ICIC Express Letters
Volume **15**, Number **2**, February **2021**

ICIC International ©2021 ISSN 1881-803X
pp. 149–155

GENERATION OF TWO-ORIENTED RIPPLE IMAGES BY TWO SMOOTHING FILTERS USING DIFFERENT REGIONS IN WINDOW

Toru Hiraoka

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

Received June 2020; accepted September 2020

Abstract. *We propose a non-photorealistic rendering method for automatically generating two-oriented ripple images from gray-scale photographic images. Ripple patterns imitate a wave on the water surface and are composed of continuous lines with fluctuations. Two-oriented ripple images are expressed by generating ripple patterns with two orientations. In our method, two smoothing filters using different regions in the window are applied to photographic images for transforming it to two-oriented ripple images. To verify the effectiveness of our method, we investigated the changes in two-oriented ripple images by changing the values of the parameters, and visually evaluated the appearance of these two-oriented ripple images.*

Keywords: Non-photorealistic rendering, Two orientation, Ripple pattern, Smoothing filter, Automatic generation

1. **Introduction.** Non-photorealistic rendering (NPR) is a technology of computer graphics [1] that can generate effective illustrations and artistic images. Such non-realistic images can be used on websites and social networking services (SNS). Many NPR methods have been proposed to simulate effective illustrations and artistic images [2, 3, 4, 5, 6, 7]. In the past NPR methods, an NPR method for generating ripple images from gray-scale photographic images has been proposed [8, 9]. Ripple patterns imitate a wave on the water surface and are composed of continuous lines with fluctuations. Ripple images are mainly composed of ripple patterns in one orientation. The conventional method [8] is conducted by an iteration process with intensity gradients. The conventional method [9] proposed the method that can arbitrarily change the orientation of ripple patterns by circular-sector-type smoothing filter, and expand the range of expression of the conventional method [8]. However, the conventional methods [8, 9] cannot generate ripple images composed of ripple patterns in two or more orientations. If ripple patterns with two orientations can be generated, the appearance of ripple images is further improved. Specifically, when horizontal ripple patterns are generated by the conventional methods [8, 9], the edges of photographic images are disturbed near the vertical edges. If two ripple patterns with the vertical and horizontal orientations can be generated, vertical ripple patterns can be generated near the vertical edges, and disturbance of the edges can be reduced.

This paper proposes an NPR method that can automatically generate two-oriented ripple images composed of ripple patterns in two orientations from gray-scale photographic images. The proposed method is executed by an iterative process with two smoothing filters using different regions in the window. The proposed method can arbitrarily change two orientations of ripple patterns according to the edges of photographic images. For example, vertical ripple patterns are generated near the vertical edges of photographic images, and horizontal ripple patterns are generated near the horizontal edges. To verify

DOI: 10.24507/icicel.15.02.149

the effectiveness of the proposed method, the changes in two-oriented ripple images by changing the values of the parameters were investigated. As a result of the experiments, it was clarified that the proposed method can automatically generate two-oriented ripple images.

The rest of this paper is organized as follows. Section 2 describes the proposed method for generating two-oriented ripple images. Section 3 shows experimental results, and reveals the effectiveness of the proposed method. Finally, Section 4 concludes this paper.

2. **Proposed Method.** The proposed method is executed in three steps. In the first step, smoothed images are created by two smoothing filters using different regions in the window. In the second step, differences between the pixel values of two smoothed images are calculated. In the third step, photographic images are converted using the differences. By repeating the first to third steps, two-oriented ripple images are generated. A flow chart of the proposed method is shown in Figure 1.

