K_C interpolation unit accesses the K_C memory using I_{20} , which is sent from the binary range kernel unit, as an address, and the four K_C s corresponding to each camera gain are obtained. Here, $K_C^{i\downarrow}$ and $K_C^{i\uparrow}$ are selected among the four K_C s by a multiplexer and its selection signal (K_C_Sel), which is computed in advance while using a current camera gain in the host CPU. Subsequently, (7) is computed using two parallel multipliers and an adder. The precomputed $\frac{G_{i\uparrow} - G_i}{G_{i\uparrow} - G_{i\downarrow}}$ and $\frac{G_i - G_{i\downarrow}}{G_{i\uparrow} - G_{i\downarrow}}$ of (7) are sent from the host CPU. Finally, to generate K_C^i , the output of the adder is summed with its MSB of the fractional part for a round operation.

4.4. Spatial Kernel Unit

The 24 neighboring pixels in the window are classified into six groups according to the distance from the center pixel. The spatial kernel element (SKE) module adds up the four O_{xi} s of each same group, and the f_s of the group is computed by multiplying the sum with the spatial kernel weight that corresponded to the group. Furthermore, to obtain the normalization term (W_p) of each group, the four V_{xi} values, which denote the valid pixel after conducting the binary range kernel, are summed up. According to the sum of V_{xi} s, W_p is selected among the multiples of W_s , which are generated using the shift logic and the adder, as shown in Figure 3. Then the f_s values and W_p values computed by the six parallel SKE modules are added up, respectively, by the adder tree. Finally, a divider divides the sum of f_s s by the sum of W_p s, in order to generate the final output (O_{B-NABF}).

5. Experimental Results

5.1. Image Quality by Denoising

The quality of images are evaluated, which were filtered by the proposed NABF and B-NABF, via PC simulation before designing a dedicated VLSI. Five highly-textured test datasets are created, which were captured by a PointGrey Flea3 camera, as shown in Figure 5. Furthermore, a scene is captured 1,000 times and averaged the images in order to produce ground truth images. Six camera gain settings are used to verify the effect of noise variation. In addition, σ_s , σ_r , and α were varied to evaluate the quality variation according to the parameter setting.

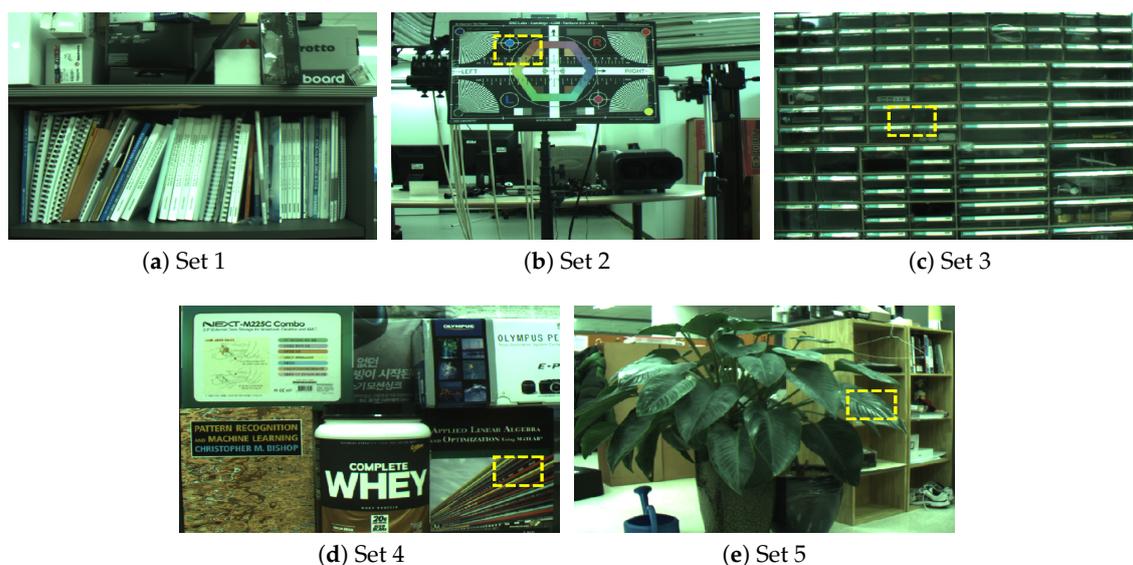


Figure 5. Dataset for the evaluation. As shown From (a) to (e), we captured highly-textured and various color-distributed scenes for evaluating the proposed method. Yellow-dashed rectangles are the regions to be enlarged for the qualitative comparison presented in Figure 6.

