ICIC Express Letters
Volume **13**, Number **1**, January **2019**
 ISSN 1881-803X
pp. 35–40

ACCELERATION USING LOOKUP TABLE AND SAMPLING FOR GENERATING STRIPE-PATCHWORK IMAGES

Toru Hiraoka[1] and Hirohumi Nonaka[2]

[1]Department of Information Systems
University of Nagasaki
1-1-1, Manabino, Nagayo-chou, Nishisonogi-gun, Nagasaki-ken 851-2195, Japan
hiraoka@sun.ac.jp

[2]Department of Information and Management Systems Engineering
Nagaoka University of Technology
1603-1, Kamitomioka-cho, Nagaoka, Nigata-ken 940-2188, Japan
nonaka@kjs.nagaokaut.ac.jp

Received August 2018; accepted October 2018

Abstract. *A non-photorealistic rendering method has been proposed for generating a stripe-patchwork image from a photographic image by inverse filtering and smoothing according to the distance from the target pixel. The stripe-patchwork image imitates stripe patchwork, and locally changes stripe-patchwork patterns in accordance with shading and edges of the photographic image. In this paper, we develop a method to accelerate the processing of the conventional method. Our method is based on three ideas. The first idea is to use a lookup table that calculates the distance from the target pixel, the second idea is to sample the pixels used in the calculation, and the third idea is to combine the first and second ideas. To verify the effectiveness of our method, experiments are conducted using Lenna image. In the experiments, we show that our method can make the process faster than the conventional method.*
Keywords: Non-photorealistic rendering, Stripe patchwork, Acceleration, Lookup table, Sampling

1. **Introduction.** Non-photorealistic rendering methods have been proposed for generating stripe-patchwork images from photographic images [1, 2, 3]. Non-photorealistic rendering is a technique of computer graphics that creates non-photorealistic images [4, 5, 6, 7, 8]. Stripe-patchwork images imitate stripe patchwork, and locally change stripe-patchwork patterns in accordance with shading and edges of photographic images. The conventional method [1] is executed by an iterative process using inverse filtering [9] and k-means clustering, the conventional method [2] is executed by an iterative process using inverse filtering and entropy, and the conventional method [3] is calculated by an iterative process using inverse filtering and smoothing according to the distance from the target pixel. Among these three methods, the conventional method [3] can perform processing at the highest speed. Among these three stripe-patchwork images, the conventional method [3] can best preserve brightness and edges of photographic images.

In this paper, we focus on the conventional method [3], which is hereinafter referred to as the conventional method. We develop a method to accelerate the processing of the conventional method. For example, the acceleration of the processing is necessary for processing by general users on portable terminals such as smart phones and tablet terminals. In addition, if the high-speed processing can be realized, work efficiency of professional designers can be improved. The proposed method is based on three ideas. The first idea uses a lookup table that calculates the distance from the target pixel, the second idea

DOI: 10.24507/icicel.13.01.35

samples the pixels used in the calculation, and the third idea combines the first and second ideas. To verify the effectiveness of the proposed method, experiments are conducted using Lenna image. In the experiment of computational time, we show that the proposed method can make the process faster than the conventional method. Since it is considered that a difference occurs between stripe-patchwork images of the conventional method and the second and third ideas, we conduct the quantitative appearance experiment.

The rest of this paper is organized as follows. Section 2 outlines the conventional method. Section 3 describes the proposed method with the three ideas. Section 4 shows experimental results, and reveals the effectiveness of the proposed method. Finally, Section 5 concludes this paper.

2. **Conventional Method.** The conventional method generates stripe-patchwork images from photographic images, and is executed by the iterative process using inverse filtering and smoothing according to the distance from the target pixel. The pixel values on coordinates (i, j) of the gray-scale photographic image are defined as $f_{i,j}$ $(i = 1, 2, \ldots, I;$ $j = 1, 2, \ldots, J)$. The pixel values $f_{i,j}$ have value of 256 gradation from 0 to 255.

First, the pixel values $f_{i,j}$ are smoothed to the pixel values $SM(f_{i,j})$ as

$$SM(f_{i,j}) = \frac{\displaystyle\sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(1 - \frac{k^2 + l^2}{D}\right) f_{i+k,j+l}}{\displaystyle\sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(1 - \frac{k^2 + l^2}{D}\right)} \tag{1}$$

$$D = \sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(k^2 + l^2\right) \tag{2}$$

where W is the window size.