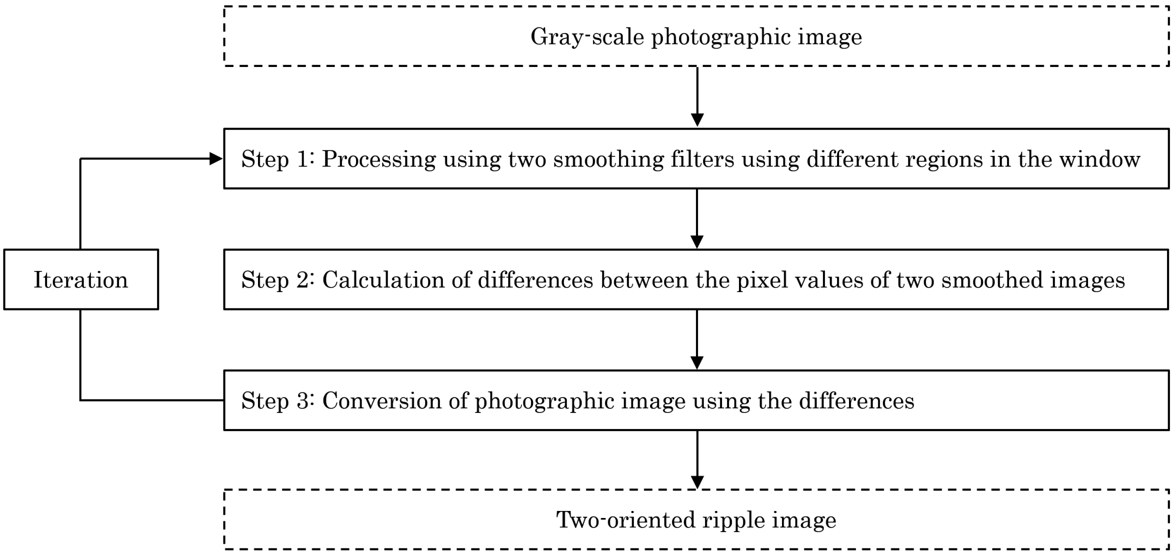


FIGURE 1. Flow chart of the proposed method

The detailed procedure in Figure 1 is shown as follows.

Step 0: The input pixel values for spatial coordinates (i, j) of a gray-scale photographic image are defined as $f_{i,j}$. Then, the pixel values of the image at the t-th iteration number are defined as $f_{i,j}^{(t)}$, where $f_{i,j}^{(1)} = f_{i,j}$. The pixel values $f_{i,j}^{(t)}$ have value of U gradation from 0 to $U - 1$.

Step 1: The image with the pixel values $f_{i,j}^{(t)}$ is smoothed using two different regions in the window of W pixels. Smoothing is calculated as the average of the pixel values included in each region. The two regions shown in Figure 2 are windows separated by a straight line with a slope of θ_1 radian passing through the center of the window. The pixels on the straight line are included in the two regions. The pixel values of two smoothed images are defined as $s_{\theta_1,1,i,j}^{(t)}$ and $s_{\theta_1,2,i,j}^{(t)}$, respectively. Similarly, two smoothed images composed of the pixel values $s_{\theta_2,1,i,j}^{(t)}$ and $s_{\theta_2,2,i,j}^{(t)}$ are obtained by setting a straight line to θ_2 radian.

Step 2: The differences $d_{\theta_1,i,j}^{(t)}$ between the pixel values $s_{\theta_1,1,i,j}^{(t)}$ and $s_{\theta_1,2,i,j}^{(t)}$ of the smoothed images are calculated as the following equation.

$$d_{\theta_1,i,j}^{(t)} = s_{\theta_1,2,i,j}^{(t)} - s_{\theta_1,1,i,j}^{(t)} \tag{1}$$

