ICIC Express Letters
Volume **12**, Number **5**, May **2018**
ICIC International ©2018 ISSN 1881-803X
pp. 409–415

GENERATION OF INTERFERENCE-RIPPLE IMAGES BY INVERSE SOBEL FILTER

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 October 2017; accepted January 2018

Abstract. *We proposed a non-photorealistic rendering method for generating interference-ripple images from photographs. The proposed method generates interference-ripple images by an iterative process using inverse Sobel filter. Interference-ripple images generated by the proposed method have the following two characteristics. The first feature is that a smoothly curving interference-ripple pattern that follows the outline of the source photograph can be generated automatically. The second feature is that colorful interference-ripple patterns that increase the saturation from the source photograph can be automatically generated. To validate the effectiveness of the proposed method, we apply the proposed method to several photographs and confirm the visual effects of interference-ripple images.*
Keywords: Non-photorealistic rendering, Interference, Ripple, Inverse filter, Sobel filter

1. **Introduction.** Examples of previous non-photorealistic rendering (NPR) research include creating maze images [1, 2, 3], pointillistic images [4, 5, 6], and stripe images [7, 8, 9] from photographs. There is demand for NPR images for their entertainment value, and such features are often provided in smartphone or PC applications, for example to produce non-photorealistic images to be uploaded to social networking sites like Facebook or Twitter.

This article presents a method for producing NPR images with an interference wave pattern from a photograph. An interference-ripple pattern image is similar to a ripple pattern like the one formed in water or desert sand (Figure 1) under interference by an oil film on water or glass (Figure 2). The interference-ripple image used in this paper has two features.

- It is possible to automatically generate a smoothly curving interference-ripple pattern that follows the outline of the source photograph.
- It is possible to automatically generate colorful interference-ripple patterns that increase the saturation from the source photograph.

As an example of use, interference-ripple images make it difficult to visually recognize people and backgrounds in a photograph, and this property can be used for security when uploading a photography to websites or social networking services. The proposed method generates the interference-ripple image by an iterative process [10, 11] that uses an inverse Sobel filter. To verify the effectiveness of the proposed method, we investigate the visual effect in interference-ripple images generated by applying the proposed method to various photographs.

Literature [12] presents a conventional method for generating ripple image using the inverse Sobel filter. If the conventional method is applied to an unprocessed RGB image, then the interference-ripple pattern (Figure 3) will have fine resolution and be difficult to recognize as a smooth curve like the interference-ripple pattern in the proposed method.

FIGURE 1. Example of a ripple effect

FIGURE 2. Example of interference

FIGURE 3. Interference-ripple image of the conventional method [12]

FIGURE 4. Interference image of the conventional method [13]

Literature [13] presents a conventional method for generating interference image (Figure 4) using a bilateral lower envelope filter. In the interference image of the conventional method, the interference pattern is fine and the pattern forms a smooth curve, like an ellipse or a quadratic curve, that can make it difficult to visually recognize the smooth curve, such as the interference-ripple pattern of the proposed method.

The rest of this paper is organized as follows. Section 2 describes the proposed method using the inverse Sobel filter. Section 3 shows experimental results, and reveals that the proposed method can generate automatically the smooth interference-ripple pattern that follows outlines in the source photograph, and can generate automatically colorful interference-ripple patterns that increase the saturation from the source photograph. Finally, Section 4 concludes this paper.

2. **Proposed Method.** The proposed method generates the interference-ripple image by the iterative process that uses the inverse Sobel filter. A Sobel filter extracts the counter line from a photograph. An inverse filter restores an image converted by a processing to an original image. The inverse Sobel filter is calculated by using a procedure that restores the image converted by the Sobel filter. A flow chart of the proposed method is shown in Figure 5.

