Generation of Flowing-Line Images Using Vertical and Horizontal Smoothing Filters

Karin Kuroki
Department of Information Systems, University of Nagasaki, Japan

Toru Hiraoka
Department of Information Systems, University of Nagasaki, Japan
E-mail: hiraoka@sun.ac.jp

Abstract

We propose a non-photorealistic rendering method for automatically generating flowing-line images from photographic images. Flowing-line images consist of unidirectional lines with smooth curves. Our method is executed by an iterative calculation using vertical and horizontal smoothing filters. To verify the effectiveness of our method, we conducted an experiment using various photographic images to confirm that flowing-line patterns can be generated on the entire image. Additionally, we conducted an experiment to visually examine how flowing-line patterns generated by changing the values of the parameters in our method change.

Keywords: Non-photorealistic rendering, Flowing line, Vertical smoothing filter, Horizontal smoothing filter, Automatic generation

1. Introduction

Many studies [1],[2] have been conducted on non-photorealistic rendering using computer technology. Since non-photorealistic rendering targets all expressions other than realistic expressions by photorealistic rendering, there are various methods for dealing with various styles and uses. We focus on non-photorealistic rendering methods that use image processing to generate non-photorealistic images that are different from conventional art expressions such as oil paintings and watercolors. As non-photorealistic rendering methods of the unprecedented art expressions, many methods for generating labyrinth images [3], cell-like images [4], and moire-like images [5] from photographic images have been proposed. Labyrinth images are composed of equally spaced lines like a maze and are generated using minimum spanning trees, cell-like images are composed of cell-like patterns with cell membranes and cell nuclei and are generated using inverse iris filter, and moire-like images are a kind of op art and are generated using bilateral filter and unsharp musk.

We propose a non-photorealistic rendering method for automatically generating flowing-line images from photographic images. Flowing-line images are composed of unidirectional (here vertical) lines with smooth curves. The black lines are thickly represented at the edges and black areas of photographic images, and are represented along the edges in the nearly vertical direction. Our method is executed by an iterative calculation using vertical and horizontal smoothing filters. As images similar to flowing-line images, non-photorealistic rendering methods for generating ripple images has been proposed [6],[7]. Ripple images are composed of continuous lines with fluctuations and are generated using inverse Sobel filter [6] or intensity gradients [7]. Flowing-line images are represented by the lines closer to the binary value of black and white than ripple images, and flowing-line images and ripple images have different impressions. The conventional method [8] is executed by an iterative calculation using circular-sector-type smoothing filter and inverse filter [9]. Ripple images [8] are more similar to flowing-line images than ripple images [6],[7], but flowing-line patterns are slightly more linear than ripple patterns. In the conventional methods [6],[7],[8], ripple patterns cannot be generated in white and black areas. On the other hand, our method can generate flowing-line patterns in the white and black areas. Through an experiment using various photographic images, our method examines that flowing-line patterns can be generated on the entire image. Additionally, through an experiment that changes the values of the parameters in our method, how the generated flowing-line patterns change is examined.

2. Our method

Our method is largely executed in three steps. First, since flowing-line patterns are unlikely to occur in the white and black areas, gray scale transformation is performed on a photographic image. The gray scale transformation eliminates pixel values close to white and black, and eliminates areas where flowing-line patterns cannot be generated. Second, processing is performed using horizontal smoothing filter and inverse filter. Third, processing is performed using vertical smoothing filter. A flowing-line image is generated by iterating through the second and third steps. A flow chart of our method is shown in Fig. 1.