Table 1 shows the average peak signal-to-noise ratios (PSNRs) of the conventional bilateral filter (BF), the proposed NABF, and B-NABF. The 15(I) of camera gain settings used the K_{CS} , which were interpolated using (7) based on measured K_{CS} in 10 and 18. The NABF improves the PSNR when compared with the input image under all noise conditions and parameter changes. However, the conventional BF decreases the PSNR when the noise level is low. Furthermore, the PSNR degradation by B-NABF and by using interpolated K_{CS} is negligible when compared with that by NABF and when measured K_{CS} were used, despite the approximation. In addition, the conventional BF is quite sensitive to variation of σ_s and σ_r . Whereas, the PSNR by NABF is stable despite variations of σ_s and α . For instance, the PSNR differences of the NABF and the conventional BF according to the parameter setting are 0.2 dB and 6.4 dB, respectively, when the camera gain is 0 dB.

Table 1. Average peak signal-to-noise ratio (PSNR) under Various Noise Conditions.

| Parameter | Input | Conventional BF | | | NABF | | B-NABF | | |
|--------------|---------|-----------------|------|------|------|------|--------|------|------|
| σ_s | | 1.0 | 1.0 | 2.0 | 1.0 | 2.0 | 1.0 | 1.0 | |
| σ_r | N/A † | 0.04 | 0.08 | 0.08 | N/A | | N/A | | |
| α | | N/A | | | N/A | | 0.1 | 0.05 | |
| Gain [dB] | 0 | 45.1 | 42.3 | 38.9 | 35.9 | 45.7 | 45.5 | 45.5 | 45.3 |
| | 5 | 43.7 | 42.0 | 38.2 | 35.4 | 44.4 | 44.2 | 44.2 | 44.1 |
| | 10 | 42.1 | 41.7 | 38.0 | 35.3 | 43.1 | 42.8 | 42.8 | 42.7 |
| | 15 | 39.3 | 40.0 | 37.4 | 35.0 | 40.5 | 40.2 | 40.2 | 40.1 |
| | 15(I) ‡ | | N/A | | | 40.4 | 40.2 | 40.1 | 40.1 |
| | 18 | 36.7 | 37.7 | 36.0 | 34.1 | 37.8 | 37.5 | 37.6 | 37.5 |

† N/A = Not Applicable, ‡ The 15(I) of camera gain settings used K_C values, which were interpolated using (7) based on measured K_C s in 10 and 18.

Table 2 shows the averaged values of PSNR of all camera gains for each test set. With scene variation, the NABF and B-NABF are also more robust to parameter variation than the conventional BF, and the PSNR is improved.

Table 2. Average PSNR with Scene Variation.

| Parameter | Input | Conventional BF | | | NABF | | B-NABF | | |
|-------------|-------|-----------------|------|------|------|------|--------|------|------|
| σ_s | | 1.0 | 1.0 | 2.0 | 1.0 | 2.0 | 1.0 | 1.0 | |
| σ_r | N/A † | 0.04 | 0.08 | 0.08 | N/A | | N/A | | |
| α | | N/A | | | N/A | | 0.1 | 0.05 | |
| Test Set | #1 | 42.4 | 41.9 | 38.5 | 35.7 | 43.5 | 43.2 | 43.2 | 43.1 |
| | #2 | 38.9 | 38.9 | 36.9 | 34.7 | 39.6 | 39.4 | 39.5 | 39.5 |
| | #3 | 42.5 | 41.7 | 37.9 | 35.3 | 43.7 | 43.4 | 43.4 | 43.4 |
| | #4 | 40.6 | 39.7 | 36.7 | 34.5 | 41.2 | 41.0 | 40.9 | 40.8 |
| | #5 | 42.5 | 41.5 | 38.5 | 35.6 | 43.4 | 43.1 | 43.0 | 42.9 |

† N/A = Not Applicable.