Second, the pixel values $f_{i,j}^{(t)}$ using inverse filtering are computed as

$$f_{i,j}^{(t)} = f_{i,j}^{(t-1)} - SM\left(f_{i,j}^{(t-1)}\right) + f_{i,j} \tag{3}$$

where t $(t = 1, 2, \ldots)$ is the number of iterations and $f_{i,j}^{(0)} = f_{i,j}$. The pixel values $f_{i,j}^{(t)}$ are set to 0 if their values are less than 0, and set to 255 if their values are greater than 255.

Finally, the stripe-patchwork image is obtained after T times iteration of the first and second processes.

3. **Proposed Method.** The proposed method accelerates the processing of the conventional method with three ideas (Figure 1). Idea 1 uses the lookup table that calculates the distance from the target pixel. Idea 2 samples the pixels used in the calculation. Idea 3 combines Ideas 1 and 2. We show details of each idea as follows.

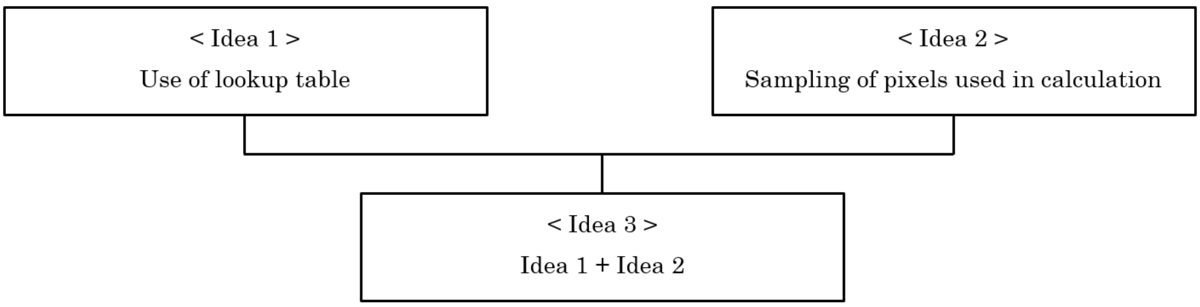


Figure 1. Ideas of the proposed method

Idea 1: We calculate all combinations of $1-\frac{k^2+l^2}{D}$ in Equation (1) at the first, and store the calculation result in memory (lookup table). When we calculate Equation (1), we reduce the calculation of $1-\frac{k^2+l^2}{D}$ by calling from memory. The calculation of Idea 1 is $(2W+1)^2$ times. On the other hand, the conventional method requires $IJ(2W+1)^2$ calculation times. The difference does not occur between stripe-patchwork images of Idea 1 and the conventional method.

Idea 2: In the calculation of Equation (1), we reduce the computational complexity by sampling the pixels in the window. By shifting the start pixel position in the window according to the iterative number t, the sampling scans in interval D from the upper left to the lower right in the window. In Equation (1), the computational complexity of Idea 2 is $1/D$ of that of the conventional method. The difference occurs between stripe-patchwork images of Idea 2 and the conventional method.

Idea 3: Ideas 1 and 2 are combined (Figure 2). Idea 3 performs the reduction in the calculations by lookup table and sampling the pixels in the window. The same difference as Idea 2 occurs between stripe-patchwork images of Idea 3 and the conventional method.

