Electronic Preprint for Journal of Information Processing Vol.22 No.4

Regular Paper

High Density Impulse Noise Removal Based on the Total Observation Kernel Element for Image Sequences

Fitri Utaminingrum[1,2,a)] Keiichi Uchimura[1,b)] Gou Koutaki[3,c)]

Received: November 27, 2013, Accepted: May 17, 2014

Abstract: Several different methods for impulse noise removal in image sequences have been proposed. However, all of them are not successful in removing high density of impulse noise. Hence, this paper proposes a filtering method for reducing high density impulse noise in the image sequences. We use three windows with size 3 × 3 to obtain a new window with similar size. Three windows are taken from the next-frame, current frames and previous frames. The recursive window is applied in the current frames. The filtering process uses decision-based method. Meanwhile, a pixel for replacing the noisy pixel is calculated from a new window based on weighting method. Our experimental results show that the proposed method can not only reduce the high impulse noise in image sequences well, but also preserve more details and textures.

Keywords: impulse noise removal, image sequences, frames, *PSNR*, *MSSIM*

1. Introduction

Image sequence processing is a field that continues to grow, with new applications being developed in many aspects. Image sequence processing usually refers to a two-dimensional digital signal processing by using a computer. Displaying several frames of two-dimensional (2D) images in the sequence image creates the illusion of motion.

The common noise problem in the digital image sequence is impulse noise. It is caused by the imperfection of camera sensors and communication channel, error in the data-acquisition system, interference from the outside instrumentation and error in the transmission channel, etc. [1], [2], [3]. The noise needs to be eliminated because it will be decreasing the quality of an image.

The main purpose of our research is to obtain a good quality of image denoising. However, many cases also needing attention include:

(1) The visual result of the proposed method must have a smooth and clear texture.

(2) The filtering process is conducted especially on the noisy pixel, without engaging the important pixel that is indicated as the original pixel.

(3) An approach of the filtering method is created to obtain a fast computation time.

Several 2D filter methods can be used to filter an image sequence or video. There are three steps for applying a 2D filter in the image sequence filtering as follows: Firstly, the video is converted into frames and then the frames are converted into images. Secondly, a 2D filter method is applied for filtering the corrupted images which are separated from the frames. Thirdly, after the frame was filtered, these frames are converted back into the movie.

Standard Median Filter (SMF) is often used in the 2D image or in the image sequence, especially for the filtering process. However, this method is only optimal to reduce the small impulse noise density. Recently, many researchers have adopted median filter to improve the capability of the filtering method. Wang et al. [3] have proposed a new median method based on switching filter, that is called Progressive Switching Median Filter (PSMF). In this regard, impulse noise detection and filter are applied progressively in iterative manners. However, this method is still not optimal to reduce the high density impulse noise. This is because the PSMF method using the typical median filter only changes the size of the window filter from 3 × 3 to 5 × 5. The problem arises when the impulse noise density is higher than 50% [4]. Srinivasan et al. [5] proposes a filter based on decision, that only filters the noisy pixel. This method is known as a Decision-Based Algorithm (DBA). This method uses a median filter with three times sorting (row, column and right) to obtain a pixel in center position. This method has a problem when the total of the noisy pixel is greater than four in the 3 × 3 window. Automatically, the median position is the noisy pixel. The improvements of the boundary discriminative noise detection (BDND) filter is proposed by

[1] Computer Science and Electrical Engineering, Graduate School of Science and Technology, Kumamoto University, Chuoku, Kumamoto 860–8555, Japan

[2] Information Technology and Computer Science Program, Brawijaya University, Jl. Veteran No.8 Malang, East Java, Indonesia 65145

[3] Priority Organization for Innovation and Excellence, Kumamoto University, Kumamoto 860–8555, Japan

[a)] f3-ningrum@navi.cs.kumamoto-u.ac.jp

[b)] uchimura@cs.kumamoto-u.ac.jp

[c)] koutaki@cs.kumamoto-u.ac.jp

Ref. [6]. Here, the modification method of the BDND basically loosens the condition imposed on expanding the filtering window and combine the spatial information of the pixels in the filtering process. These modification able to improve of BDND method performance. However, it still can not repair the damage area in the edge image.

The related work about impulse noise removal in image sequences has been proposed by Seon et al. [7]. They solve the problem using the Rank-Order-Mean (ROM) filter in the recursive window [8]. The method [7] uses 11-pixels. In here, nine pixels are obtained from the current-frame, one-pixel is taken from the next-frame and one-pixel is taken from the previous-frame in the (i, j) position. As well as in the filtering based on median filter, the ROM filter will be failed, if the number of noisy pixel is greater than half of the number pixels in a window. The Lone Diagonal Sorting (LDS) method is also related to the filtering impulse noise on the image sequence that uses a lone diagonal for denoising the impulse noise [9]. LDS method does not consider the pixel from the frames that has been conducted in the previous process and the next frames to be filtered.

To overcome the problem of the previous methods, we propose the impulse noise removal in image sequences using total observation element kernels. A new 3 × 3-window is obtained from the observation kernel from three windows that are derived from the next-frame, the current-frame and the previous-frame. Meanwhile, the 3 × 3-window in the current-frame is the recursive window. Furthermore, a pixel for replacing the corrupted pixel at (i, j) position is calculated from a new window based on the weighting method.

2. Corrupted Image by Impulse Noise

Figure 1 shows the mixing process of an impulse noise in the clean image (I). In general, there are two types of impulse noise namely: salt-and-pepper noise and random-valued impulse noise [10]. Both of noise types can be written as follows:

(1) The salt-and-pepper noise model.

$$F_{ij} = \begin{cases} d_{\min} & \text{with probability } p/2 \\ d_{\max} & \text{with probability } p/2 \\ I_{ij} & \text{with probability } 1\text{-}p \end{cases} \quad (1)$$

F_{ij} is the noisy image. I_{ij} is the gray value of clean image I at location (i, j). $d_{\min}$ and $d_{\max}$ are dynamic range of an image. p is the noise density level of salt-and pepper with $0 \leq p \leq 1$.

Sometimes the salt-and-pepper noise is also called a fixed-impulse noise. The gray value of noise pixel is fixed at 0 and 255 for $d_{\min}$ and $d_{\max}$ respectively.

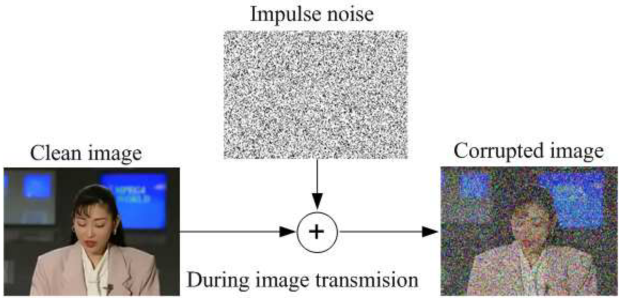


Fig. 1 Corrupted image by impulse noise.