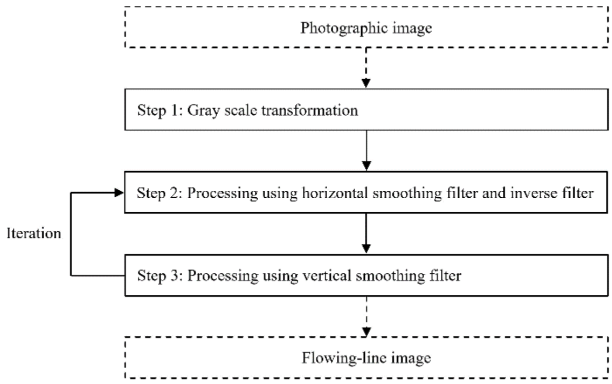


Fig. 1. Flow chart of our method

Details of the steps in Fig. 1. are explained below.

Step 0: Let the input pixel values on coordinates (i, j) of a gray-scale photographic image be $o_{i,j}$. The pixel values $o_{i,j}$ have value of U gradation from 0 to $U-1$.

Step 1: Let the pixel values that have undergone gray scale transformation as shown in the following equation be $f_{i,j}$.

$$f_{i,j} = o_{i,j}\,\frac{U-2D-1}{U-1} + D \qquad (1)$$

where D is a natural number. The larger the value of D, the fewer pixels that are close to white or black. When the value of D is 0, the pixel values $f_{i,j}$ have the same pixel values as the original photographic image.

Step 2: Let the pixel values of the image at the t-th iteration number be $f_{i,j}^{(t)}$, where $f_{i,j}^{(0)} = f_{i,j}$. Let the pixel values that horizontal smoothing filter is applied as shown in the following equation be $s_{1,i,j}^{(t)}$.

$$s_{1,i,j}^{(t)} = \frac{\sum_{k=-W_1}^{W_1} f_{i+k,j}^{(t-1)}}{2W_1+1} \qquad (2)$$

where W_1 is the window size and k is the positions in the window. Let the pixel values that inverse filter is applied as shown in the following equation be $g_{i,j}^{(t)}$.

$$g_{i,j}^{(t)} = f_{i,j}^{(t-1)} - s_{1,i,j}^{(t)} + f_{i,j} \qquad (3)$$

If $g_{i,j}^{(t)}$ is smaller than 0, then $g_{i,j}^{(t)}$ must be set to 0, and if $g_{i,j}^{(t)}$ is greater than $U-1$, then $g_{i,j}^{(t)}$ must be set to $U-1$.

Step 3: Let the pixel values that vertical smoothing filter is applied as shown in the following equation be $f_{i,j}^{(t)}$.

$$f_{i,j}^{(t)} = \frac{\sum_{l=-W_2}^{W_2} g_{i,j+l}^{(t)}}{2W_2+1} \qquad (4)$$

where W_2 is the window size and l is the positions in the window.

A flowing-line image is obtained after T times iteration of Steps 2 and 3.

3. Experiments

We conducted two experiments. First, we visually confirmed flowing-line patterns by changing the values of the parameters in our method using Lighthouse image shown in Fig. 2. Second, we applied our method to eight photographic images shown in Fig. 3. All photographic images used in the experiments were 512 * 512 pixels and 256 gradations.

Fig. 2. Lighthouse image

Fig. 3. Photographic images

3.1. *Experiment with changing parameter values*

Flowing-line images generated by changing the iteration number T were visually confirmed using Lighthouse image. The value of T was set to 5, 10, 20, and 40. The values of the other parameters D, W_1, and W_2 were set to 10, 6, and 3, respectively. The results of the experiment are shown in Fig. 4. As the value of T was larger, flowing-line patterns became clearer and were expressed finely.