Figure 6 presents the qualitative results of the BF, the proposed NABF, and B-NABF with the ground truth images. The NABF shows clearer edges and textures similar to those of the ground truth images, while the conventional BF produces less distinct results. Furthermore, the results of the B-NABF are almost the same as them of the NABF.

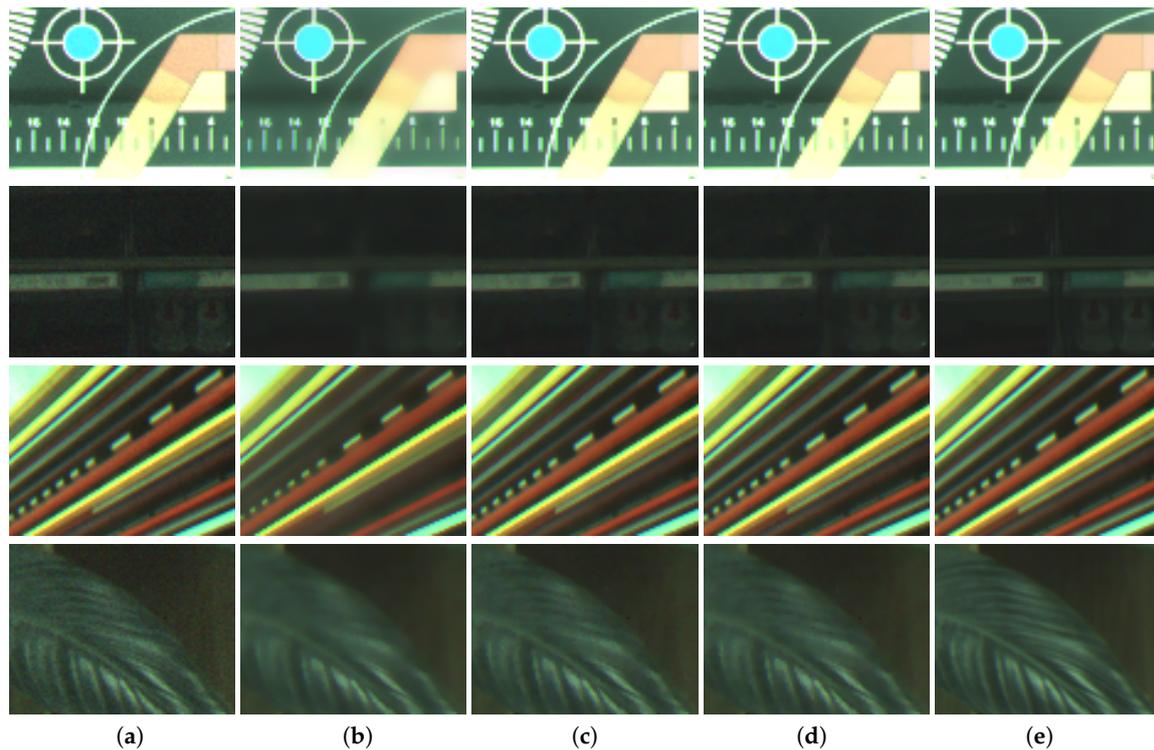


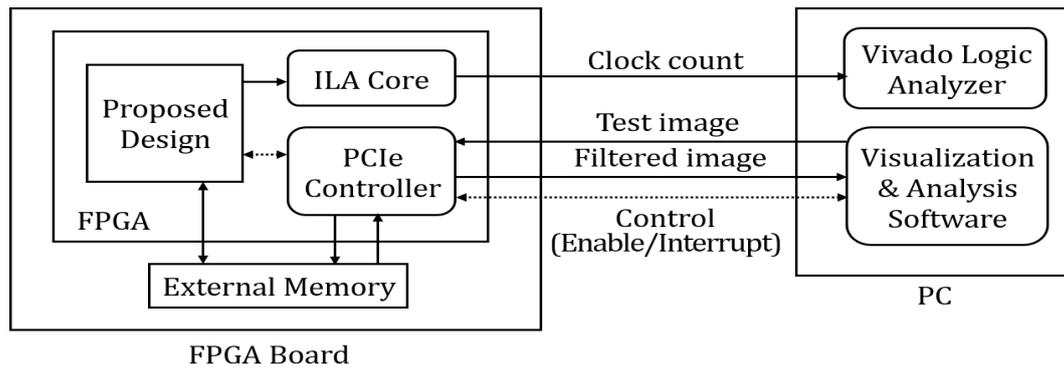
Figure 6. Qualitative comparison: (a) input images, the results of (b) the conventional bilateral filter, (c) the proposed NABF, (d) B-NABF, and (e) the ground truth

5.2. Implementation Result and Comparison

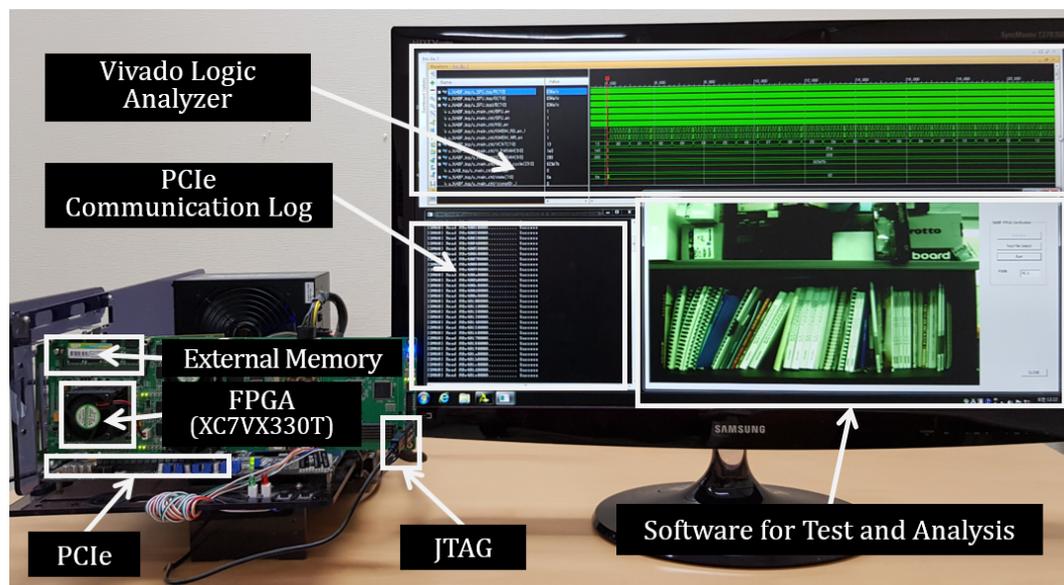
Based on an architecture proposed in Section 4, a VLSI of B-NABF is designed while using Verilog hardware description language. It is verified that the register-transfer level (RTL) simulation results of the designed VLSI coincide with simulation results from the C model of B-NABF which is used in Section 5.1. Finally, the verified VLSI design was implemented in a Xilinx Virtex 7 FPGA (XC7VX330T).

5.2.1. System Configuration for Measurement

A PC is connected with a FPGA board using a peripheral component interconnect express (PCIe) to verify the proposed VLSI design and measure its performance, as shown in Figure 7. A test image from the PC is stored in an external memory of a FPGA board, and the VLSI design is enabled by the register setting from the PC. After bilateral filtering is finished, an interrupt signal is generated by the main controller of VLSI and then sent to the PC. The interrupt service routine reads the filtered image from the external memory and displays it. The PSNR is calculated by comparing the filtered image with the ground truth image. To execute these processes automatically, a software programs for visualization and analysis is implemented, as shown in Figure 7b. Moreover, the throughput is measured using a clock counter, which is placed in the proposed design. The total number of clocks is probed using Xilinx integrated logic analyzer (ILA), and it is displayed in the Vivado logic analyzer via joint test action group (JTAG) interface. The logic and memory usage are reported after post-implementation in the Vivado design suite tool.



