A Study on Fast Removal Method of Impulsive Noise Using Parallel Processing with GPU

Takashi Miyazaki^a, Yusuke Koshimura^{a,*}, Yasushi Amari^a, Yasuki Yokoyama^a, Hiroaki Yamamoto^b

^aNational Institute of Technology, Nagano College, 716 Tokuma, Nagano city, Nagano Prefecture and 381-8550, Japan

^bShinshu University, 4-17-1 Wakasato, Nagano city, Nagano Prefecture and 380-0928, Japan

*Yusuke KOSHIMURA: 17906@g.nagano-nct.ac.jp

Takashi MIYAZAKI: miya@nagano-nct.ac.jp

Abstract

The median filter is used to remove image degradation due to impulsive noise. However, since this process is also applied to non-noise pixels, image quality deteriorates. Therefore, the Switching median filter for determining whether the pixel of interest is noise is proposed as an effective method. In addition, several improved methods have been proposed, one of which is Multi-directional Switching Median Filter developed by our laboratory. In this method, detection and removal of noise candidate pixels from four directions are performed as the first step by using a 2x2 detection operator, and density values are corrected by averaging processing of the second step, to restore the image. But, with this method, it takes time to remove noises in the high resolution image. A parallelized method to improve processing speed was developed by allocating pixels of this Multi-directional Switching Median Filter to threads of the GPU.

For the difference in noise rate and image size, image denoising was performed on five test images by the proposed method and some conventional methods. In this paper, we report the effectiveness of our proposed method for in high resolution images from the results of comparative experiment.

Keywords: image denoising, Switching median filter, GPU, parallel algorithm.

1. Introduction

When processing a digital image, impulsive noise is superimposed on the digital image by imaging with C-MOS

or CCD under low illuminance or influence of the variation in sensitivity among light receiving cells due to high resolution of sensor, and deterioration is likely to occur. In removing these noises, the median filter method (MF) is said to be effective ^(1,2). However, since this method performs processing on all pixels, there is a problem of degrading non-noise pixels. As a method to solve this problem, switching median filter method (SMF) using switching type noise reduction filter has been proposed. In this method, the noise non-noise of the pixel of interest is determined by the threshold value, and denoising by MF processing only for the noise. Therefore, degradation of non-noise pixels caused by MF is able to be suppressed and both denoising and edge preservation are able to be achieved ⁽³⁻¹²⁾.

Many SMFs tend to complicate algorithms to increase noise detection and removal performance. Therefore, there is a tendency that it takes more time for removal processing than MF. However, from the perspective of hardware implementation, a simpler algorithm is preferred. So, Mr. Yokoyama proposes a multidirectional scanning type SMF (Multi-directional SMF) using a 2x2 small noise detection operator ⁽⁸⁾. Although this method is a simple algorithm, it has both original signal preservability and noise elimination capability, making it possible to denoise high quality.

Furthermore, in recent years, the resolution of digital images such as 4k and 8k has been remarkably improved, so that the processing time accompanying the noise removal has also increased. And, there is a technology to perform many parallel processing using GPU called GPGPU. When similar processing is performed for all pixels like MF, this GPGPU is greatly able to speed up. The feature of the Multi-directional SMF is suitable for parallel processing since the

density difference by the 2x2 noise detector is processed from different directions. In this paper, we report on realization of high speed by parallel processing of Multi-directional SMF GPU.

2. Conventional method

2.1 Multi-directional SMF

Multi-directional SMF is a simple algorithm that the density difference of interest pixel calculated using a small detector of 2x2 size, which is characterized by a small calculation amount. Fig.1 shows an overview of the Multi-directional SMF in 4 scanning directions with image quality priority. Multidirectional SMF consists of two steps: one is a processing of noise detection and removal with a MF, the other is an average processing for improving image quality of the restored image.

2.1.1 First step

At the time of noise removal, since the restored pixel is reflected on the original image, there is a noise removed area and a noise pre-removal area in the MF window. Therefore, in the noise detector of 2x2 size, other pixels except the target pixel are non-noise pixels or noise removal pixels. Further, since the pixel information used for each scanning direction is different, removal is performed from a plurality of scanning directions, thereby reducing bias.

Duplicate target image deteriorated due to noise is duplicated by the number of scanning direction. The proprietary SMF with 2x2 noise detection operator is recursively applied to the duplicated images from 4 different scanning directions. Calculation of the density difference

value is calculated by multiply-add operation with the noise detection operator. The density different value g_{ij} of the target pixel is expressed by

$$D = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad (1)$$

$$g_{ij} = |X_{ij}^{WD} * D| \quad (2)$$

where X_{ij}^{WD} is the set of pixels in the window of the detector, D is the detecting operator, and $*$ is the product sum operator.