(2) The random-valued impulse noise model.

$$F_{ij} = \begin{cases} d_{ij} & \text{with probability } p \\ I_{ij} & \text{with probability } 1 \text{ -}p \end{cases} \quad (2)$$

d_{ij} is uniformly distributed random numbers between the minimum value $d_{\min}$ and the maximum value $d_{\max}$. In here, p is the noise density of a random-valued impulse noise with $0 \leq p \leq 1$

The impulse noise in image sequences has a different impulse noise distribution in each frame.

3. Proposed Method

Figure 2 shows the illustration of the filtering process in the current frame with a recursive sliding window. Figure 2 expresses the shifting process of the frame from the right to the left as indicated by arrow lines. After the filtering process in the current frame (F^t) has been finished, frame shifts to the left side. The current frame that has been filtered ($\hat{F}^t$), is shifted to $\hat{F}^{t-1}$. Further, F^{t+1} will be shifted to the current frame (F^t) and F^{t+2} will be shifted to F^{t+1}. As shown in Fig. 2, the filtering process requires three data from $\hat{F}^{t-1}$, F^t and F^{t+1} frames.

In this method, we provide three kernels with 3 × 3 window size (W^{t-1}, W^t, W^{t+1}), which are obtained from $\hat{F}^{t-1}$ (frame that has been filtered), F^t (current frame) and F^{t+1}(frame that will be filtered) respectively.

Furthermore, the filtering process with a decision method is presented in **Fig. 3**. Figure 3 is the detailed explanation of Fig. 2 via a block diagram. Figure 3 shows three frames (F^{t-1}, F^t and F^{t+1}. In addition, another block of the filtering processes such as: the noise detector, the kernel observation and the image recon-

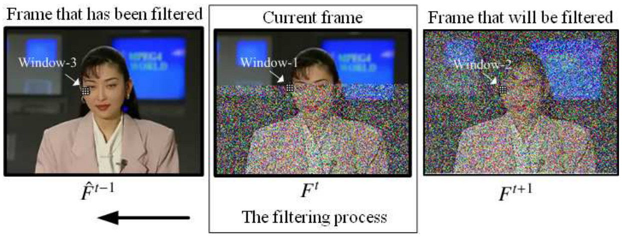


Fig. 2 Illustration of the filtering process.

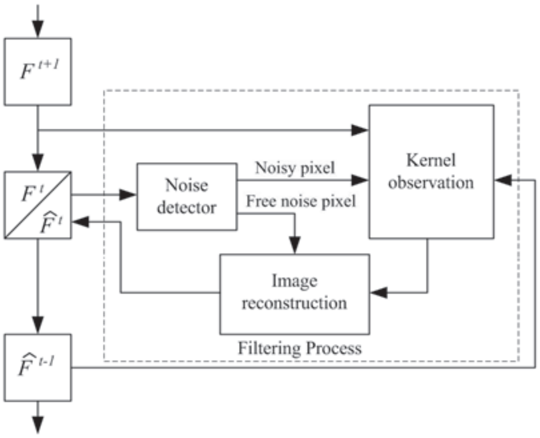


Fig. 3 Diagram block of the filtering process.

struction are also presented.

The first time in the filtering process, there is no frame on $\hat{F}^{t-1}$. In this case, we duplicate a frame on the current position (F^t) to F^{t-1}. Similarly, when the last frame F^{t+1} does not exists in the filtering process, we also duplicate the current frame (F^t) to F^{t+1} and then the process finishes. Thus, the filtering process for the current frame still refers to three frames.

The detailed description about the noise detector, kernel observation and image reconstruction will be explained in the following subsection.

3.1 Impulse Noise Detector

Impulse noise detector is used to detect a pixel at (*i,j*) position. The noise detector has two outputs, there are the noisy pixel and free-noise pixel. If the pixel at (*i,j*) position is a noisy pixel, then it will be issued at the noisy pixel output. The noisy pixel at (*i,j*) position will be replaced with a new pixel, which is calculated by the kernel observation. More details about the observation kernel will be described in the following subsection. If the pixel at (*i,j*) position is a free-noise pixel, then go straight to an image reconstruction as shown in Fig. 3.

Detecting of fixed impulse noise is more easily compared to a random-valued impulse noise. For salt-and-pepper noise, a noisy pixel at (*i,j*) position can be directly predictable. If the pixel value is 0 or 255, then the pixel is indicated as a noisy pixel. Meanwhile, if the pixel value is not 0 or 255, then it is considered as free-noise pixel. Whilst for the random-valued impulse noise, the pixel at (*i,j*) position cannot be guessed directly, but it will be compared with the neighbourhood pixels.

In this paper, we make an impulse noise detector which is capable of detecting both types of impulse noise. Both types of impulse noise are salt-and-pepper and random-valued impulse noise. Impulse noise detection (d) at each position (*i,j*) in the corrupted frame (F^t) is presented in Eq. (3).

$$d_{ij} = \begin{cases} 1 & \text{if } F^t_{ij} = 0 \text{ or } F^t_{ij} = 255 \text{ or } F^t_{ij} \leq \tau^L \text{ or } F^t_{ij} \geq \tau^H \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

τ^L and τ^H are the minimum and maximum limits of threshold value respectively, which is calculated by Eqs. (4) and (5).

$$\tau^L = MIN(W^t) + \frac{MED(W^t) - MIN(W^t)}{2} \tag{4}$$

$$\tau^H = MED(W^t) + \frac{MAX(W^t) - MED(W^t)}{2} \tag{5}$$

MIN, MED and *MAX* are operators to obtain minimum, median and maximum value respectively.

Fixed impulse noise (0 or 255) is easy to be detected by using Eq. (3). However, the noise detector in the Eq. (3) is also able to detect a random value impulse noise. Meanwhile, detecting of random value impulse noise requires a high and low value threshold(τ^L and τ^H). In here, τ^L is obtained between the median value and minimum value of the W^t-window, whilst τ^H is obtained between median value and maximum value of the W^t-window as illustrated in Eqs. (4) and (5). Equation (3) uses four indicator values (0, 255, τ^L and τ^H) for detecting random value impulse noise and also the fixed impulse noise.