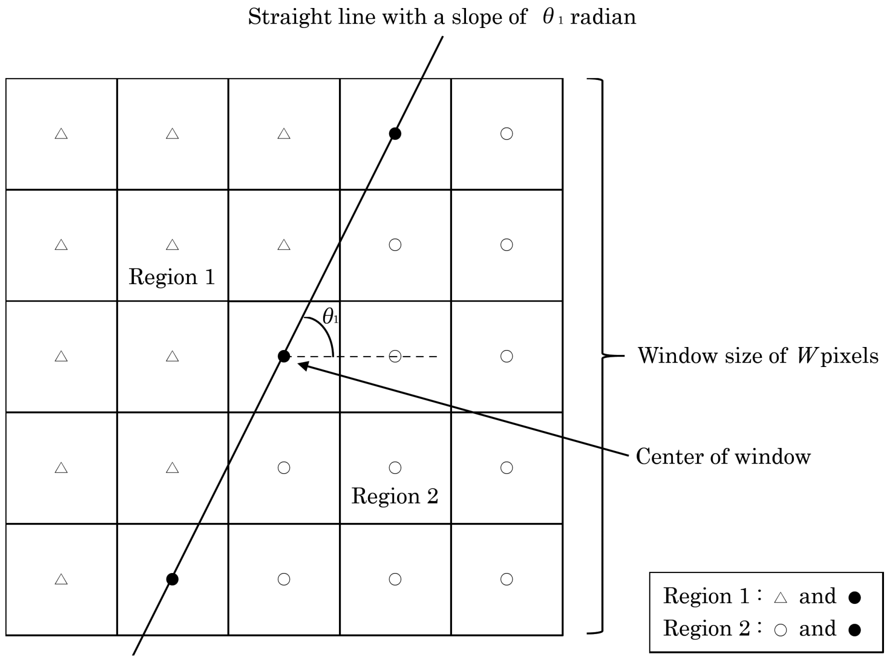


FIGURE 2. Conceptual diagram of two regions in the window

Similarly, the differences $d_{\theta_2,i,j}^{(t)}$ are obtained. In case $\left|d_{\theta_1,i,j}^{(t)}\right|$ is greater than or equal to $\left|d_{\theta_2,i,j}^{(t)}\right|$, $d_{i,j}^{(t)}$ is set to $d_{\theta_1,i,j}^{(t)}$. In case $\left|d_{\theta_1,i,j}^{(t)}\right|$ is less than $\left|d_{\theta_2,i,j}^{(t)}\right|$, $d_{i,j}^{(t)}$ is set to $d_{\theta_2,i,j}^{(t)}$.

Step 3: The differences $d_{i,j}^{(t)}$ are added to the pixel values $f_{i,j}$ of the photographic image, and the pixel values $f_{i,j}^{(t)}$ are updated to $f_{i,j}^{(t+1)}$ as following equation.

$$f_{i,j}^{(t+1)} = f_{i,j} + ad_{i,j}^{(t)} \tag{2}$$

$$f_{i,j}^{(t+1)} = \begin{cases} f_{i,j} + ad_{i,j}^{(t)} & (t \,\%\, 2 = 0) \\ f_{i,j} - ad_{i,j}^{(t)} & (t \,\%\, 2 = 1) \end{cases} \tag{3}$$

where a is a positive constant, and the notation % represents a remainder operation. In case $f_{i,j}^{(t+1)}$ is less than 0, then $f_{i,j}^{(t+1)}$ must be set to 0. In case $f_{i,j}^{(t+1)}$ is greater than $U-1$, then $f_{i,j}^{(t+1)}$ must be set to $U-1$.

A two-oriented ripple image is obtained after the above processing (Steps 1 to 3) of T times iteration.

3. **Experiments.** The experiments with changing the values of the parameters θ_1 and θ_2 in the proposed method were conducted using six gray-scale photographic images shown in Figure 3. Since many photos of people and scenery were uploaded on the website

(a) Boat (b) Building (c) Cameraman (d) Lenna (e) Lighthouse (f) Woman

FIGURE 3. Photographic images

and SNS, six gray-scale photographic images were picked up of people and scenery. All photographic images used in the experiments were $512 * 512$ pixels and 256 gradation. The values of the parameters θ_1 and θ_2 were respectively set to 0 and $\pi/2$, and were respectively set to $\pi/4$ and $3\pi/4$. Since ripple patterns were generated clear, the values of the parameters W, a and T were respectively set to 5, 1.2 and 30 in the experiments.

For reference, two-oriented ripple images generated from Lenna image (Figure 3(d)) according to the value of the window size W are shown in Figure 4. Figure 4 shows the case that the value of the window size W is 3, 5, 7 and 9. From Figure 4, it could be seen that the spacing between ripple patterns became wider as the value of the window size W increases. And, two-oriented ripple images generated from Lenna image according to the value of the parameter a are shown in Figure 5. Figure 5 shows the case that the value of the parameter a is 1.1, 1.2, 1.3 and 1.4. From Figure 5, it could be seen that the contrast of ripple patterns became higher as the value of the parameter a increases. And, two-oriented ripple images generated from Lenna image according to the value of