The comparison is made by comparing the density difference value g_{ij} with the set threshold value.

2.1.2 Second step

The noise detection result in the first step is different for each scanning direction, and the obtained noise reduction image is also different. This is because the noise detection window is small relative to the MF window. For these different noise reduced images, a more excellent removed image is obtained by calculating the density average of the positions. By performing this processing, it is possible to suppress the possibility that the effect of erroneous detection, which is likely to occur in scanning in only one direction, becomes an average processed pixel in noise determination in which noise due to multidirectional scanning and non-noise.

3. Outline of GPGPU

3.1 CUDA

This section describes CUDA (Compute Unified Device Architecture)^(13,14). CUDA is a general purpose parallel computing platform and programming model for GPU. We call GPGPU (General-Purpose computing on Graphics Processing Units) a technology that uses GPU for general purpose computation. CUDA's architecture includes new components designed for GPU computing, and the limitations so far have been overcome. This architecture is not specialized in graphics, it is designed to use an instruction set prepared for general purpose computation. CUDA C is cross-platform and has openness to accommodate calls from various languages. While GPU is good at simple and parallel executable calculations, it has a feature that is not good at calculation with many branches and calculation with low parallel degree. In this paper, we develop programs with CUDA 8.0.

Fig.1. Processing outline of multi-directional. SMF

3.2 Program composition

The CUDA program performs calculation using both CPU and GPU. A CPU called a host reads a program or data stored in the main memory, processes the data according to an instruction described in the program, and writes the result in the main memory. On the other hand, when the GPU performs processing, it reads the program and data from the video memory, and writes the result to the video memory after processing. GPU and video memory as "device", programs to run as GPU is called "kernel". Hosts and devices are handled independently of each other. In other words, in processing by CUDA, data is first recorded in the host memory, and then its contents are transferred to the corresponding device memory.

3.3 Thread structure

Execution on the device side starts by calling the kernel function from the host side. A large number of threads are generated in the device, and statements defined by the kernel function are executed by each thread. A thread is the minimum unit for executing a program when the kernel is operated.

In the proposed method in this paper, parallelization is performed by associating one pixel of an image with this thread.

4. Parallel Denoising Algorithm using GPGPU

In this algorithm, each thread corresponds to each pixel in the image, and since the kernel statement is executed for each thread, the processing of removal noise is performed almost simultaneously for all the pixels. However, this parallel processing algorithm makes it difficult to perform recursive processing and sequential removal processing of

the algorithm processed by using only CPU such as Multi-directional SMF. Therefore, we try to improve the restored image by using iterative processing instead of that two process. In the parallelization method GPUSMF using GPU proposed by Koshiyama and others, in order to reproduce the operation of the Multidirectional SMF on the GPU, the noise image is duplicated on four sheets and the processing by the noise detection operator corresponding to the Multi-directional SMF. The reason why duplication is made on four images is to perform the same processing as the Multi-directional SMF from four directions at the same time. After that, iterative processing is performed 15 times, and the four different removal results obtained are averaged to obtain a restored image⁽¹⁵⁾.

However, in GPUSMF, the number of times to use MF for noise removal increases and processing time increases due to averaging processing. Fig. 2 shows the procedures of the proposed algorithm which was revised from GPUSMF.

In this proposed method, instead of raster scanning for noise detection from different directions of Multi-directional SMF, detection operators corresponding to four directions are used for the target pixel at the same time as the parallel processing. When one or more detection operators detect the noise from these four directions, the MF processing is performed with the target pixel as the center pixel. The features of the proposed algorithm is reduces the four directional processing of MF executed in the GPUSMF method and eliminates the averaging processing that was carried out for improving the quality of the restored image. The algorithm depending on eliminating these two processing can be simplified and the processing speed can be improved. The main part of the algorithm is shown below.

Algorithm (i, j)

- 1: $d_1 \leftarrow |x(i, j) + x(i - 1, j - 1) - x(i - 1, j) - x(i, j - 1)|$
- 2: $d_2 \leftarrow |x(i, j) + x(i + 1, j - 1) - x(i + 1, j) - x(i, j - 1)|$
- 3: $d_3 \leftarrow |x(i, j) + x(i + 1, j + 1) - x(i + 1, j) - x(i, j + 1)|$
- 4: $d_4 \leftarrow |x(i, j) + x(i - 1, j + 1) - x(i - 1, j) + x(i, j + 1)|$
- 5: if $d_1 \geq T_D \mid d_2 \geq T_D \mid d_3 \geq T_D \mid d_4 \geq T_D$ then
- 6: $y_1 \leftarrow \text{MED}(X_{i,j})$
- 7: else
- 8: $y_1 \leftarrow x(i, j)$
- 9: end if

Fig.2. Outline of proposed method.