For example, we show a sample pixel for Lena's image in the 3 × 3 window at coordinates (22,4) up to (24,6) to determine the performance of the detector noise. **Figure 4** (a) shows sample pixels from the original image, and Fig. 4 (b) is the sample pixel of the corrupted image by a random impulse noise 50%.

Referring to Fig. 4 (b), we get the values of $MIN(W^t) = 3$, $MED(W^t) = 156$ and $MAX(W^t) = 236$. By applying Eqs. (4) and (5), we can obtain $\tau^L = 79.5$ and $\tau^H = 196$.

In this case, the pixel value at position F^t_{ij} is 3. Based on Eq. (3), this pixel value is included in the criteria when the condition $F^t_{ij} \leq \tau^H$, then F^t_{ij} is detected as a noisy pixel. In the same manner, τ^L and τ^H can also be used to detect a free-noise pixel.

3.2 Kernel Observation

The aim of kernel observation is to get a new-kernel (W^n), which is observed from three kernel W^t, W^{t+1} and W^{t-1}. A kernel at (*i,j*) position for W can be written as in the Eqs. (6), (7) and (8).

$$W^t = \begin{bmatrix} \hat{F}^t_{i-1,j-1} & \hat{F}^t_{i-1,j} & \hat{F}^t_{i-1,j+1} \\ \hat{F}^t_{i,j-1} & F^t_{i,j} & F^t_{i,j+1} \\ F^t_{i+1,j-1} & F^t_{i+1,j} & F^t_{i+1,j+1} \end{bmatrix} \tag{6}$$

The hat symbol on $\hat{F}^t$ represents the pixel that has been be filtered.

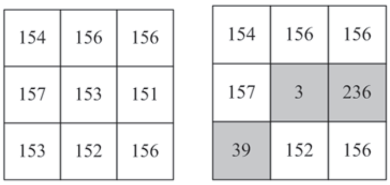
$$W^{t+1} = \begin{bmatrix} F^{t+1}_{i-1,j-1} & F^{t+1}_{i-1,j} & F^{t+1}_{i-1,j+1} \\ F^{t+1}_{i,j-1} & F^{t+1}_{i,j} & F^{t+1}_{i,j+1} \\ F^{t+1}_{i+1,j-1} & F^{t+1}_{i+1,j} & F^{t+1}_{i+1,j+1} \end{bmatrix} \tag{7}$$

$$W^{t-1} = \begin{bmatrix} \hat{F}^{t-1}_{i-1,j-1} & \hat{F}^{t-1}_{i-1,j} & \hat{F}^{t-1}_{i-1,j+1} \\ \hat{F}^{t-1}_{i,j-1} & \hat{F}^{t-1}_{i,j} & \hat{F}^{t-1}_{i,j+1} \\ \hat{F}^{t-1}_{i+1,j-1} & \hat{F}^{t-1}_{i+1,j} & \hat{F}^{t-1}_{i+1,j+1} \end{bmatrix} \tag{8}$$

We observe each element of the W^t window. If one of the elements of the W^t window at (*k,l*) position is indicated as a noise, then it will be replaced by a pixel from the window (W^{t+1}) in the same position, and if the pixel at the same position is also indicated as a noisy pixel, then it will be replaced by the pixel from the window (W^{t-1}) at the same position too. The observation window produces a new window W^n which each element not indicated as a noisy pixel. The window W^n has the same size with window 'W^t'.

3.3 Image Reconstruction

One of the blocks in Fig. 3 is image reconstruction. A free-noise pixel and a replacement pixel for the noisy pixel is placed



154	156	156
157	153	151
153	152	156

(a)

154	156	156
157	3	236
39	152	156

(b)

Fig. 4 The sample pixel value in the 3 × 3 windows from Lena's image 512 × 512. (a) 3 × 3 sample pixel window from the original image, and (b) 3 × 3 sample pixel window from the corrupted image by random impulse noise 50%.

at (*i,j*) position. Replacement for a noisy pixel is obtained from the sum of all element multiplication between window W^n and weight window W^w as in Eq. (9).

$$F^n = \sum_{k=1}^{3}\sum_{l=1}^{3} W^n_{kl} \times W^w_{kl} \tag{9}$$

F^n is a pixel that is used to replace a noisy pixel in the (*i,j*) position. W^n_{kl} is the element matrix of W^n window. While, W^w_{kl} is the element matrix of the weight window W^w.

The total of weight window W^w_{kl} is one. Each weight values in W^w_{kl} depends on the similarity of a pixel in each direction. We calculate the similarity of pixel in each direction by using Absolute Different Value (ADV). The ADV is obtained from window W^n in horizontal, vertical, left and right diagonal directions, as presented in Eq. (10) up to Eq. (13).

$$A^{W^n}_H = |W^n_{21} - W^n_{23}| \tag{10}$$
$$A^{W^n}_V = |W^n_{12} - W^n_{32}| \tag{11}$$
$$A^{W^n}_{LD} = |W^n_{13} - W^n_{31}| \tag{12}$$
$$A^{W^n}_{RD} = |W^n_{11} - W^n_{33}| \tag{13}$$

Window W^w uses the weighted adaptive window. In here, the weighted value is always adapted to the pixel values in the window W^n. The highest weighted value is given to the pixels that have the smallest ADV value and vice versa. In other words, if the ADV in the horizontal direction has the smallest value, the window W^w in this direction is given the highest weighted value. This also applies to the opposite condition. The minimum ADV indicating both of pixels have similarity. Generally, the edge image direction has similarity pixel intensity. Hence, W^w window would be able to smoothing the edge image area, especially for horizontal, vertical, left-diagonal and right-diagonal directions.

Elements of W^w_{kl} are obtained by the following steps:

1 Subtracting 1 with the division of ADV in each direction by the addition of four ADV ($A^{W^n}_H, A^{W^n}_V, A^{W^n}_{LD}$ and $A^{W^n}_{RD}$) plus C^A. In here, $C^A = max[A^{W^n}_H; A^{W^n}_V; A^{W^n}_{LD}; A^{W^n}_{RD}]$. Mathematical formulas to obtain A in each direction and C are written in Eqs. (14) to (18).

$$A_H = 1 - \frac{A^{W^n}_H}{A^{W^n}_H + A^{W^n}_V + A^{W^n}_{LD} + A^{W^n}_{RD} + C^A} \tag{14}$$
$$A_V = 1 - \frac{A^{W^n}_V}{A^{W^n}_H + A^{W^n}_V + A^{W^n}_{LD} + A^{W^n}_{RD} + C^A} \tag{15}$$
$$A_{LD} = 1 - \frac{A^{W^n}_{LD}}{A^{W^n}_H + A^{W^n}_V + A^{W^n}_{LD} + A^{W^n}_{RD} + C^A} \tag{16}$$
$$A_{RD} = 1 - \frac{A^{W^n}_{RD}}{A^{W^n}_H + A^{W^n}_V + A^{W^n}_{LD} + A^{W^n}_{RD} + C^A} \tag{17}$$
$$C = 1 - \frac{C^A}{A^{W^n}_H + A^{W^n}_V + A^{W^n}_{LD} + A^{W^n}_{RD} + C^A} \tag{18}$$

Normalization of each value of A_H, A_V, A_{LD}, A_{RD} and C to obtain the total value equal to 1 as calculated in Eq. (19).

$$\hat{A}_H = \frac{A_H}{(A_H + A_V + A_{LD} + A_{RD} + C)} \tag{19}$$

The same way with Eq. (19), the other normalization value ($\hat{A}_V, \hat{A}_{LD}, \hat{A}_{RD}$ and $\hat{C}$) will be obtained.

