ICIC Express Letters ICIC International ©2018 ISSN 1881-803X
Volume **12**, Number **11**, November **2018** pp. 1131–1136

GENERATION OF PARALLEL-FINE-CURVE-LINE IMAGES BY ITERATIVE CALCULATION USING CORRELATION COEFFICIENT

Toru Hiraoka[1], Tetsuya Katayama[1] and Kiichi Urahama[2]

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

[2]Department of Communication Design Science
Kyushu University
4-9-1, Shiobaru, Minami-ku, Fukuoka-shi, Fukuoka-ken 815-8540, Japan
urahama@design.kyushu-u.ac.jp

Received June 2018; accepted August 2018

Abstract. *We propose a visual effect method for generating parallel-fine-curve-line (PFCL) images from photographic images. PFCL images have smooth fine curve lines aligned along edges of photographic images. Our method is executed by an iterative calculation using correlation coefficient. To verify the visual effects of PFCL images, we conduct experiments with varying the value of parameters of our method and experiments using various photographic images. As a result of experiments, we clarified the optimal parameter values of our method and confirmed that the PFCL patterns can be automatically generated on the whole image with all photographic images.*
Keywords: Visual effects, Parallel-fine-curve-line, Correlation coefficient, Iterative calculation

1. **Introduction.** Techniques for processing images using computer graphics have appeared since the 1980's, and have been used in movies and televisions. Such techniques are called visual effects. In recent years, visual effects are used in applications which are embedded in personal computers and portable terminals. In addition, mobile phones and smart phones equipped with cameras have appeared, and anyone can easily apply special effects to photographic images. By emphasizing the brightness and saturation of photographic images as such special effects, it is possible to make the face of a person look beautiful and make the dish look delicious. On the other hand, many researches [1, 2, 3, 4, 5, 6, 7] have also been done to convert photographic images into new expression images that are impossible in the real world and are not in the art expression of the past. Weickert [1] generates flow-pattern images, Sparavigna and Montrucchio [2] and Inoue and Urahama [3] generate labyrinthine images, Inglis et al. [4] generate op-art images with lines and curves, Chi et al. [5] generate reaction-diffusion-pattern images, and Hiraoka et al. [6, 7] generate cell-like and interference-ripple images.

In this paper, we develop a new visual effect method for generating new expression images from photographic images. As new expression images, we propose parallel-fine-curve-line (PFCL) images. PFCL images have smooth fine curve lines aligned along edges of photographic images. PFCL images are completely different from the images generated by the conventional methods [1, 2, 3, 4, 5, 6, 7]. Our method is executed by an iterative calculation using correlation coefficient. Our method has characteristic that PFCL patterns can be automatically generated in accordance with the luminance,

DOI: 10.24507/icicel.12.11.1131

shading, and edge of photographic images. To verify the visual effects of PFCL images, we conduct experiments with varying the value of parameters of our method. And, we also conduct experiments using various photographic images. As a result of experiments, we found that our method can automatically generate PFCL images, and how to generate the PFCL patterns by varying the value of parameters.

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

2. **Our Method.** Our method generates PFCL images from photographic images. Our method is executed in two processes. In the first process, we calculate correlation coefficient from photographic images. In the second process, we convert photographic images using correlation coefficient. PFCL images are generated by repeating the two processes. We show a flow chart of our method in Figure 1.

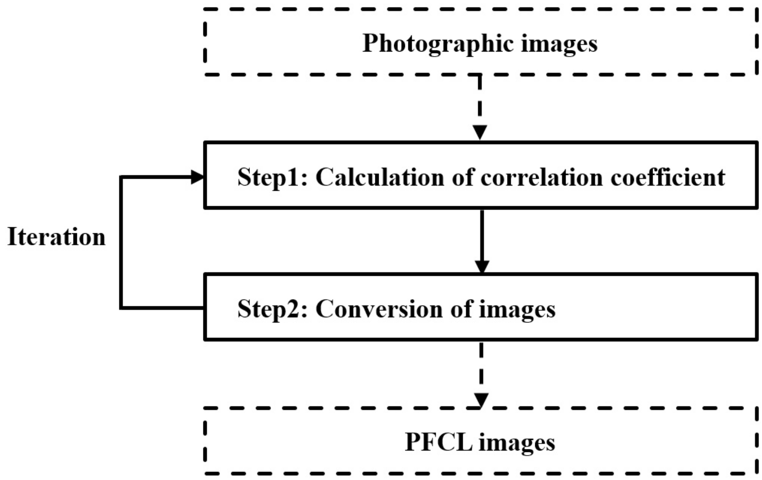


Figure 1. Flow chart of our method

We show the detailed procedure of our method as follows.