5. Experiments and Results

5.1 Experimental method

- (i) Superimpose a random value as impulsive noise on the test image
- (ii) Remove noise by the proposed method and remove noise by the conventional methods.
- (iii) Evaluate the speed of removal processing.
- (iv) Evaluate the quality of restored image.

In addition, in order to investigate the differences and the effectiveness of each technique accompanying the increase in resolution, prepare three sizes of test images 512x512, 1024x1024, 2048x2048. For the three kinds of noise overlap rates of 10%, 30%, and 50%, the differences of the restored image is examined by the differences of the image resolution.

5.2 Superimposed noise and evaluation index

We describe a type of noise used in the experiment. In the work of noise removal, it is normally used two types of noise, one is salt-and-pepper noise, and the other is random value noise. Fig. 4 shows two types of noise model which have the density variation rate of the noise. Fig. 4 (a) is the salt-and-pepper noise which has only two values of 0 or 255 as a density value of superimposed noise, and Fig. 4 (b) is the random value noise which takes the value from 0 to 255. The random noise level V within a certain gradation is expressed by

$$V = v(U_{max} - U_{min}) \quad (3)$$

Fig.3. Test images.

Here, v is the gradation variation rate, U_{max} is the maximum gradation value and U_{min} is the minimum value that can be expressed in the image. Therefore, this type noise is a model close to actual noise.

When the pixel position is (i, j) , the signal of the degraded image is $x(i, j)$, and the signal of the original image is $x_0(i, j)$, the noise model is defined by

$$x(i, j) = \begin{cases} x_0(i, j) & : p = q - 1 \\ \text{RND}(U_{max}, -V) & : p = q/2 \\ \text{RND}(U_{min}, V) & : p = q/2 \end{cases} \quad (4)$$

Here, $\text{RND}(c, d)$ is a uniform random number that takes the value of the interval $(c, c + d)$, p is the possibility that the value is selected, and q is the noise rate. That is, the salt-and-pepper noise of Fig. 4 (a) is $v = 0.0$.

Next, we describe the evaluation index in the evaluation of restored image quality. PSNR (Peak Signal to Noise Ratio) is used in this experiment. The formula used for PSNR calculation is shown as

$$\text{PSNR}[\text{dB}] = 10 \log_{10} \frac{255^2}{\text{MSE}} \quad (5)$$

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - y_{ij})^2 \quad (6)$$

(a) Salt and Pepper Noise

(b) Random Value Noise

Fig.4. Noise model.

Here, MSE (Mean Square Error) is the mean square error between the original image and the restored image. The number of pixels in the horizontal direction is M , the vertical direction is N , the current pixel density of the coordinates (i, j) is x_{ij} , and the restored pixel density is y_{ij} . Image quality estimation using PSNR which is popular in image processing research takes a higher value as it is closer to the error between the origin image and the restored image.

5.3 Experiment results

Fig. 5 is a partial enlargement of the image to which Fig. 3 (a) is added with noise. These image sizes are 2048x2048. Fig. 6 shows the result of removing Noise 30 % image's noise by each method. These degrees of restoration correspond to the graph of Fig. 8.

The results of comparing the image quality of the noise-removed image and its processing speed by this method and

Fig.5. Noise image examples.

Fig.6. Noise removal image by each method (Noise ratio 30%).

Fig.7. Noise ratio 10%.

Fig.10. Noise ratio 10%.

Fig.8. Noise ratio 30%.

Fig.11. Noise ratio 30%.

Fig.9. Noise ratio 50%.

Fig.12. Noise ratio 50%.

another method are shown in Figures 7 - 12. The image quality of each method for image size (256×256, 1024×1024, and 2048×2048) is evaluated by PSNR with the noise addition rate kept constant at 10%, 30%, and 50%, as shown in Figures 7, 8, and 9. The evaluation value indicates the average of the image qualities of five test images. Figures 10, 11, and 12 show the processing times using the respective methods for each image size. These figures also show the average value of the time taken for noise removal of five images, similarly to the image quality.

From Figures 7, 8, and 9, it was found that the removal results by the proposed method are superior to GPUSMF for the 2048 x 2048 size image. Moreover, it was found that it is effective even when the noise rate is high. For the high quality image with the noise rate of 10% shown in Fig. 7, the degree of restitution of MF is high. In this experiment, since the set threshold value is fixed except MF, it is considered that the MF which does not use the threshold becomes high of restored image quality. In addition, as the boundary pixel width such as edges increases in the high-quality image, the possibility that edge degradation which is a problem of MF processing may be suppressed is considered.