(a) $W = 3$ (b) $W = 5$ (c) $W = 7$ (d) $W = 9$

FIGURE 4. Two-oriented ripple images when the window size W is changed

(a) $a = 1.1$ (b) $a = 1.2$ (c) $a = 1.3$ (d) $a = 1.4$

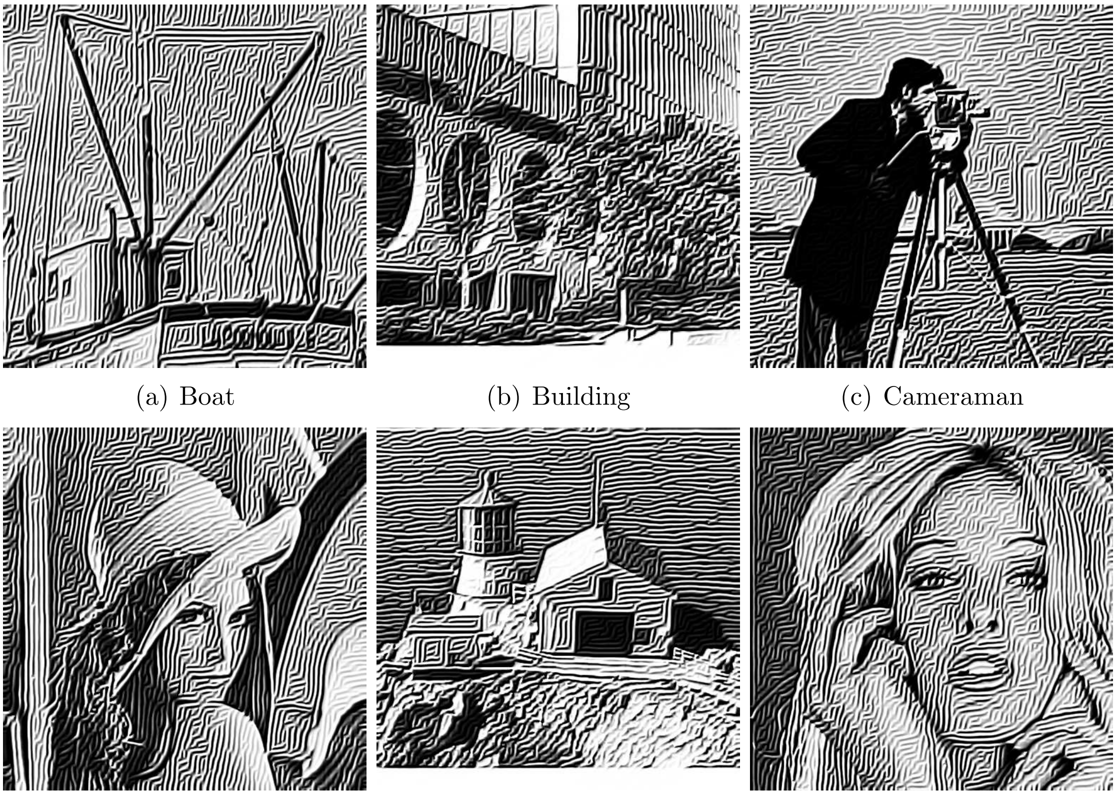
FIGURE 5. Two-oriented ripple images when the parameter a is changed

(a) $T = 5$ (b) $T = 10$ (c) $T = 20$ (d) $T = 30$

FIGURE 6. Two-oriented ripple images when the iteration number T is changed

the iteration number T are shown in Figure 6. Figure 6 shows the case that the value of the iteration number T is 5, 10, 20 and 30. From Figure 6, it could be seen that ripple patterns became clearer as the value of the iteration number T increases.

The results of the experiments are shown in Figure 7 and Figure 8. Figure 7 shows two-oriented ripple images in case the values of the parameters θ_1 and θ_2 were set to 0 and $\pi/2$, respectively. Figure 8 shows two-oriented ripple images in case the values of the parameters θ_1 and θ_2 were set to $\pi/4$ and $3\pi/4$, respectively. Two-oriented ripple images in Figure 7 were mainly composed of ripple patterns of 0 and $\pi/2$ radians. From Figure 7, vertical ripple patterns were generated near the vertical edges of photographic images, and horizontal ripple patterns were generated near the horizontal edges. Two-oriented ripple images in Figure 8 were mainly composed of ripple patterns of $\pi/4$ and $3\pi/4$ radians. Compared to ripple patterns in Figure 7, ripple patterns in Figure 8 could generate more along the diagonally right and left edges. Hence, compared to the conventional methods [8, 9], the proposed method could generate two-oriented ripple patterns and could suppress the disorder of ripple patterns around the edges. In all cases, ripple patterns could be automatically generated on the whole image. However, like the black cloth of (c) Cameraman and the white roof of (e) Lighthouse, it was difficult to generate ripple patterns in the black and white areas.