(a) Block diagram



(b) System verification

Figure 7. System configuration for the experiment.

5.2.2. Comparison with Recent VLSI Designs of Bilateral Filter

As in Table 3, VLSI designs [12,13] are considered that do not use an external memory in this study, because designs [10] that use an external memory are inappropriate for mobile systems due to the large power consumption. One earlier design [12] handles a large image in realtime owing to parallelized and pipelined implementation of (1). Pixels in a pixel window are grouped according to the distance from the center, and the pixels in each group are processed subsequently in a separate computing pipeline in order to reduce the usage of the logic and internal memory. However, the throughput is reduced due to the sharing of computing pipeline. Another design [13] implements an iterative algorithm of the bilateral filter, which was presented in [23], in order to handle an arbitrary pixel window size. However, the throughput is reduced despite a small image size due to the iterative nature of this algorithm. Moreover, a large amount of the internal memory is used for the storage of intermediate data between iterations.

In contrast, the throughput of the proposed VLSI design is 10.5 and 95.7 times higher than those of [12,13], respectively, owing to the proposed binary noise-aware bilateral filtering scheme. Furthermore, 63.6% and 97.5% less internal memory are used in comparison to [12,13], respectively. The logic usage is also significantly reduced. In terms of the image quality after filtering, as described in Section 5.1, the proposed noise-aware bilateral filter provides more stable and better quality under various noise conditions in comparison to the conventional bilateral filter adopted in [12,13].

Table 3. Comparisons with State-of-the-Art VLSI Designs of Bilateral Filter.

| | [12] | [13] | Proposed |
|------------------------|-----------------------------|----------------------------|------------------------------|
| Device | Xilinx-5 FPGA (XC5VLX50) | Xilinx-7 FPGA (XC7Z020) | Xilinx-7 FPGA (XC7VX330T) |
| Image Resolution | 1024×1024 | 256×256 | 1920×1080 |
| Max. Freq. (MHz) | 320 | 63 | 330 |
| Throughput (Mpixels/s) | 31.5 | 3.45 | 330 |
| Logic Usage (ea) | Slice [‡] | 1060 | + 476 ^a |
| | LUT | + | 2647 |
| | FF | + | 686 |
| | DSP | 29 | 10 |
| Memory Usage (KByte) | 49.5 | 706.5 | 18 |

[†] It is not reported, [‡] It consists of multiple LUTs and FFs, ^a Synthesis result in XC7VLX50 (Xilinx-5 FPGA) for a fair comparison with [12].

5.2.3. Integration of Implemented VLSI Design and Image Sensor

A rapid prototyping system (<http://huins.com/en/m11.php?m=rd&no=86>) is used, which is based on Xilinx-6 XC6VLX760 FPGA, to integrate implemented VLSI design and CMOS image sensor (CIS). Omnivision OV5642 CIS is connected with the VLSI design in XC6VLX760 FPGA, and filtered output image is displayed in a monitor via high definition multimedia interface (HDMI), as shown in Figure 8.

In addition, a VLSI design of the conventional bilateral filter is implemented for comparing with a result of the proposed VLSI design qualitatively. As a result, an image that is filtered by the proposed VLSI design shows clearer edges and textures when compared with an image from a VLSI design of the conventional BF, as shown in Figure 9.

Figure 9 presents test results, which are filtered by the conventional bilateral filter and the proposed B-NABF, respectively, with the original image. A noise of the original image is reduced by a VLSI design of the proposed B-NABF. Furthermore, an image of the proposed VLSI design shows clearer edges and textures when compared with an image from a VLSI design of the conventional bilateral filter.