Step 0: Let the input pixel values on coordinates (i, j) of a gray-scale photographic image be $f_{i,j}$. The pixel values $f_{i,j}$ have value of 256 gradation from 0 to 255.

Step 1: Correlation coefficients are calculated using pixel values $f_{i,j}^{(t)}$ in the window of $2W + 1$ centered on the pixel (i, j), where t $(= 1, 2, \ldots)$ is the number of iterations and $f_{i,j}^{(1)} = f_{i,j}$. The averages $a_{i,j}^{(t)}$ of the pixel values in the window are calculated as follows.

$$a_{i,j}^{(t)} = \frac{\sum_{k=-W}^{W} \sum_{l=-W}^{W} f_{i+k,j+l}^{(t)}}{(2W+1)^2} \tag{1}$$

Vectors $\vec{v}_{1,i,j}^{(t)}$ are created by arranging the pixel values in the window as follows.

$$\begin{aligned} \vec{v}_{1,i,j}^{(t)} = \Big(& f_{i-W,j-W}^{(t)} - a_{i,j}^{(t)}, f_{i-W+1,j-W}^{(t)} - a_{i,j}^{(t)}, f_{i-W+2,j-W}^{(t)} - a_{i,j}^{(t)}, \ldots, \\ & f_{i-2,j}^{(t)} - a_{i,j}^{(t)}, f_{i-1,j}^{(t)} - a_{i,j}^{(t)}, f_{i,j}^{(t)} - a_{i,j}^{(t)}, f_{i+1,j}^{(t)} - a_{i,j}^{(t)}, f_{i+2,j}^{(t)} - a_{i,j}^{(t)}, \ldots, \\ & f_{i+W-2,j+W}^{(t)} - a_{i,j}^{(t)}, f_{i+W-1,j+W}^{(t)} - a_{i,j}^{(t)}, f_{i+W,j+W}^{(t)} - a_{i,j}^{(t)} \Big) \end{aligned} \tag{2}$$

Vectors $\vec{v}_{2,i,j}^{(t)}$ are created by inverting the elements of vectors $\vec{v}_{1,i,j}^{(t)}$ as follows.

$$\vec{v}_{2,i,j}^{(t)} = \Big(f_{i+W,j+W}^{(t)} - a_{i,j}^{(t)}, f_{i+W-1,j+W}^{(t)} - a_{i,j}^{(t)}, f_{i+W-2,j+W}^{(t)} - a_{i,j}^{(t)}, \ldots,$$

$$f_{i+2,j}^{(t)} - a_{i,j}^{(t)}, f_{i+1,j}^{(t)} - a_{i,j}^{(t)}, f_{i,j}^{(t)} - a_{i,j}^{(t)}, f_{i-1,j}^{(t)} - a_{i,j}^{(t)}, f_{i-2,j}^{(t)} - a_{i,j}^{(t)}, \ldots, \quad (3)$$
$$f_{i-W+2,j-W}^{(t)} - a_{i,j}^{(t)}, f_{i-W+1,j-W}^{(t)} - a_{i,j}^{(t)}, f_{i-W,j-W}^{(t)} - a_{i,j}^{(t)}\Big)$$

Correlation coefficients $c_{i,j}^{(t)}$ are calculated using vectors $\vec{v}_{1,i,j}^{(t)}$ and $\vec{v}_{2,i,j}^{(t)}$ as follows.

$$c_{i,j}^{(t)} = \frac{\vec{v}_{1,i,j}^{(t)} \cdot \vec{v}_{2,i,j}^{(t)}}{\left|\vec{v}_{1,i,j}^{(t)}\right| \left|\vec{v}_{2,i,j}^{(t)}\right|} \quad (4)$$

Step 2: The pixel values $f_{i,j}^{(t+1)}$ using correlation coefficients $c_{i,j}^{(t)}$ are converted as

$$f_{i,j}^{(t+1)} = f_{i,j} + bc_{i,j}^{(t)} \quad (5)$$

where b is a positive constant. The pixel values $f_{i,j}^{(t+1)}$ are set to 0 if the values are less than 0, and set to 255 if the values are greater than 255.

Step 3: The PFCL image is generated after T times iteration of Steps 1 and 2.