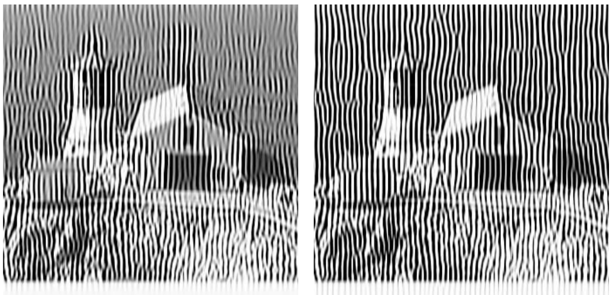
Flowing-line images generated by changing the value of the parameter D were visually confirmed using Lighthouse image. The value of D was set to 0, 10, 20, and 30. The values of the other parameters T, W_1, and W_2 were set to 40, 6, and 3, respectively. The results of the experiment are shown in Fig. 5. As the value of D was larger, flowing-line patterns became clearer in the white area at the bottom of Lighthouse image. On the other hand, as the value of D was larger, it became more difficult to recall Lighthouse image.

Flowing-line images generated by changing the window size W_1 were visually confirmed using

Lighthouse image. The window size W_1 was set to 2, 4, 6, and 8. The values of the other parameters T, D, and W_2 were set to 40, 10, and 3, respectively. The results of the experiment are shown in Fig. 6. As the value of W_1 was larger, the interval of flowing-line patterns became wider.

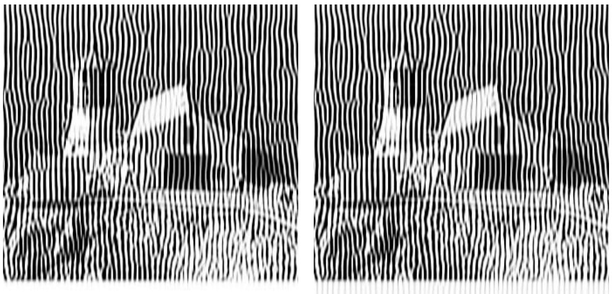
Flowing-line images generated by changing the window size W_2 were visually confirmed using Lighthouse image. The window size W_2 was set to 1, 2, 3, and 4. The values of the other parameters T, D, and W_1 were set to 40, 10, and 6, respectively. The results of the experiment are shown in Fig. 7. As the value of W_2 was larger, flowing-line patterns became smoother. Furthermore, in Figs. 4 to 7, it was also found that flowing-line patterns can be generate in the white area at the bottom of Lighthouse image.

(a) T = 5 (b) T = 10

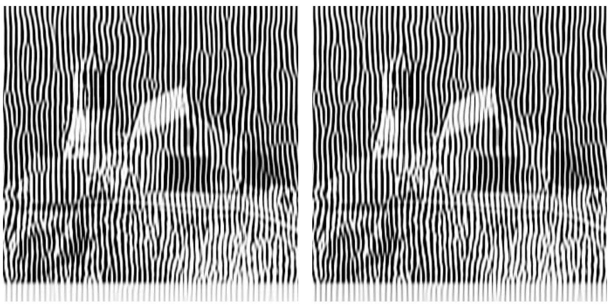


(c) T = 20 (d) T = 40

Fig. 4. Flowing-line images in the case of the iteration number T = 5, 10, 20, and 40

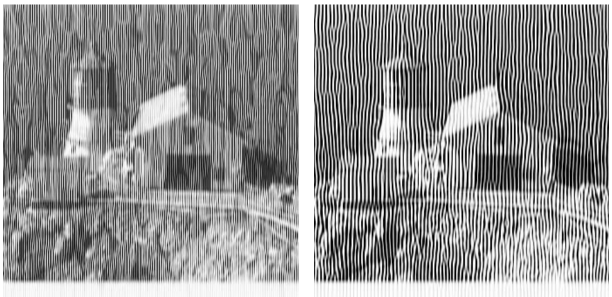


(a) D = 0 (b) D = 10

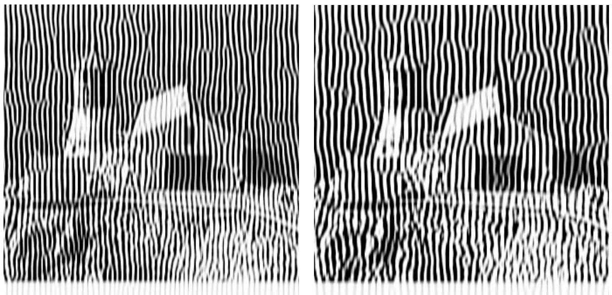


(c) D = 20 (d) D = 30

Fig. 5. Flowing-line images in the case of the parameter value D = 0, 10, 20, and 30