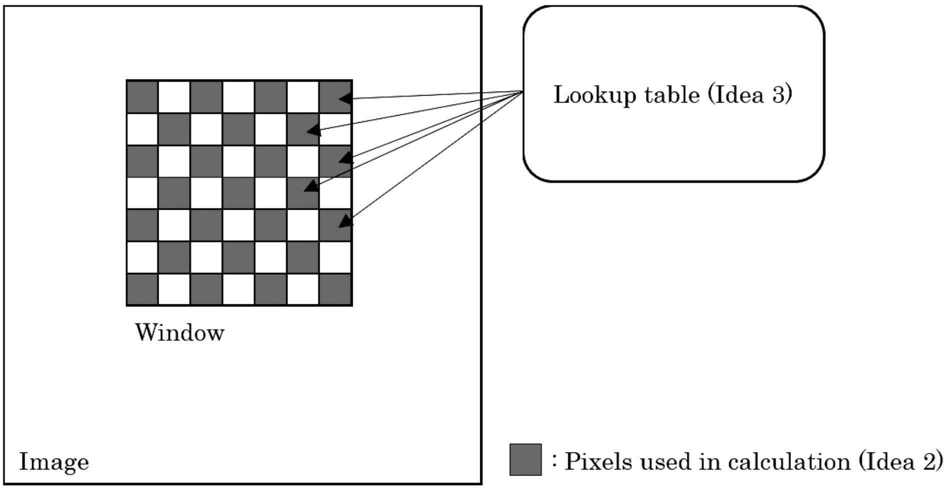


FIGURE 2. Conceptual diagram of Idea 3

4. **Experiments.** We conduct all experiments using Lenna image (Figure 3) with $512 * 512$ size and 256 gradation, and set the parameter T to 40 based on [3]. The computing environment for all experiments is Windows 10 Enterprise 2016 LTSB operating system on a computer with 3.29 GHz CPU, 8.00 GB of memory, and Intel(R) HD Graphics 530 graphics board.

FIGURE 3. Lenna image

First, we compare the calculation time between the proposed method and the conventional method. In the experiments of the conventional method and Idea 1, we set the window size W to 1, 2, 3, 4, and 5. In the experiments of Idea 2 and Idea 3, since the number of pixels that can be used in the calculation of Equation (1) is reduced, we set the window size W to 3, 4, and 5. And, we set the interval number D to 2, 3, 4, and 5 in the experiments of Idea 2 and Idea 3. The results of the experiments of the conventional method, Idea 1, Idea 2, and Idea 3 are shown in Tables 1, 2, 3, and 4, respectively. Ideas 1, 2, and 3 can make the process faster than the conventional method. As the window size W of Ideas 1, 2, and 3 is larger, the effect of acceleration is larger than the conventional method. As the interval number D of Ideas 2 and 3 is larger, the calculation time becomes shorter. Idea 3 has shorter calculation time than Idea 2.

Next, since it is considered that the difference occurs between stripe-patchwork images of the conventional method and Idea 2, we conduct the quantitative appearance experiment. In the quantitative experiment, we calculate the mean of the absolute difference

Table 1. Calculation times of the conventional method [seconds]

Window size W	Calculation time
1	1.199
2	3.519
3	6.951
4	11.560
5	17.315

Table 2. Calculation times of Idea 1 [seconds]

Window size W	Calculation time
1	0.770
2	1.750
3	3.446
4	5.306
5	7.702

Table 3. Calculation times of Idea 2 [seconds]

Window size W	Calculation time			
	Interval number D			
	2	3	4	5
3	3.661	2.558	2.131	1.984
4	5.953	4.102	3.458	3.194
5	8.866	6.023	5.019	4.776

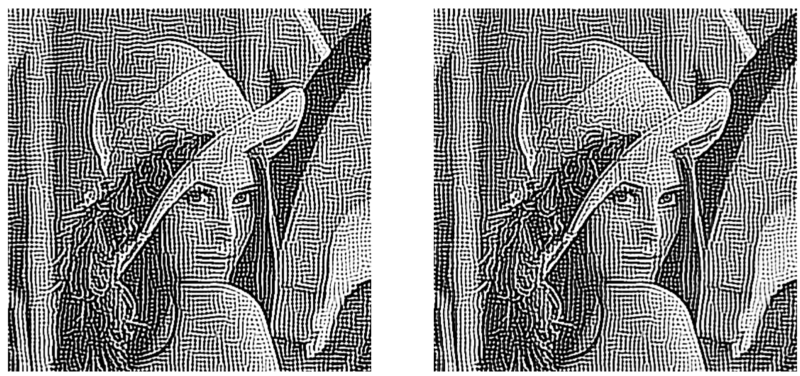
Table 4. Calculation times of Idea 3 [seconds]