2 Further, the weighting W^w matrix can be obtained by Eq. (20).

$$W^w = \begin{vmatrix} \frac{\hat{A}_{LD}}{2} & \frac{\hat{A}_V}{2} & \frac{\hat{A}_{RD}}{2} \\ \frac{\hat{A}_H}{2} & \hat{C} & \frac{\hat{A}_H}{2} \\ \frac{\hat{A}_{RD}}{2} & \frac{\hat{A}_V}{2} & \frac{\hat{A}_{LD}}{2} \end{vmatrix} \tag{20}$$

Finally, the image reconstruction $\hat{F}^t$ is obtained by replacing the pixel which is detected as impulse noise with the pixel calculation results (F^n). Reconstruction in every element of $\hat{F}^t_{ij}$ image can be written in Eq. (21).

$$\hat{F}^t_{ij} = \begin{cases} F^n & \text{if } F^t_{ij} \text{ is a noisy pixel} \\ F^t_{ij} & \text{if free-noise pixel} \end{cases} \tag{21}$$

4. Experimental Results

The test images Akiyo and Xylophone have been used to assess the performance of the proposed method by comparing it with the previous method (SMF, PSMF [3], Non-recursive [7], Recursive [7], LDS [9], DBA [5] and BDND [6]). A simulation result is obtained from MATLAB 7.5.0 release 2007b. The quality of the filtering image is evaluated by quantitative and qualitative parameters as in the subsection below.

4.1 Qualitative Parameter

The visual result of the several filtered image is analysed by a qualitative parameter. We take a sample image of the filtering result using Akiyo image from the impulse noise density ($p = 70\%$) and Xylophone image from the impulse noise density ($p = 50\%$). We use two types of qualitative parameters. Both are the visual result of the filtering image and SSIM index map.

4.1.1 Visual Results of the Filtering Image

Figures 5 and **6** are the visual of the filtering result of Akiyo and Xylophone image, respectively. It presents the several methods as the comparative methods.

Figure 5 (a) shows the clean or uncorrupted image of Akiyo. Figure 5 (b) shows the corrupted frame of Akiyo image sequence by impulse noise ($p = 70\%$). Furthermore, the filtering result of SMF and PSMF [3] are presented in Fig. 5 (c) and (d) respectively. However, both methods are not optimal for reducing the high impulse noise. Applying non-recursive ROM on 11-sample pixel is capable of improving the performance of the SMF. The filtering result is shown in Fig. 5 (e). Meanwhile, the visual result of the recursive method is presented in Fig. 5 (f). The filtering process by the recursive method is capable of resulting in a better image quality than a non-recursive method. The applying of the recursive method in the median filter based on the sorting process using LDS [9] and DBA [5] methods are shown in Fig. 5 (g) and (h) respectively. The DBA [5] is better than the LDS [9] method. However, many artefacts in the edge image area are shown in the DBA [5] method. The jagged edge image in the filtering result is also resulted by Ref. [6] method as shown in Fig. 5 (i). This problem is solved by the proposed method which uses the weighting filter based on pixel similarity in four direction observation kernel. The blur around in the edge area can be minimized by using