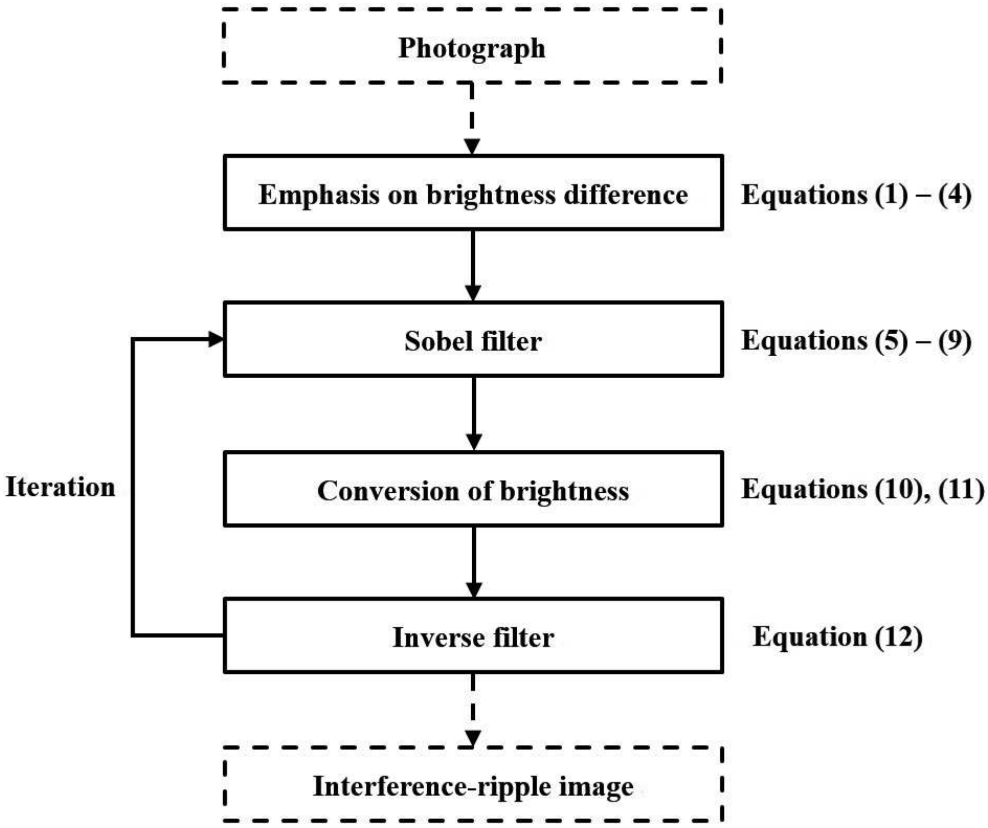


FIGURE 5. Flow chart of the proposed method

Let I and J be pixel counts in the x (horizontal) and y (vertical) directions, respectively, of a color photograph, and let $o_{R,i,j}$, $o_{G,i,j}$, and $o_{B,i,j}$ be RGB pixel values (within $0, \ldots, 255$) at spatial coordinates (i, j) ($i = 0, 1, \ldots, I-1$; $j = 0, 1, \ldots, J-1$). We use pixel values $o_{R,i,j}$, $o_{G,i,j}$, and $o_{B,i,j}$ to calculate the pixel average $o_{AVE,i,j}$ by Equation (1).

$$o_{AVE,i,j} = \frac{o_{R,i,j} + o_{G,i,j} + o_{B,i,j}}{3} \quad (1)$$

We use the pixel average $o_{AVE,i,j}$ to calculate the overall image average o_{AVE} by Equation (2).

$$o_{AVE} = \frac{\sum_{i=0}^{I-1}\sum_{j=0}^{J-1} o_{AVE,i,j}}{IJ} \tag{2}$$

We use Equation (3) to convert pixel value $o_{R,i,j}$ to pixel value $f_{R,i,j}$ if $o_{AVE,i,j}$ is less than or equal to o_{AVE}, and use Equation (4) otherwise.

$$f_{R,i,j} = 127\frac{o_{R,i,j}}{o_{AVE}} \tag{3}$$

$$f_{R,i,j} = 127 + 128\frac{o_{R,i,j} - o_{AVE,i,j}}{255 - o_{AVE}} \tag{4}$$

If $f_{R,i,j}$ exceeds 255 in Equation (3) it is assigned 255, and if $f_{R,i,j}$ is less than 0 in Equation (4) it is assigned 0. We similarly convert pixel values $o_{G,i,j}$ and $o_{B,i,j}$ into values $f_{G,i,j}$ and $f_{B,i,j}$, respectively. The hue of each pixel after conversion is largely preserved, and the average of the entire image after conversion is about 127 (i.e., centered). Further, as the difference between light and dark increases in the converted image, it is possible to raise the saturation, making it easier to produce an interference-ripple pattern that follows the outline of the photography by processing using the Sobel filter as described below.