3. **Experiments.** First, we visually confirm PFCL images by varying the value of parameters T, W, and b. In the experiments, we use Lenna image shown in Figure 2. Next, we visually verify PFCL images generated from various photographic images. All photographic images used in these experiments are $512 * 512$ size and 256 gradation.

3.1. **Experiments with varying parameters.** We visually confirm PFCL images generated by varying the value of the iteration number T using Lenna image. The value of T is set to 1, 3, 5, and 7. The values of the parameters W and b are set to 4 and 120, respectively. The results of the experiment are shown in Figure 3. As the value of T is larger, the PFCL patterns become clearer and are expressed finely. And, as the value of

Figure 2. Lenna image

(a) $T = 1$ (b) $T = 3$ (c) $T = 5$ (d) $T = 7$

Figure 3. PFCL images generated by varying the value of the iterative number T

(a) $W = 2$ (b) $W = 4$ (c) $W = 6$ (d) $W = 8$

FIGURE 4. PFCL images generated by varying the value of the window size W

(a) $b = 40$ (b) $b = 80$ (c) $b = 120$ (d) $b = 160$

FIGURE 5. PFCL images generated by varying the value of the parameter b

T is around 5, the PFCL patterns are generated on the whole image. However, as the value of T is larger, PFCL images and Lenna image increase in difference. We think that the value of T is around 5.

We visually confirm PFCL images generated by varying the value of the window size W using Lenna image. The value of W is set to 2, 4, 6, and 8. The values of the parameters T and b are set to 5 and 120, respectively. The results of the experiment are shown in Figure 4. As the value of W is larger, the PFCL patterns become wider. However, as the value of W is larger, PFCL images and Lenna image increase in difference. We think that the value of W is around 4. And, we think that the value of W may be changed in accordance with the purpose of use.

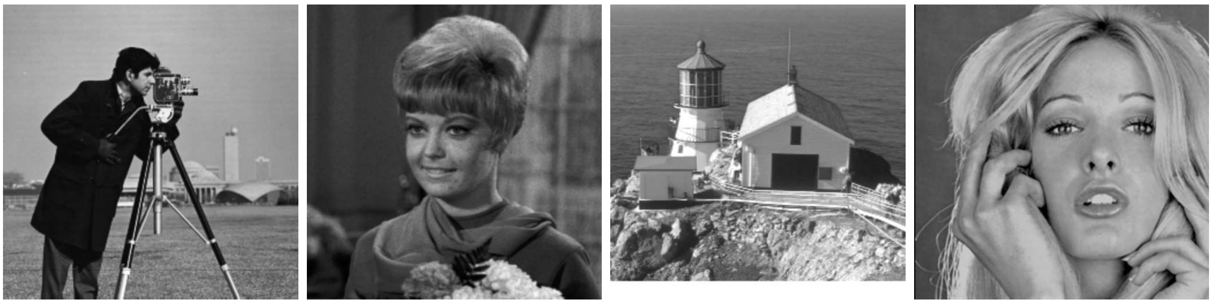
We visually confirm PFCL images generated by varying the parameter b using Lenna image. The value of b is set to 40, 80, 120, and 160. The values of the parameters T and W are set to 5 and 4, respectively. The results of the experiment are shown in Figure 5. As the value of b is larger, the PFCL patterns become clear. However, as the value of b is larger, PFCL images and Lenna image increase in difference. Thus, we think that the value of b is around 120.

For reference, we show an enlarged view of PFCL image with $T = 5$, $W = 4$, and $b = 120$ in Figure 6.

3.2. **Experiment using various photographic images.** We apply our method to eight photographic images shown in Figure 7. The values of the parameters T, W, and b are set to 5, 4, and 120, respectively. The results of the experiment are shown in Figure 8. The PFCL patterns are automatically generated in accordance with the luminance, shading, and edge of photographic images throughout entire regions for all PFCL images. Also, PFCL patterns can be generated in regions where brightness change is small such as white regions at the bottom of Figures 8(d) and 8(e). In the dark region such as black clothing in Figure 8(e), PFCL patterns are difficult to see for people.