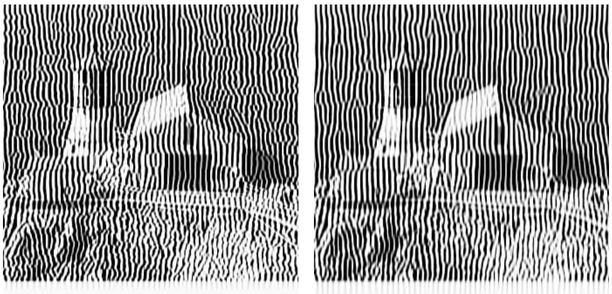


(a) W_1= 2 (b) W_1 = 4

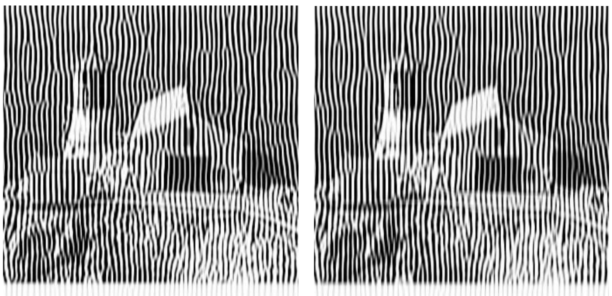


(c) W_1= 6 (d) W_1 = 8

Fig. 6. Flowing-line images in the case of the window size W_1 = 2, 4, 6, and 8



(a) W_2= 1 (b) W_2 = 2



(c) W_2= 3 (d) W_2 = 4

Fig. 7. Flowing-line images in the case of the window size W_2 = 1, 2, 3, and 4

3.2. *Experiment using eight photographic images*

Our method was applied to eight photographic images shown in Fig. 3. The values of the parameters T, D, W_1, and W_2 were set to 40, 10, 6, and 3, respectively. The results of the experiment are shown in Fig. 8. For all flowing-line images, flowing-line patterns were composed of unidirectional (here vertical) lines with smooth curves, and could be automatically generated on the entire image. Looking at more details, the black lines were thickly represented at the edges and black areas (clothes, hair, etc. in the lower left image) of photographic images, and were represented along the edges in the nearly vertical direction.