Window size W	Calculation time			
	Interval number D			
	2	3	4	5
3	2.547	2.172	2.031	1.922
4	4.332	3.703	3.203	3.062
5	6.100	5.250	4.719	4.484

in pixel value between stripe-patchwork images of the conventional method and Idea 2 (hereinafter, the difference average). The smaller the value of the difference average is, the more stripe-patchwork images of the conventional method and Idea 2 are similar. The results of the experiment are shown in Table 5. As the window size W is larger, the difference average decreases. As the interval number D is larger, the difference average increases. When $W = 4$ and $D = 3$, the difference average exceptionally gets bigger. It is considered that the selected pixels in the window are arranged in three vertical rows. We think that it is better that the selected pixels in the window are randomly arranged. In order to visually check the stripe-patchwork image, stripe-patchwork images of the conventional method and Idea 2 when $W = 4$ and $D = 3$ are shown in Figure 4, respectively. We cannot see the difference between two stripe-patchwork images. We think that there is no large difference between stripe-patchwork images of the conventional method and Idea 2. Since the difference average of Idea 1 is 0 and Idea 3 is affected by the difference averages of ideas 1 and 2, Ideas 2 and 3 have the same result.

Table 5. Difference average of Idea 2

Window size W	Difference average			
	Interval number D			
	2	3	4	5
3	2.980	4.315	7.157	10.121
4	1.489	10.195	2.896	5.448
5	0.890	1.260	1.671	2.636



(a) Conventional method (b) Idea 2

Figure 4. Stripe-patchwork images of the conventional method and Idea 2 when $W = 4$ and $D = 3$

5. **Conclusions.** We developed a method to accelerate the processing of the conventional method [3]. The proposed method accelerated the processing of the conventional method with three ideas. The first idea used a lookup table that calculated the distance from the target pixel, the second idea sampled the pixels used in the calculation, and the third idea combined the first and second ideas. As a result of the experiments, all ideas could make the process faster than the conventional method, and the third idea worked fastest. And, since it was considered that a difference occurs between stripe-patchwork images of the conventional method and the second and third ideas, as a result of the quantitative appearance experiment, there was no large difference between stripe-patchwork images of the conventional method and the second and third ideas.

In future work, we will verify the effectiveness of the proposed method with various photographic images, and try to apply the proposed method to color photographic images and videos.

REFERENCES

[1] T. Hiraoka and H. Nonaka, Generating stripe-patchwork-like halftoning by k-means clustering and inverse filter, *ICIC Express Letters*, vol.11, no.5, pp.961-965, 2017.
[2] T. Hiraoka and K. Urahama, Generation of stripe-patchwork images by entropy and inverse filter, *ICIC Express Letters*, vol.11, no.12, pp.1787-1792, 2017.
[3] T. Hiraoka and H. Nonaka, A high-speed method for generating stripe-patchwork image, *ICIC Express Letters*, vol.12, no.9, pp.923-929, 2018.
[4] J. Lansdown and S. Schofield, Expressive rendering: A review of nonphotorealistic techniques, *IEEE Computer Graphics and Applications*, vol.15, no.3, pp.29-37, 1995.
[5] T. Isenberg, P. Neumann, S. Carpendale, M. C. Sousa and J. A. Jorge, Non-photorealistic rendering in context: An observational study, *Proc. of the 4th International Symposium on Non-photorealistic Animation and Rendering*, pp.115-126, 2006.
[6] A. Selim, M. Elgharib and L. Doyle, Painting style transfer for head portraits using convolutional neural networks, *ACM Trans. Graphics*, vol.35, no.4, pp.115-126, 129, 2016.
[7] P. L. Rosin, T. Wang, H. Winnemoller, D. Mould, I. Berger, J. Collomosse, Y. K. Lai, C. Li, H. Li, A. Shamir and M. L. Wand, Benchmarking non-photorealistic rendering of portraits, *Proc. of the Symposium on Non-Photorealistic Animation and Rendering*, pp.1-12, 2017.
[8] T. Wu, Saliency-aware generative art, *Proc. of the 2018 10th International Conference on Machine Learning and Computing*, pp.198-202, 2018.
[9] Z. Yu and K. Urahama, Iterative method for inverse nonlinear image processing, *IEICE Trans. Fundamentals*, vol.E97-A, no.2, pp.719-721, 2014.