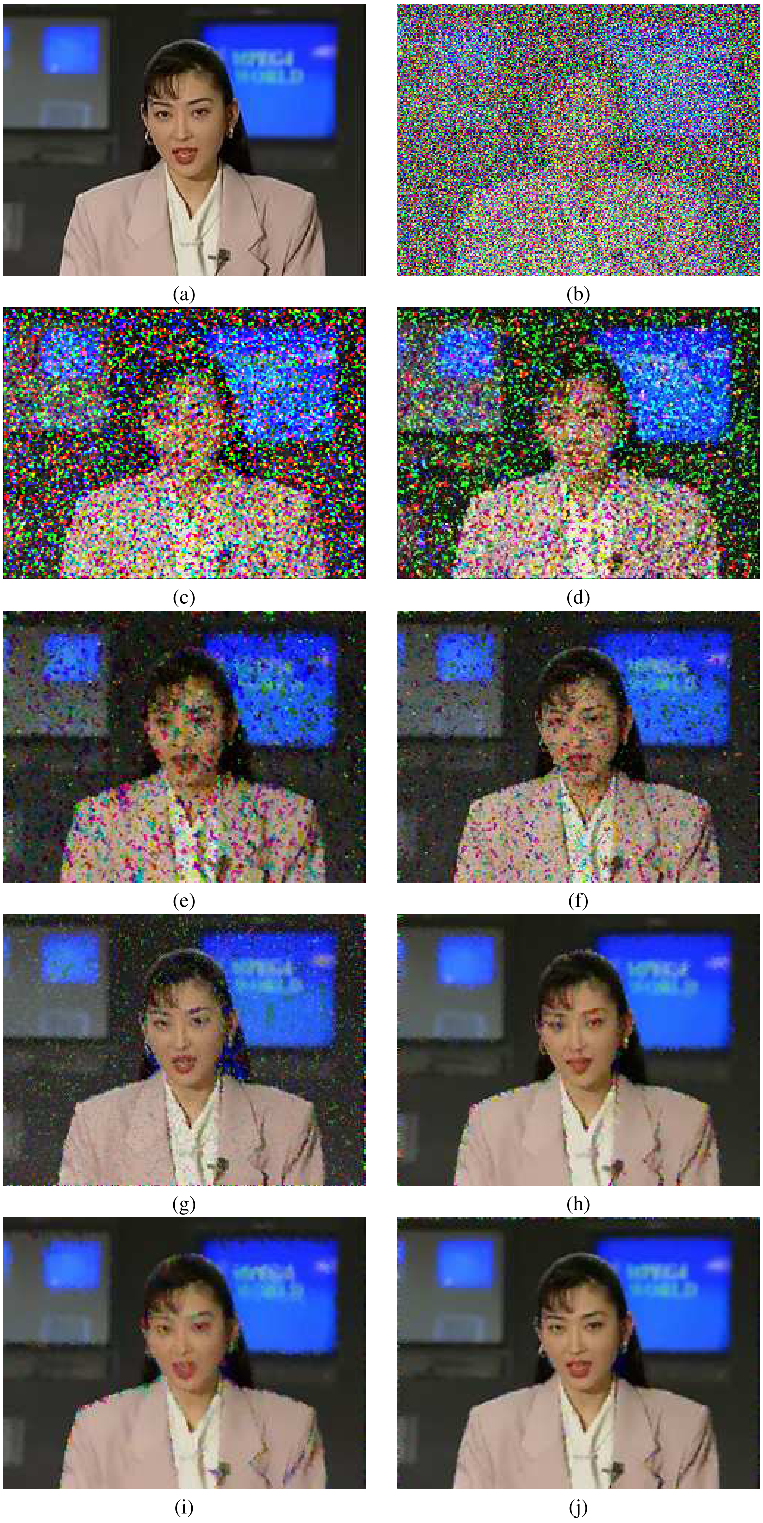


Fig. 5 The filtering results of the corrupted Akiyo image by impulse noise density $p = 70\%$. (a) Clean image, (b) Corrupted image with $p = 70\%$ (c) SMF, (d) PSMF [3], (e) Non-recursive method [7], (f) Recursive method [7], (g) LDS [9], (h) DBA [5], (i) BDND [6], (j) Proposed method.

the weighted sum filter based on ADV. This caused, the similar pixel values are given greater weight value than the other pixels. So that, the contribution of pixels which are similar especially on the pixels in horizontal, vertical, left-diagonal and right-diagonal directions will be larger than others.

The filtering result of the proposed method is shown in Fig. 5 (j). The proposed method looks very significant for eliminating impulse noise. Moreover, an important texture on the image is well maintained.

Figure 6 (a) shows the clean image of Xylophone. Meanwhile, Fig. 6 (b) shows the corrupted frame of Xylophone image sequence by impulse noise ($p = 50\%$). The filtering result of SMF and PSMF [3] are presented in Fig. 6 (c) and (d) respectively. Both methods are only capable of erasing a little noise. Further-

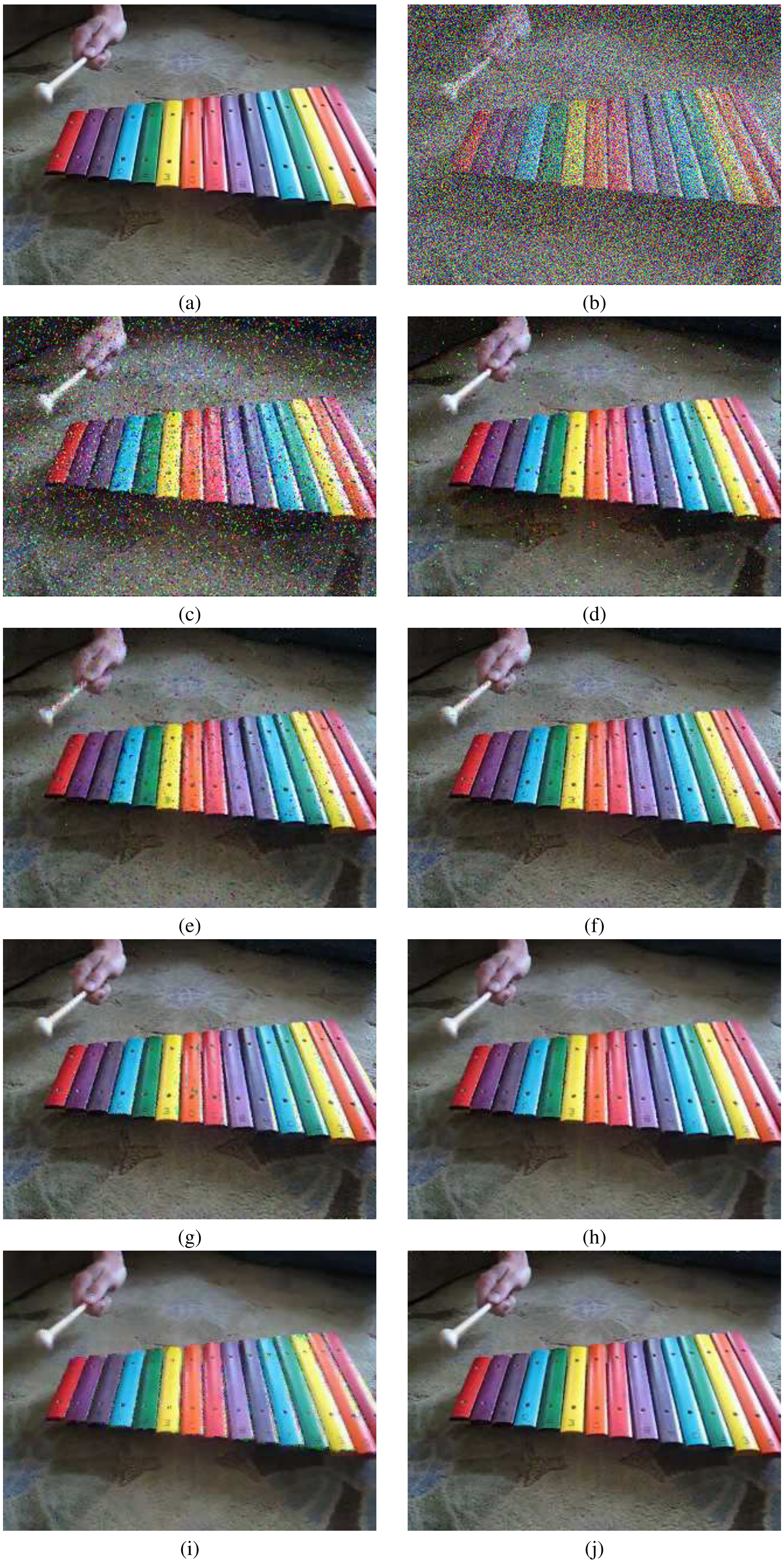


Fig. 6 The filtering results of the corrupted xylophone image by impulse noise density $p = 50\%$. (a) Clean image, (b) Corrupted image with $p = 50\%$ (c) SMF, (d) PSMF [3], (e) Non-recursive method [7], (f) Recursive method [7], (g) LDS [9], (h) DBA [5], (i)BDND [6], (j) Proposed method.

more, Fig. 6 (e) and (f) are the visual result of Non-recursive [7] and Recursive methods [7], respectively. The filtering result of both methods are capable a obtaining a better image quality than SMF and PSMF [3] methods. The filtering results of LDS [9], DBA [5], BDND [6] and the proposed method have almost similar results as shown in Fig. 6 (g), (h), (i) and (j) respectively. However, the proposed method is smoother than LDS [9], DBA [5] and BDND [6] and also all the comparison methods.