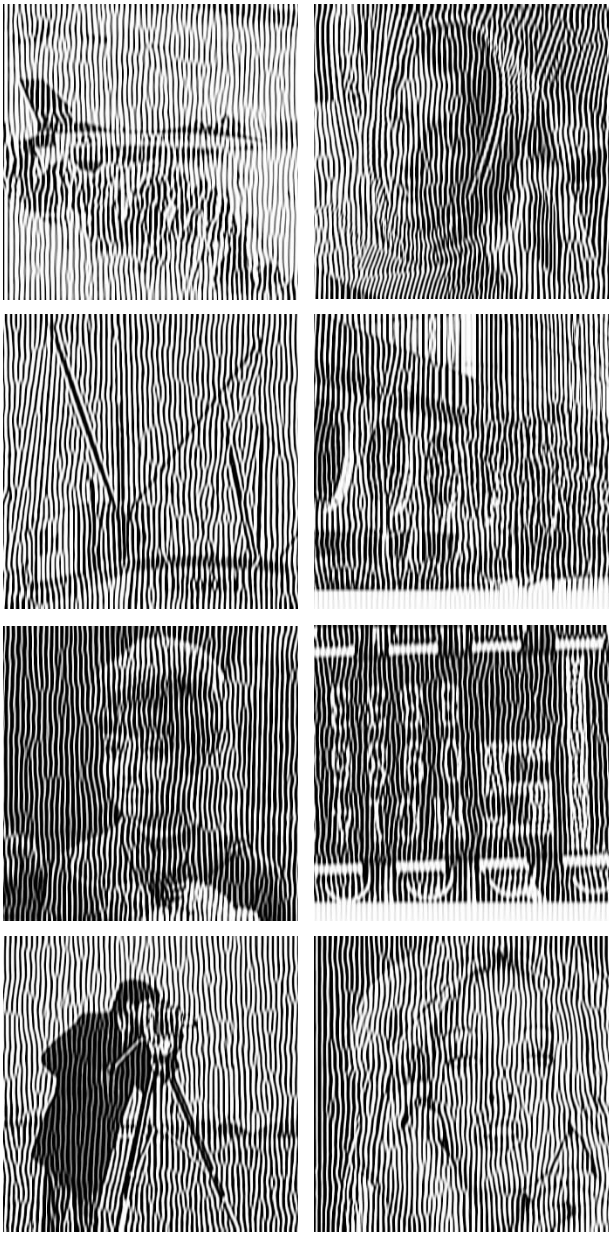


Fig. 8. Flowing-line images

4. Conclusion

We proposed a non-photorealistic rendering method for generating flowing-line images from gray-scale photographic images using vertical and horizontal smoothing filters. Through an experiment using nine photographic images, it was found that our method can automatically generate flowing-line patterns on the entire images. Additionally, through an experiment that the values of the parameters in our method were changed, it was found that the interval and smoothness of flowing-line patterns can be changed.

A subject for future study is to expand our method for application to color photographic images, videos, and three-dimensional data.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP23K11727 and The Telecommunications Advancement Foundation Grant.

References

1. P. Haeberli. Paint by numbers, abstract image representations, ACM SIGGRAPH Computer Graphics 24(4), 1990, pp. 207-214.
2. J. Lansdown and S. Schofield, Expressive rendering, a review of nonphotorealistic techniques, IEEE Computer Graphics and Applications 15(3), 1995 pp. 29-37.
3. K. Inoue and K. Uraham, Halftoning with minimum spanning trees and its application to maze-like images, LSEVIER Computers & Graphics 33(5), 2009, pp. 638-647.
4. T. Hiraoka, M. Hirota, K. Inoue and K. Urahama, Generating cell-like color images by inverse iris filter, ICIC Express Letters 11(2), 2017, pp.399-404.
5. T. Hiraoka and K. Urahama, Generation of moire-picture-like color images by bilateral filter, IEICE Transactions on Information and Systems E96-D(8), 2013, pp. 1862-1866.
6. T. Hiraoka and K. Urahama, Generating ripple-pattern-like images by inverse Sobel filter, Journal of the Institute of Image Electronics Engineers of Japan 44(2), 2015, pp. 349-352.
7. T. Hiraoka, Generation of ripple images using intensity gradient, ICIC Express Letters 13(12), 2019, pp. 1097-1101.
8. T. Hiraoka, Generation of arbitrarily-oriented ripple images using circular-sector-type smoothing filter and inverse filter, Journal of Robotics, Networking and Artificial Life 6(4), 2020, pp. 213-216.
9. Z. Yu and K. Urahama, Iterative method for inverse nonlinear image processing, IEICE Transactions on Fundamentals E97-A(2), 2014, pp. 719-721.

Authors Introduction

Ms. Karin Kuroki

She is a student in the Department of Information Systems, University of Nagasaki. His research interests inculede image processing and non-photorealistic rendering.

Dr. Toru Hiraoka

He received B.Des., M.Des. and D.Eng. degrees from Kyushu Institute of Design in 1995, 1997 and 2005, respectively. He is currently a Professor in University of Nagasaki. His research interests include non-photorealistic rendering and disaster prevention.