From the results of Figures 10, 11, and 12, it was possible to suppress increase of processing time by the proposed method even due to increase of noise rate. In particularly, the processing speed time of 2048×2048 images at a noise rate of 30% or more can be improved to less than half compared with parallel GPUSMF using parallel processing.

6. Conclusions

In this paper, we proposed a novel noise removal method which was revised the condition of using median filter in GPUSMF method. As for the time of noise removal processing and the quality of the restored image, the proposed method was shown the superior and effective performances from the results of the comparison with other methods. In particularly, as the image size increases, its effect becomes prominent remarkably. In recent years, as the resolution of images and images increases as high as 4 K or 8 K, it can be a very effective noise elimination method.

However, the noise removal rate can still be improved because the threshold value is kept the default value while the denoising process. Therefore, it is necessary to improve to a method that can be varied to a more appropriate threshold value using peripheral information of the target pixel. Moreover, if the number of repetitions can be reduced, we

think that the processing speed can be further improved.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 16K00260 and 17H00383.

References

- (1) Kazuhiro FUJITA, and Mitsuji MUNEYASU, Nobuyuki NAKAMORI, Hiroshi KONDO, Riti NAKURA, Nobuyuki WATANABE, Ryo HONEN : “Restoration of deteriorated image · Improvement of image quality by noise removal”, Torikeppusu, Vol. 1, No. 1, pp. 50-63, 2008.
- (2) Narendra, M.Patrenahalli : “A Separable Median Filter for Image Noise Smoothing”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 1, pp.20-29, 1981.
- (3) Image processing standard textbook editing committee : “Image processing <Image processing standard text book>”, CG-ARTS, Vol. 1, No.1, pp. 165-181, 2010.
- (4) Tong Sun and Yrjo Neuvo : “Detail-preserving median based filters in image processing”, Pattern Recognition Letters, Vol. 15, No. 4, pp.341-347, 1994.
- (5) Z. Wang and D. Zhang : “Progressive switching median filter for the removal of impulse noise from highly corrupted images”, IEEE Trans. Circuits Syst. II CAS II, vol.46, No.1, pp.78-80, 1999.
- (6) S.Zhang and M.A.Karim : “A new impulse detector for switching median filters”, IEEE Signal Process. Lett., Vol9, No.11, pp.360-363, 2002.
- (7) B.Xiong and Z.Yin : “A universal denoising framework with a new impulse detector and nonlocal means”, IEEE Trans. Image Process., Vol.21, No.4, pp.1663-1675, 2012.
- (8) Yasuki YOKOYAMA, Takashi MIYAZAKI, Mitsuo SONE, et al. : “Switching Median Filter Using Multi-direction Scanning 2x2 Noise Detector and Averaging Method”, IEICE Trans(A), Vol. J95-A, No. 10, pp.737-750, 2012.
- (9) Yuhei HASHIMOTO, Yoshinobu KAJIKAWA, Yasuo NOMURA : “Removal of Impulse Noise Using Impulse Noise Position Information by High Performance Impulse Detector”, IEICE Trans(A), Vol.J84-A, No.6, pp.759-768, 2001.
- (10) D.R.K. Browning : “The weighted median filter”, Comm. ACM, Vol.27, No.8, pp807-818, 1984.

- (11) R.H.Chan, C.-W.Ho, M.Nikolova : “Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization”, IEEE Trans. Image Processing, Vol.14, No.10, pp.1479-1485, 2005.
- (12) S.-S Wang, C.-H Wu : “A New Impulse Detection and Filtering Method for Removal of Wide Range Impulse Noises”, Elsevier. Pattern Recognition, Vol.42, Issue9, pp.2194-2202, Jan.2009.
- (13) J. Sanders and E. Kandrot, CUDA by Example: CUDA by Examples, An Introduction to General-Purpose GPU programming, Addison-Wesley, 2011.
- (14) Masatomo INUI : “Introduction to GPU parallel graphics processing - Introduction and utilization of CUDA · OpenGL”, Technical commentary, pp114-136, 2014.
- (15) Shota KOSHIYAMA, Hiroaki YAMAMOTO, Takashi YAMAZAKI, Yasuki YOKOYAMA, Mitso SONE : “A Fast GPU-Based Parallel Multi-Direction Switching Median Filter”, IEICE Trans(A), Vol.J99-A, No.1, pp69-72, 2016.