We perform the following processing for pixel values $f_{R,i,j}$, $f_{G,i,j}$, and $f_{B,i,j}$. Note that the following describes processing for only pixel values $f_{R,i,j}$, but processing for $f_{G,i,j}$ and $f_{B,i,j}$ is analogous. Let the pixel value after Sobel filter processing be $SF(f_{R,i,j})$. Equations (5) and (6) show the operator for the Sobel filter with window size 3×3.

$$s_x = [-1, 0, 1; -2, 0, 2; -1, 0, 1] \tag{5}$$

$$s_y = [-1, -2, -1; 0, 0, 0; 1, 2, 1] \tag{6}$$

Values in the operators in Equations (5) and (6) are for, the upper left, upper, upper right, left, central, right, lower left, lower, and lower right pixels of the window, in that order. In the calculation of $SF(f_{R,i,j})$, the operators in Equations (7) and (8) are first used to find $SF_x(f_{R,i,j})$ and $SF_y(f_{R,i,j})$, respectively, and then $SF_s(f_{R,i,j})$ is taken as the square root of the sum of squares of $SF_x(f_{R,i,j})$ and $SF_y(f_{R,i,j})$ (Equation (9)).

$$SF_x(f_{R,i,j}) = -f_{R,i-1,j-1} + f_{R,i+1,j-1} - 2f_{R,i-1,j} + 2f_{R,i+1,j} - f_{R,i-1,j+1} + f_{R,i+1,j+1} \tag{7}$$

$$SF_y(f_{R,i,j}) = -f_{R,i-1,j-1} - 2f_{R,i,j-1} - f_{R,i+1,j-1} + f_{R,i-1,j+1} + 2f_{R,i,j+1} + f_{R,i+1,j+1} \tag{8}$$

$$SF_s(f_{R,i,j}) = \sqrt{SF_x(f_{R,i,j})^2 + SF_y(f_{R,i,j})^2} \tag{9}$$

Let $SF_{s,\min}$ and $SF_{s,\max}$ be the minimum and maximum values, respectively, of $SF_s(f_{R,i,j})$. In Equations (10) and (11), $SF_s(f_{R,i,j})$ is converted to $SF(f_{R,i,j})$, taking a value from -256 to 255. This conversion uses Equation (10) for all $SF_s(f_{R,i,j})$, $SF_s(f_{G,i,j})$, and $SF_s(f_{B,i,j})$ values of 127 or less, and Equation (11) otherwise.

$$SF(f_{R,i,j}) = 255 - 511\frac{SF_s(f_{R,i,j}) - SF_{s,\min}}{SF_{s,\max} - SF_{s,\min}} - a(127 - f_{R,i,j}) \tag{10}$$

$$SF(f_{R,i,j}) = 255 - 511\frac{SF_s(f_{R,i,j}) - SF_{s,\min}}{SF_{s,\max} - SF_{s,\min}} \tag{11}$$

If $SF(f_{R,i,j})$ is less than -256 in Equation (10), it is assigned -256; and if it is greater than 255 it is assigned 255. In Equation (10), a is a positive constant, set as 1.0 in the following experiment. Equation (11) is a process performed on areas not considered to be contour areas, and approaches -256 with proximity to contour areas. Equation (10) is a process performed on areas considered to be contour areas, and is applied to the $-a(127 - f_{R,i,j})$ term in Equation (11). This term brings values in dark areas of the photograph closer to −256 as a contour area because the interference-ripple pattern is less likely to occur in dark areas than in bright areas.

The inverse Sobel filter is calculated as

$$f_{R,i,j}^{(t+1)} = \frac{f_{R,i,j}^{(t)} - SF\left(f_{R,i,j}^{(t)}\right)}{2} + f_{R,i,j} \tag{12}$$

where t is the number of iterations. Taking $f_{R,i,j}$ to have initial value $f_{R,i,j}^{(0)}$, we find $f_{R,i,j}^{(1)}, f_{R,i,j}^{(2)}, \ldots$ from Equation (12). In Equation (12), if $f_{R,i,j}^{(t)}$ becomes less than 0 (resp., more than 255), it is assigned 0 (resp., 255).

The image produced from RGB pixel values $f_{R,i,j}^{(T)}$, $f_{G,i,j}^{(T)}$, and $f_{B,i,j}^{(T)}$ obtained after T repetitions of the inverse Sobel filter is the interference-ripple image. In the following experiment, T was taken as 100 because all interference-ripple images had converged by the 100th iteration.