4.1.2 SSIM Index Map

In addition, the qualitative measurement is also analysed using SSIM index map. SSIM index map can be viewed as the quality map of the distorted images [11]. The higher illumination of the

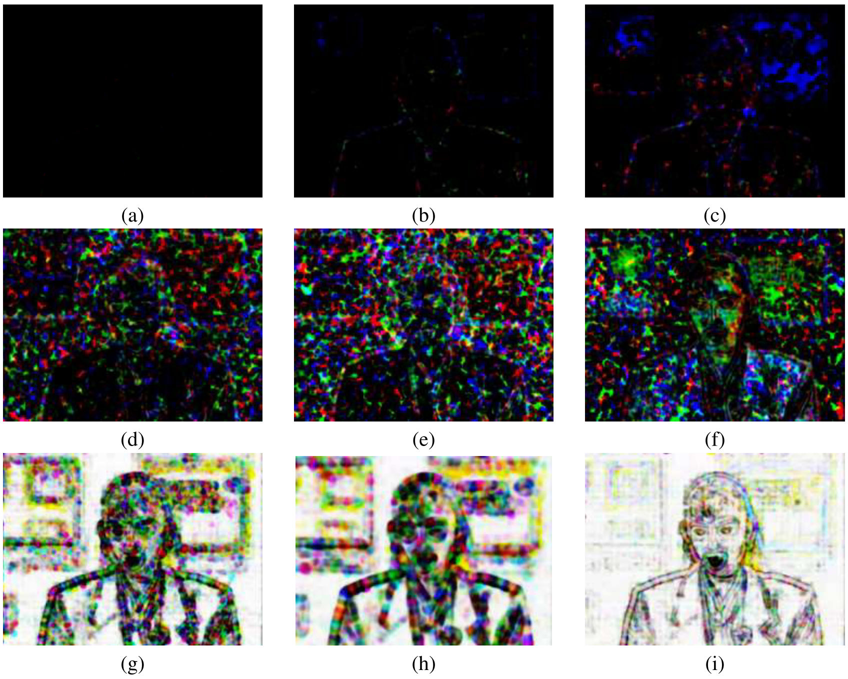


Fig. 7 SSIM-map of the filtering image results in Fig. 5. (a) Corrupted image, (b) SMF, (c) PSMF [3], (d) Non-recursive method [7], (e) Recursive method [7], (f) LDS [9], (g) DBA [5], (h) BDND [6], (i) Proposed method.

SSIM index maps, the better quality of the filtering image result and vice versa [12].

In this paper, we use a constant default value that is suggested in the related paper [13] with the *SSIM* index constants: $K_1 = 0.01$, $K_2 = 0.03$, In here, K_1 and K_2 are the small constant for C_1 and C_2. In here, $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ are constant to avoid instability. L is dynamic range of the pixel values (255 for 8-bit gray-scale images). We also use 11 × 11 Gaussian window size with $\sigma = 1.5$.

$$SSIM(i, f) = \frac{(2\mu_i\mu_f + C_1)(2\sigma_{if} + C_2)}{(\mu_i^2 + \mu_f^2 + C_1)(\sigma_i^2 + \sigma_f^2 + C_2)} \tag{22}$$

μ_i and μ_f are the mean intensity of image i and f respectively. σ_i and σ_f are standard deviation of image i and f respectively. σ_{if} is the covariance.

Implementation of SSIM in RGB image will produce several colors on SSIM index maps. The similarity of the reference and the filtered images will produce a high-intensity or close to white color in the SSIM index maps and vice versa.

(1) If the SSIM index map produces a Red (R) color, it means the similarity pixel value only occurs on the Red (R) side of the filtered image. However, the value pixel of Green (G) and Blue (B) have a different value with the reference image.

(2) If the SSIM index map produces a Green (G) color, it means the similarity pixel value only occurs on the Green (G) side of the filtered image. However, the value pixel of Red (R) and Blue (B) have a different value with the reference image.

(3) If the SSIM index map produces a Blue (B) color, it means the similarity pixel value only occurs on the Blue (B) side of the filtered image. However, the value pixel of Red (R) and Green (G) have a different value with the reference image.

(4) The combination of inequalities on the R, G and B between the reference image and the image filtering results will produce the variety of combination colors (cyan, magenta, yellow, etc.).

Figure 7 shows the SSIM index map result of the Akiyo in the several filter methods at impulse noise density (p = 70%). Figure 7 (a) is the SSIM index map of the corrupted image that has black image results. It means the similarity of the reference and the corrupted images has a low-intensity color. Furthermore, Fig. 7 (b) and (c) are the SSIM index map results from SMF and PSMF [3] methods, respectively. Both images of SSIM index maps have dominant black color and a little blue and red color. Figure 7 (d), (e) and (f) are the SSIM index map results from Non-recursive [7], Recursive [7] and LDS [9] methods, respectively. Here, the images of SSIM index maps also have dominant black color and a little blue, green, red and yellow color. Meanwhile, the edge texture of Akiyo image can be seen more clearly in Fig. 7 (f) compared with Fig. 7 (d) and (e). However, it has low illumination. Figure 7 (g) and (h) have the illumination quality almost similar in both images. However, the illumination result of the proposed method in Fig. 7 (i) is higher than DBA [5] and BDND [6] methods as shown in Fig. 7 (g). On the other hand, the proposed method has the highest illumination of all the comparison methods.

Figure 8 shows the SSIM index map result of the Xylophone in the several filter methods at impulse noise density (p = 50%). Figure 8 (a) is the SSIM index map of the corrupted image that has black image results that result in low intensity. Figure 8 (b) is the SSIM index map of SMF method. It has a lowest illumination than all the comparison methods. Furthermore, Fig. 8 (c), (d), (e) and (f) are the SSIM index map results from PSMF [3], Non-recursive [7], Recursive [7] and LDS [9] methods, respectively.