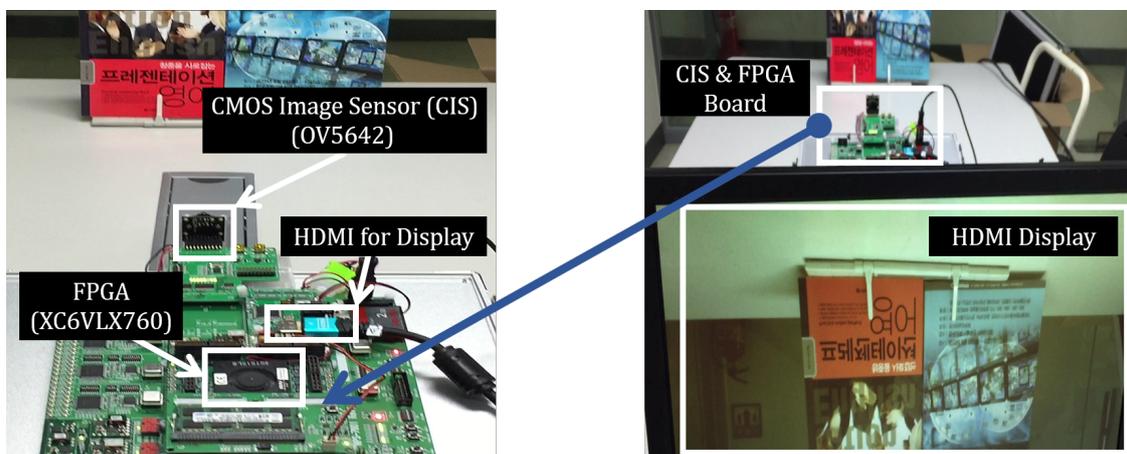


Figure 8. System configuration for integrating the implemented VLSI design of the proposed filter with CMOS image sensor.