3. **Experiment.** We applied the proposed method to the eight 512×512, 256-level photographs shown in Figure 6. Figure 7 shows the interference-ripple images generated by the proposed method.

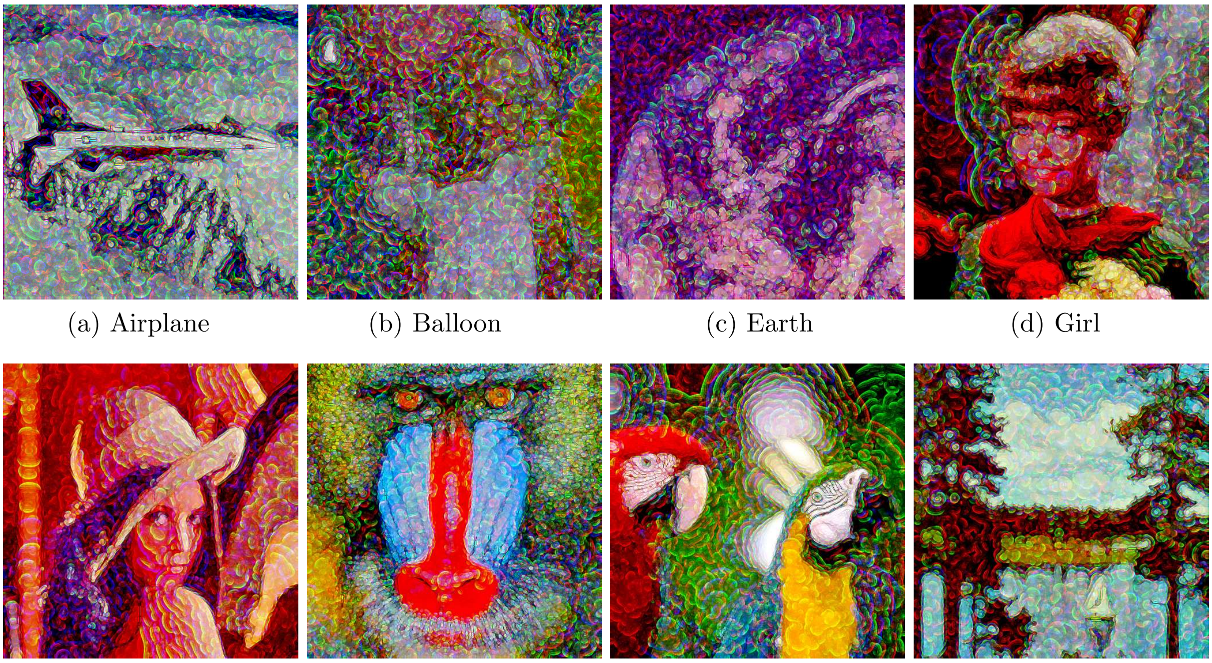
First, Figure 7 shows that the interference-ripple pattern forms a smooth curve that follows the outline in the photograph. For example, interference-ripple patterns are generated along the outline of the clouds in Figure 7(c) and the white area at the top of 7(g). However, there are also areas where interference-ripple patterns are not generated, such as at the bottom left in Figure 7(d) and the red nose in 7(f). In those areas, it is difficult for the interference-ripple pattern to appear due to little variation in pixel values.

Further, Figure 7 shows that all interference-ripple images are more saturated than the source photographs, so colorful interference-ripple patterns can be generated. For example, considering white areas other than the aircraft in Figure 7(a) and the entirety of 7(b), we can see that the interference-ripple pattern is colorfully expressed in red, green, and blue. We also calculated the saturation of the source photograph (Figure 6) and the interference-ripple image (Figure 7). In calculating pixel saturation, we let $p_{\min}$ and $p_{\max}$

(a) Airplane (b) Balloon (c) Earth (d) Girl

(e) Lenna (f) Mandrill (g) Parrots (h) Sailboat

Figure 6. Photographies used in the experiment (color online)



(a) Airplane (b) Balloon (c) Earth (d) Girl

(e) Lenna (f) Mandrill (g) Parrots (h) Sailboat

Figure 7. Interference-ripple images (color online)