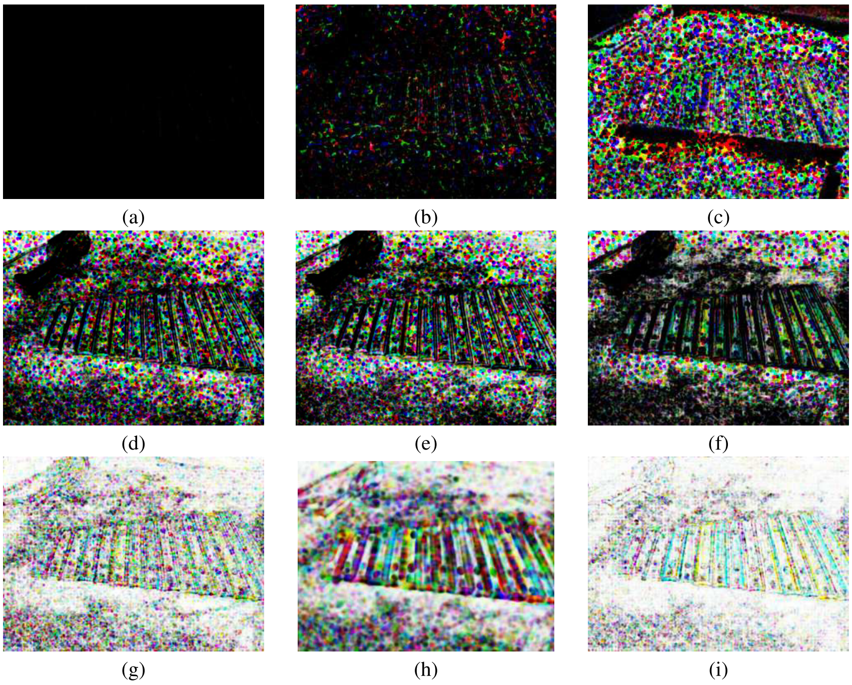


Fig. 8 SSIM-map of the filtering image results in Fig. 6. (a) Corrupted image, (b) SMF, (c) PSMF [3], (d) Non-recursive method [7], (e) Recursive method [7], (f) LDS [9], (g) DBA [5], (h) BDND [6], (i) Proposed method.

Table 1 The PSNR results from Akiyo and Xylophone image sequence.

Methods	Akiyo				Xylophone			
	p = 30%	p = 50%	p = 70%	p = 90%	p = 30%	p = 50%	p = 70%	p = 90%
SMF	22.87	14.34	8.98	5.60	23.57	14.88	9.59	6.18
PSMF [3]	26.21	20.47	11.24	5.64	28.54	21.12	12.58	6.20
Non-recursive [7]	30.71	20.38	13.28	8.00	30.39	20.68	13.63	8.52
Recursive [7]	36.68	29.29	21.60	11.86	37.00	29.51	22.12	12.09
LDS [9]	25.80	23.40	19.31	13.20	26.58	24.71	20.58	14.57
DBA [5]	33.89	30.18	25.27	19.29	34.60	30.33	25.74	20.26
BDND [6]	34.14	29,99	25.99	14.89	33.78	30.49	26.31	15.75
Proposed Method	37.10	33.06	29.11	23.62	37.05	33.70	29.68	24.39

Four images of SSIM index maps have dominant black color and a little blue, green, red and yellow color. However, edge texture of Xylophone image is visible.

Regarding to Fig. 8 (g), (h) and (i) are the results of SSIM index map from DBA [5], BDND [6] and proposed method, respectively. The illumination quality of DBA [5] method is higher than BDND [6]. Meanwhile, DBA [5] and the proposed method have the illumination quality almost similar. However, the illumination result of the proposed method in Fig. 8 (i) is a little bit higher than DBA [5] methods as shown in Fig. 8 (g).

Overall, the proposed method has the highest illumination of all the comparison methods. It means the quality of the filtering image have a good performance result and almost similar to the original image.

4.2 Quantitative Parameter

The quantitative parameter is necessary since the qualitative measurement by visual assessment of an image is subjective. In this evaluation, we use three types of evaluation, namely Peak Signal-to-Noise Ratio (*PSNR*), Mean Structural Similarity (*MSSIM*) index [13] and computational time.

4.2.1 Peak Signal-to-Noise Ratio (*PSNR*)

The first evaluation of image quality uses *PSNR*, which is calculated from the ratio between the maximum possible power of the original image and the image reconstructed in the logarithmic domain, as formulated in Eq. (23).

$$PSNR = 10log_{10}\frac{255^2NM}{\sum_{j=1}^{M}\sum_{i=1}^{N}(I_{ij}-\hat{F}^t_{ij})^2} \tag{23}$$

I_{ij} and $\hat{F}^t_{ij}$ are gray images values of reference image and filtered image, respectively. M and N are the width and height of the image respectively. The bigger *PSNR* value, the better image quality and vice versa.

In this experiment, we add the impulse noise with density values increasing from 30% up to 90%. Variations of *PSNR* values of Akiyo and Xylophone image from different methods are shown in **Table 1**. The proposed method always has a higher *PSNR* value compared with all comparison methods that have been used in this paper. The higher *PSNR* value indicates a high-quality result of the filtering image and vice versa. Here, the SMF, PSMF [3], Non-recursive [7], Recursive [7], LDS [9], DBA [5] and BDND [6] methods are not optimal for reducing the

Table 2 The MSSIM results from Akiyo and Xylophone images sequence.

Methods	Akiyo				Xylophone			
	p = 30%	p = 50%	p = 70%	p = 90%	p = 30%	p = 50%	p = 70%	p = 90%
SMF	0.60	0.14	0.03	0.01	0.74	0.24	0.05	0.01
PSMF [3]	0.70	0.38	0.05	0.02	0.84	0.55	0.15	0.01
Non-recursive [7]	0.92	0.50	0.12	0.02	0.90	0.48	0.12	0.02
Recursive [7]	0.98	0.91	0.65	0.17	0.97	0.89	0.61	0.14
LDS [9]	0.88	0.68	0.29	0.18	0.87	0.76	0.44	0.20
DBA [5]	0.97	0.93	0.83	0.73	0.97	0.92	0.82	0.61
BDND [6]	0.97	0.94	0.86	0.48	0.95	0.91	0.83	0.44
Proposed Method	0.98	0.97	0.93	0.88	0.98	0.96	0.92	0.80

Table 3 The average of computation time per frame in second (s).

Images Sequences	Filtering Methods						
	PSMF [3]	Non-recursive [7]	Recursive [7]	LDS [9]	DBA [5]	BDND [6]	Proposed method
Akiyo	3.057	0.401	0.393	2.289	2.657	12.149	0.028
Xylophone	11.977	1.312	1.315	7.623	8.909	40.296	0.102

high impulse noise density.