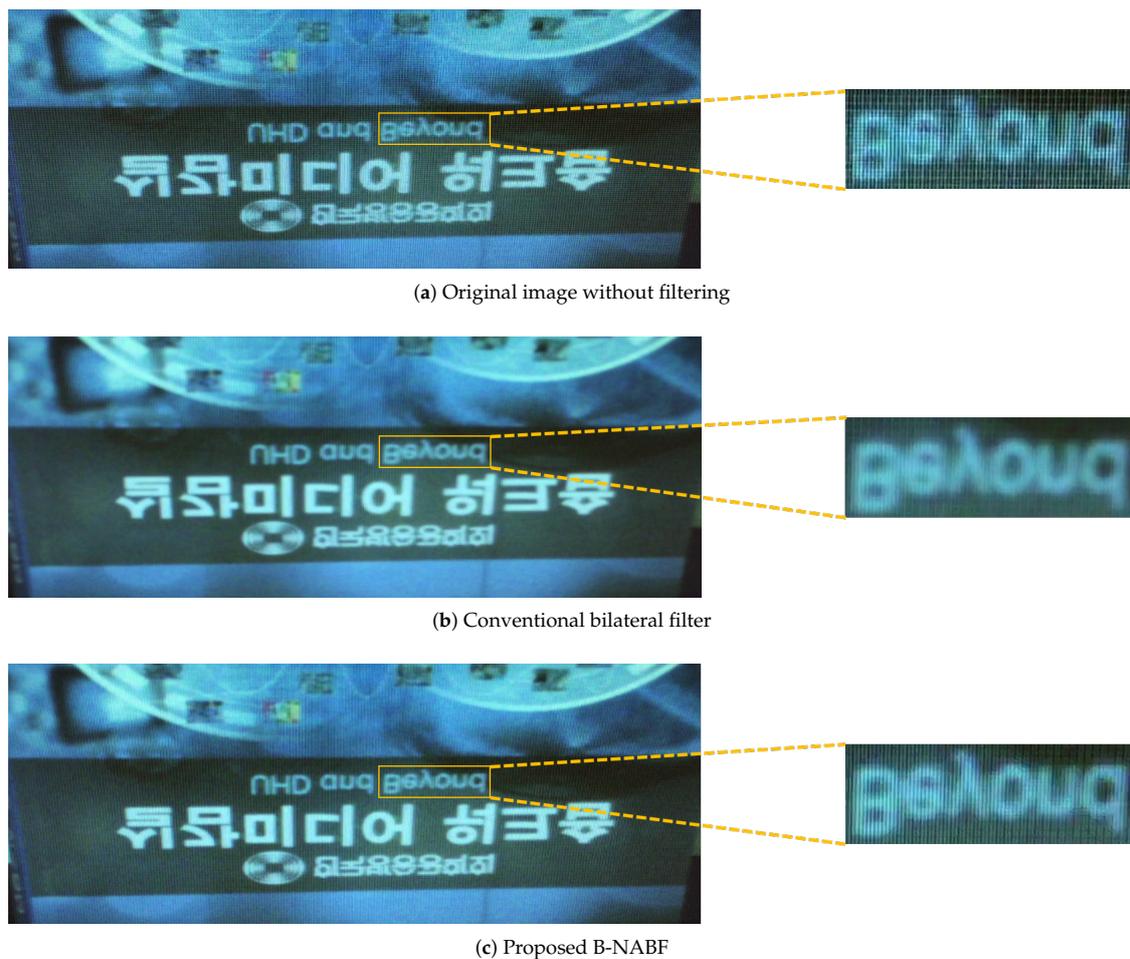


Figure 9. Filtered images by VLSI designs of the conventional bilateral filter and the proposed B-NABF.

6. Conclusions

We have proposed a noise-aware bilateral filter (NABF) to overcome the disadvantage of the conventional bilateral filter in which image quality is degraded by randomly generated noise in image sensors. The NABF estimates the noise using an intensity difference-based image noise model and dynamically adjusts the weight of the range kernel according to the estimated noise. In addition, a light weighting scheme is introduced, which approximates the range kernel of the NABF into a binary kernel while using the statistical hypothesis test method, for resource-limited mobile systems. Finally, a fully parallelized and pipelined VLSI architecture of NABF based on the proposed binary range kernel is presented, which was successfully implemented in a FPGA. Our experimental results demonstrated that the proposed NABF is more robust to noise than the conventional bilateral filter under various noise conditions. Furthermore, the proposed VLSI design of the NABF achieves 10.5 and 95.7 times higher throughput and uses 63.6%–97.5% less internal memory than recent designs of the bilateral filter.