Table 1. Saturation of photographies and interference-ripple images

Name	Photography	Interference-ripple image
(a) Airplane	0.126	0.431
(b) Balloon	0.136	0.560
(c) Earth	0.313	0.694
(d) Girl	0.326	0.678
(e) Lenna	0.520	0.851
(f) Mandrill	0.319	0.688
(g) Parrots	0.349	0.784
(h) Sailboat	0.333	0.643

be the minimum and maximum RGB values, respectively, and calculate the average $ave = (p_{\min}+p_{\max})/2$. If ave is 127.5 or less, the saturation is taken as $(p_{\max}-p_{\min})/(p_{\max}+p_{\min})$, and otherwise as $(p_{\max} - p_{\min})/(510 - p_{\max} - p_{\min})$. The saturation of the overall image is calculated as the average value of all pixels. The saturation of the image takes a value from 0 to 1, with higher values indicating higher saturation. Table 1 shows saturation values for the source and interference-ripple images. Table 1 shows that all interference-ripple images are more saturated than the source photographs. The saturation of the interference-ripple images (a), (b), (d), and others are about 3.0, 4.0, 1.5, and 2.0 times higher than the source photographs, respectively.

Finally, the interference-ripple image (Figure 3) and interference image (Figure 4) of the proposed methods and the interference-ripple image (Figure 7(e)) of the proposed method are compared. The images of the conventional method are harder to see as a wavy smooth curve and have a narrow pattern spacing than the image of the proposed method. The saturation of Figures 3 and 4 are 0.889 and 0.614, respectively. Compared with the saturation 0.851 of Figure 7(e), Figure 3 has the same degree of saturation and Figure 4 has greatly low saturation.

4. **Conclusions.** We proposed a method for producing NPR images that generates interference-ripple images through iterative processing of photographs using a inverse Sobel filter. To verify the effectiveness of the proposed method, we examined the visual effect in interference-ripple images generated by applying the proposed method to various photographs. Experimental results show that the proposed method automatically generates a smooth interference-ripple pattern that follows outlines in the source photograph. We also found that saturation of the photography is increased, automatically generating colorful interference-ripple images. However, we also found that it is difficult to generate interference ripples in some areas.

One future task is improving the proposed method so that interference-ripple patterns can also be generated in regions where they do not easily appear. Another topic for investigation is applying the proposed method to video images and distance images.

REFERENCES

[1] H. Pedersen and K. Singh, Organic labyrinths and mazes, *NPAR'6*, pp.76-86, 2006.

[2] A. Sparavigna and B. Montrucchio, *Non-photorealistic Image Rendering with a Labyrinthine Tiling*, Cornell University Library, arXiv:cs/0609084, 2006.

[3] H. Zhao, F. Gu, Q.-X. Huang, J. Garcia, Y. Chen, C. Tu, B. Benes, H. Zhang, D. Cohen-Or and B. Chen, Connected fermat spirals for layered fabrication, *Proc. of ACM SIGGRAPH 2016*, vol.35, no.4, p.100, 2016.

[4] A. Hausner, Pointillist halftoning, *Proc. of CGI*, pp.134-139, 2005.

[5] C. Yang and H. Yang, Realization of Seurat's pointillism via non-photorealistic rendering, *The Visual Computer*, vol.24, no.5, pp.303-322, 2008.

[6] J. Sugita and T. Takahashi, A method to generate hybrid pointillistic image, *Proc. of ACM SIGGRAPH Asia 2015*, Posters, 2015.

[7] J. Weickert, Coherence-enhancing shock filters, *Pattern Recognition Lecture Notes in Computer Science*, vol.2781, pp.1-8, 2003.

[8] T. Hiraoka and K. Urahama, Generating striped color images by inverse line convergence index filter, *IIEEJ Trans. Image Electronics and Visual Computing*, vol.2, no.2, pp.190-194, 2014.

[9] 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.

[10] J. M. Ortega and W. C. Rheinboldt, Iterative solutions of nonlinear equations in several variables, *Society for Industrial and Applied Mathematics*, 1987.

[11] Z. Yu and K. Urahama, Iterative method for inverse nonlinear image processing, *IEICE Trans. Fundamentals*, vol.E97-A, no.2, pp.719-721, 2014.

[12] T. Hiraoka and K. Urahama, Generating ripple-pattern-like images by inverse Sobel filter, *Journal of the Institute of Image Electronics Engineers of Japan*, vol.44, no.2, pp.349-352, 2015.

[13] T. Hiraoka and K. Urahama, Generation of oil-film-like images by bilateral infra-envelope filter, *IEICE Trans. Information and Systems*, vol.E99-D, no.6, pp.1724-1728, 2016.