4.2.2 Mean Structural Similarity (*MSSIM*) Index

The second image quality evaluation uses *MSSIM* index, which is developed from *SSIM* as illustrated in Eq. (24).

$$MSSIM(I,F) = \frac{1}{U}\sum_{j=1}^{U} SSIM(i_j, f_j) \qquad (24)$$

I and F are the reference and the filtered images, respectively. The i_j and f_j are the image contents at the *j*th local window. U is the number of local windows of the image. We use Eq. (22) to obtain $SSIM$ value.

The response of *MSSIM* value is similar with *PSNR*. *MSSIM* value is ranging between 0 and 1. The higher *MSSIM* index value, the better quality of the filtering image result. Here, the *MSSIM* index value close to 1 indicates that the filtering image result is almost similar with original image.

The *MSSIM* results of Akiyo and Xylophone image in the several methods are shown in **Table 2**. The *MSSIM* index of the proposed method has a higher value than SMF, PSMF [3], Non-recursive [7], Recursive [7], LDS [9], DBA [5] and BDND [6] methods. The *MSSIM* index of the proposed method value is close to one.

4.2.3 Computation Time

In addition, the computation time is also used to calculate the length of the filtering process. The simulation results have been carried out using CPU 3.3 GHz and 4 GB RAM. We calculate the average of the computation time that has been taken from 134 frames.

Referring to **Table 3**, the BDND [6] method has the longest computation time compared to PSMF [3], Non-recursive [7], Recursive [7], LDS [9], DBA [5] and the proposed method. Meanwhile, the computation time process of the proposed method is faster than PSMF [3], Non-recursive [7], Recursive [7], LDS [9], DBA [5] and BDND [6].

The computation time of Xylophone image is longer than Akiyo image on the proposed method. It is caused by the fact that the size image of Xylophone is larger than Akiyo. Xylophone images have the element pixels (640 × 480). Meanwhile, Akiyo images have element pixels (352 × 246).

5. Conclusion

This paper proposes an impulse noise removal method for image sequences. This method uses a weighting filter method based on the similarity pixel via kernel observation. A new window for sample pixel is obtained from the previous frames, current frames and the next frames through kernel observation. The recursive window is applied in the current frame.

As a result, the quality of the filtering result of the proposed method at the high impulse noise density has a better result than SMF, PSMF, Non-recursive, Recursive, LDS, DBA and BDND that have been used as on the reference. The proposed method has a good result, especially for impulse noise removal in terms of qualitative and quantitative parameters.

For future research, we plan to improve the proposed method by combining with Gaussian filtering method, so that becomes a hybrid method for reducing the mixed Gaussian and impulse noise.

References

[1] Smolka, B.: Peer group switching filter for impulse noise reduction in color images, *Int. J. Pattern Recognition Letters*, Vol.31, pp.484–495 (2010).

[2] Utaminingrum, F., Uchimura, K. and Koutaki, G.: High density impulse noise removal by fuzzy mean linear aliasing window kernel, *IEEE International Conference Signal Processing Communication and Computing*, pp.711–716 (2012).

[3] Wang, Z. and Zhang, D.: Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images, *IEEE Trans. Circuits and System-II: Analog and Digital Signal Processing*, Vol.46, No.1 (1999).

[4] Jourabloo, A., Fenghahati, A.H. and Jamzad, M.: New algorithms for recovering highly corrupted images with impulse noise, *Journal of Computer Science Engineering and Electrical Engineering*, Vol.19, pp.1738–1745 (2012).

[5] Srinivasan, K.S. and Ebenezer, D.: New fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises, *IEEE Signal Processing Letters*, Vol.14, No.3, pp.189–192 (2007).

[6] Jafar, I.F., AlNa'mneh, R.A. and Darabkh, K.A.: Efficient improvements on the BDND filtering algorithm for the removal of high-density impulse noise, *IEEE Trans. Image*, Vol.22, No.3, pp.1223–1232 (2013).

[7] Seon, J. and Park, H.W.: Adaptive 3-D median filtering for restoration of an image sequence corrupted by impulse noise, *Signal Processing Image Communication* Vol.16, pp.657–668 (2001).

[8] Sucher, R.: A recursive nonlinear filter for removal of impulse noise, *Image processing*, pp.183–186 (1995).

[9] Rajamani, A., Krishnaveni, V., Ferose, H.W. and Kalaikamal, M.: A noise denoising approach for the removal of impulse noise from color images and video sequences, *Journal of Image Anal Stereol*, Vol.31, pp.185–191 (2012).

[10] Kong, H. and Guan, L.: A neural network adaptive filter for the removal of impulse noise in digital image, *Neural network*, Vol.9, pp.993–1003 (1996).

[11] Li, C. and Bovik, A.C.: Three-component weighted structural similarity index, *Proc. SPIE*, Vol.7242, pp.1–9 (2009).

[12] Utaminingrum, F., Uchimura, K. and Koutaki, G.: An adaptive Sliding Window based on Fuzzy Filter for removing Impulse Noise Densities on the image sequence, *Proc. IEEE/IIAE International Conference on Intelligent Systems and Image Processing*, pp.111–118 (2013).

[13] Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Processing*, Vol.13, No.4, pp.600–612 (2004).

Fitri Utaminingrum received bachelor degree in Electrical Engineering from National Institute of Technology Malang Indonesia in 2000–2004, and Master degree in the same major from Brawijaya University, Indonesia in 2004–2007. She has been full time lecturer in Informatics Department of Brawijaya University, Malang-Indonesia from 2008. Her research interests in image processing application on the filtering process. She is current PhD student in Kumamoto University, Japan. She was supported by Directorate of Higher Education (DGHE) of Indonesia.

Keiichi Uchimura received his B.Eng. and M.Eng. degrees from Kumamoto University, Kumamoto, Japan, in 1975 and 1977, respectively, and Ph.D. degree from Tohoku University, Miyagi, Japan, in 1987. He is currently a Professor with the Graduate School of Science and Technology, Kumamoto University. He is engaged in research on intelligent transportation systems, and computer vision. From 1992 to 1993, he was a Visiting Researcher at McMaster University, Hamilton, ON, Canada. His research interests are computer vision and optimization problems in the Intelligence Transport System.

Gou Koutaki received his B.Eng., M.Eng. and Ph.D. degrees from Kumamoto University, Kumamoto, Japan, in 2002, 2004, and 2007, respectively. From 2007 to 2010, he was with Production Engineering Research Laboratory, Hitachi Ltd. Currently, he is on the Priority Organization for Innovation and Excellence at Kumamoto University. He is engaged research on image processing and pattern recognition of the Intelligence Transport System.