(a) Boat (b) Building (c) Cameraman

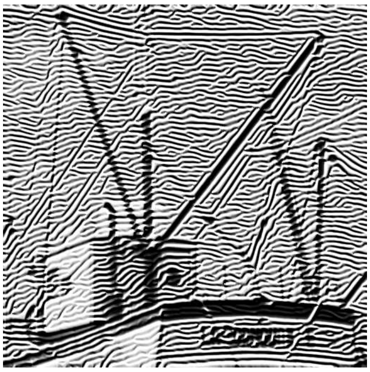
(d) Lenna (e) Lighthouse (f) Woman

FIGURE 7. Two-oriented ripple images when $\theta_1 = 0$ and $\theta_2 = \pi/2$

To generate ripple patterns in the black and white areas, the proposed method was applied by converting the pixel values $f_{i,j}$ of photographic images to the pixel values $f'_{i,j}$ as following equation.

$$f'_{i,j} = b + f_{i,j}\frac{U - 2b - 1}{U - 1} \quad (4)$$

where b is a positive constant. Two-oriented ripple images generated from Cameraman and Lighthouse images in case of $b = 20$ are shown in Figure 9. From Figure 9, ripple

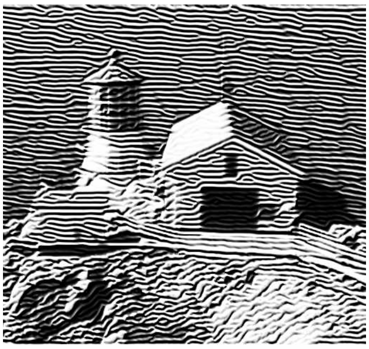


(a) Boat

(b) Building

(c) Cameraman

(d) Lenna

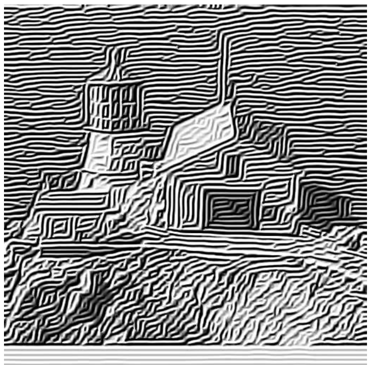


(e) Lighthouse

(f) Woman

FIGURE 8. Two-oriented ripple images when $\theta_1 = \pi/4$ and $\theta_2 = 3\pi/4$

(a) Cameraman



(b) Lighthouse

FIGURE 9. Two-oriented ripple images when $b = 20$

patterns could be generated on the black cloth of Cameraman image and the white roof of Lighthouse image. However, the reproducibility of photographic images near the edges where pixel values of photographic images changed little, such as the roof on the right of Lighthouse image, was reduced. Hence, it is necessary to set the value of the parameter b appropriately according to photographic images.

4. **Conclusions.** This paper proposed an NPR method for automatically generating two-oriented ripple images from gray-scale photographic images. The proposed method was executed by an iterative process with two smoothing filters using different regions in the window. Through experiments that change the values of parameters in the proposed method, it was revealed the changes in appearance of two-oriented ripple images. However, it was difficult to generate ripple patterns in the black and white areas.

A subject for future study is to be able to generate ripple patterns in the black and white areas better. Another task is to expand the proposed method for application to color photographic images and videos.

Acknowledgment. This work was supported by JSPS KAKENHI Grant Number JP19-K12664.

REFERENCES

[1] L. Xue, X. Yi, Y.-C. Lin and J. W. Drukker, An approach of the product form design based on GRA-fuzzy logic model: A case study of train seats, *International Journal of Innovative Computing, Information and Control*, vol.15, no.1, pp.261-274, 2019.

[2] P. Haeberli, Paint by numbers: Abstract image representations, *ACM SIGGRAPH Computer Graphics*, vol.24, no.4, pp.207-214, 1990.

[3] D. D. Seligmann and S. Feiner, Automated generation of intent-based 3D illustrations, *ACM SIGGRAPH Computer Graphics*, vol.25, no.4, pp.123-132, 1991.

[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] W. Qian, D. Xu, K. Yue, Z. Guan, Y. Pu and Y. Shi, Gourd pyrography art simulating based on non-photorealistic rendering, *Multimedia Tools and Applications*, vol.76, no.13, pp.14559-14579, 2017.

[6] D. Martin, G. Arroyo, A. Rodriguez and T. Isenberg, A survey of digital stippling, *Computers & Graphics*, vol.67, pp.24-44, 2017.

[7] W. Qian, D. Xu, J. Cao, Z. Guan and Y. Pu, Aesthetic art simulation for embroidery style, *Multimedia Tools and Applications*, vol.78, no.1, pp.995-1016, 2019.

[8] T. Hiraoka, Generation of ripple images using intensity gradient, *ICIC Express Letters*, vol.13, no.12, pp.1097-1101, 2019.

[9] T. Hiraoka, Generation of arbitrarily-oriented ripple images using circular-sector-type smoothing filter and inverse filter, *Journal of Robotics, Networking and Artificial Life*, vol.6, no.4, pp.213-216, 2020.