Figure 6. Enlarged view of PFCL image with $T = 5$, $W = 4$, and $b = 120$

(a) Airplane (b) Boat (c) Bridge (d) Building

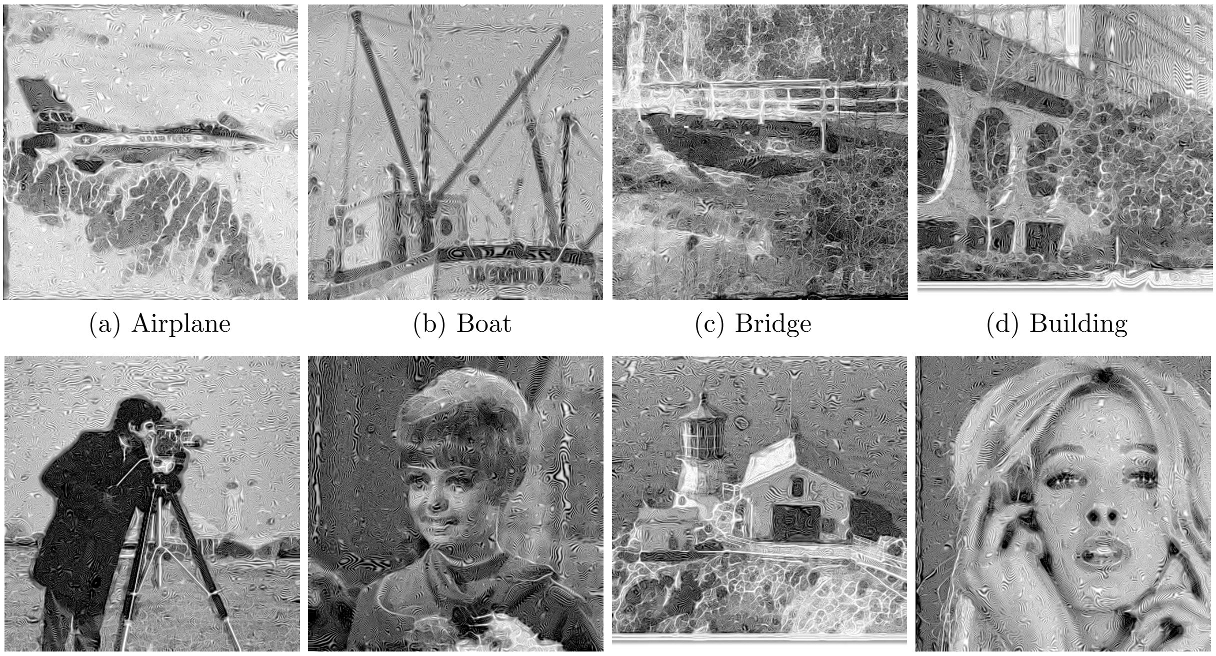


(e) Cameraman (f) Girl (g) Lighthouse (h) Woman

Figure 7. Various photographic images

4. **Conclusions.** We proposed a visual effect method for generating PFCL images from photographic images. Our method was executed in two processes. The first process calculated correlation coefficient from photographic images. The second process converted photographic images using correlation coefficient. In experiments, we visually confirmed PFCL images by varying the value of parameters using Lenna image, and verify PFCL images generated from various photographic images. As a result of the experiments, we found the values of the optimal parameters to generate PFCL images, and found that PFCL patterns can be automatically generated in accordance with the luminance, shading, and edge of photographic images.

In future work, we will try to apply our method to color photographic images and videos.



(a) Airplane (b) Boat (c) Bridge (d) Building

(e) Cameraman (f) Girl (g) Lighthouse (h) Woman

FIGURE 8. PFCL images generated from various photographic images

REFERENCES

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

[2] A. Sparavigna and B. Montrucchio, Non-photorealistic image rendering with a labyrinthine tiling, *arXiv:cs/0609084*, Cornell University Library, 2006.

[3] K. Inoue and K. Urahama, Halftoning with minimum spanning trees and its application to maze-like images, *Computers & Graphics*, vol.33, no.5, pp.638-647, 2009.

[4] T. C. Inglis, S. Inglis and C. S. Kaplan, Op-art rendering with lines and curves, *Computers & Graphics*, vol.36, no.6, pp.607-621, 2012.

[5] M. T. Chi, W. C. Liu and S. H. Hsu, Image stylization using anisotropic reaction diffusion, *The Visual Computer*, vol.32, no.12, pp.1549-1561, 2016.

[6] T. Hiraoka, M. Hirota, K. Inoue and K. Urahama, Generating cell-like color images by inverse iris filter, *ICIC Express Letters*, vol.11, no.2, pp.399-404, 2017.

[7] T. Hiraoka, Generation of interference-ripple images by inverse sobel filter, *ICIC Express Letters*, vol.12, no.5, pp.409-415, 2018.