Author Contributions: Conceptualization, S.-J.J. and Y.H.; software, Y.H.; hardware, S.-J.J.; writing—original draft preparation, S.-J.J. and Y.H.; writing—review and editing, Y.H.; visualization, S.-J.J.; project administration, Y.H.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01351, Development of ultra low-power mobile deep learning semiconductor with compression/decompression of activation/kernel data) and by the research grant of the Chungbuk National University in 2019.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goyal, B.; Dogra, A.; Agrawal, S.; Sohi, B.; Sharma, A. Image denoising review: From classical to state-of-the-art approaches. *Inf. Fusion* **2020**, *55*, 220–244. [[CrossRef](#)]
2. Liu, D.; Wen, B.; Jiao, J.; Liu, X.; Wang, Z.; Huang, T.S. Connecting Image Denoising and High-Level Vision Tasks via Deep Learning. *IEEE Trans. Image Process.* **2020**, *29*, 3695–3706. [[CrossRef](#)] [[PubMed](#)]
3. Milyaev, S.; Laptev, I. Towards reliable object detection in noisy images. *Pattern Recognit. Image Anal.* **2017**, *27*, 713–722. [[CrossRef](#)]
4. Hasan, M.; El-Sakka, M. Improved BM3D image denoising using SSIM-optimized Wiener filter. *EURASIP J. Image Video Process.* **2018**, *2018*, 25. [[CrossRef](#)] [[PubMed](#)]
5. Song, Y.; Zhu, Y.; Du, X. Dynamic Residual Dense Network for Image Denoising. *Sensors* **2019**, *19*, 3809. [[CrossRef](#)] [[PubMed](#)]
6. Lefkimiatis, S. Universal denoising networks: A novel CNN architecture for image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3204–3213.
7. Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the IEEE International Conference on Computer Vision, Bombay, India, 4–7 January 1998; pp. 839–846.
8. Choi, S.H.; Cho, J.; Tai, Y.M.; Lee, S.W. A parallel camera image signal processor for SIMD architecture. *EURASIP J. Image Video Process.* **2016**, *2016*, 29. [[CrossRef](#)]
9. Petreto, A.; Romera, T.; Lemaitre, F.; Masliyah, I.; Gaillard, B.; Bouyer, M.; Meunier, Q.L.; Lacassagne, L. A new real-time embedded video denoising algorithm. In Proceedings of the Conference on Design and Architectures for Signal and Image Processing, Montreal, QC, Canada, 16–18 October 2019; pp. 47–52.
10. Tseng, Y.; Hsu, P.; Chang, T. A 124 Mpixels/s VLSI design for histogram-based joint bilateral filtering. *IEEE Trans. Image Process.* **2011**, *20*, 3231–3241. [[CrossRef](#)] [[PubMed](#)]
11. Dutta, H.; Hannig, F.; Teich, J.; Heigl, B.; Hornegger, H. A design methodology for hardware acceleration of adaptive filter algorithms in image processing. In Proceedings of the IEEE International Conference on Application-specific Systems, Architectures and Processors, Steamboat Springs, CO, USA, 11–13 September 2006; pp. 331–340.
12. Gabiger-Rose, A.; Kube, M.; Weigel, R.; Rose, R. An FPGA-based fully synchronized design of a bilateral filter for real-time image denoising. *IEEE Trans. Ind. Electron.* **2014**, *61*, 4093–4104. [[CrossRef](#)]
13. Dabhade, S.D.; Rathna, G.N.; Chaudhury, K.N. A Reconfigurable and Scalable FPGA Architecture for Bilateral Filtering. *IEEE Trans. Ind. Electron.* **2018**, *65*, 1459–1469. [[CrossRef](#)]
14. Tsin, Y.; Ramesh, V.; Kanade, T. Statistical calibration of CCD imaging process. In Proceedings of the IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; pp. 480–487.
15. Hwang, Y.; Kim, J.; Kweon, I.S. Difference-based image noise modeling using skellam Distribution. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1329–1341. [[CrossRef](#)] [[PubMed](#)]
16. Greenland, S.; Senn, S.J.; Rothman, K.J.; Carlin, J.B.; Poole, C.; Goodman, S.N.; Altman, D.G. Statistical tests, P values, confidence intervals, and power: A guide to misinterpretations. *Eur. J. Epidemiol.* **2016**, *31*, 337–350. [[CrossRef](#)] [[PubMed](#)]
17. Peng, H.; Rao, R. Bilateral kernel parameter optimization by risk minimization. In Proceedings of the IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 3293–3296.
18. Kishan, H.; Seelamantula, C.S. Optimal parameter selection for bilateral filters using Poisson Unbiased Risk Estimate. In Proceedings of the IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012; pp. 121–124.
19. Chen, Y.; Shu, Y. Optimization of bilateral filter parameters via chi-square unbiased risk estimate. *IEEE Signal Process. Lett.* **2014**, *21*, 97–100. [[CrossRef](#)]
20. Luisier, F.; Wolfe, P.J. Chi-square unbiased risk estimate for denoising magnitude MR images. In Proceedings of the IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 1561–1564.
21. Chang, H.H.; Lin, Y.J.; Zhuang, A. An automatic parameter decision system of bilateral filtering with GPU-Based acceleration for brain MR images. *J. Digit. Imaging* **2018**, *32*, 148–161. [[CrossRef](#)]

22. Porikli, F. Constant time $O(1)$ bilateral filtering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 23–28 June 2008; pp. 1–8.
23. Chaudhury, K.N.; Dabhade, S.D. Fast and provably accurate bilateral filtering. *IEEE Trans. Image Process.* **2016**, *25*, 2519–2528. [[CrossRef](#)] [[PubMed](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).