ICIC Express Letters
Volume **13**, Number **2**, February **2019**
 ISSN 1881-803X
pp. **127–132**

GENERATION OF CONCRETE-WALL-TEXTURE-LIKE IMAGES BY AUTOCORRELATION COEFFICIENT AND INVERSE FILTERING

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 August 2018; accepted November 2018

Abstract. *This paper proposes a non-photorealistic rendering (NPR) method for generating concrete-wall-texture-like (CWTL) images from photographic images. CWTL images imitate uneven concrete walls. The proposed method is executed by an iterative calculation using correlation coefficient and inverse filtering. To verify the visual effects of CWTL images, experiments are conducted with changing the value of parameters of the proposed method and experiments using various photographic images. As a result of experiments, it is clarified how to generate the CWTL patterns by changing the value of parameters and confirmed that the CWTL patterns can be generated over the whole image for all photographic images.*
Keywords: Non-photorealistic rendering, Concrete wall texture, Autocorrelation coefficient, Inverse filtering

1. **Introduction.** NPR for converting photographic images and 3-dimensional models to oil painting [1, 2, 3], pencil drawing [4, 5, 6], and charcoal drawing [7, 8, 9] images has been the subject of many studies. In recent years, such NPR is incorporated in applications such as Adobe photoshop and Clip studio paint pro can be easily used by non-professional users. In addition of NPR of art expression so far, Adobe photoshop has new NPR for converting ripple-like and crystal-like images from photographic images. Many studies of such new NPR have been conducted for generating flow-pattern [10], line-pattern [11], labyrinthine [12, 13], op-art [14], reaction-diffusion-pattern [15, 16], cell-like [17], and interference-ripple [18] images. Flow-pattern image [10] is generated using shock filtering with the robust orientation estimation by means of the structure tensor. Line-pattern image [11] is generated using joint singular value decomposition and unsharp mask. Labyrinthine image [13] is generated by combining a labyrinthine generation algorithm and graph patterns designed by designers. Op art image [14] is composed of straight lines and curves, and is generated by a global optimization technique and an algorithm based on a physical simulation of heat flow. To generate reaction-diffusion-pattern images [15, 16], the mathematical model used in biology is used. Cell-like [17] and interference-ripple [18] images are generated by inverse iris filter and inverse Sobel filter, respectively.

In this paper, we propose a new NPR method for generating CWTL images from photographic images. CWTL images imitate uneven concrete walls shown in Figure 1. CWTL images are completely different from the images generated by the conventional methods

DOI: 10.24507/icicel.13.02.127

FIGURE 1. Example of texture of concrete wall

[10, 11, 12, 13, 14, 15, 16, 17, 18]. Our method is executed by an iterative calculation using inverse filtering [19, 20] and autocorrelation coefficient. By using our method, CWTL patterns are automatically generated in accordance with the luminance, shading, and edge of photographic images. And, our method can preserve edges of photographic images. To verify the visual effects of CWTL images, we conduct experiments with changing the value of parameters of our method and experiments conduct experiments using various photographic images. As a result of experiments, we found that our method can automatically generate CWTL images, and clarified how to generate the CWTL patterns by changing the value of parameters.

The rest of this paper is organized as follows. Section 2 describes our method for generating CWTL 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 CWTL images from photographic images. Our method is executed in three processes. In the first process, we convert photographic images using autocorrelation coefficient. Autocorrelation coefficient is a feature quantity of texture. From autocorrelation coefficient, it is possible to see the roughness, periodicity, and directionality of texture. In the second process, we restore the converted images to photographic images using inverse filtering. Inverse filtering is calculated by a procedure that restores an image by a processing to an original image. The restored images include a restoration error. In the third process, we convert the value of the pixels. The restoration error is emphasized by repeating the three processes, and CWTL images are generated. We show a flow chart of our method in Figure 2.

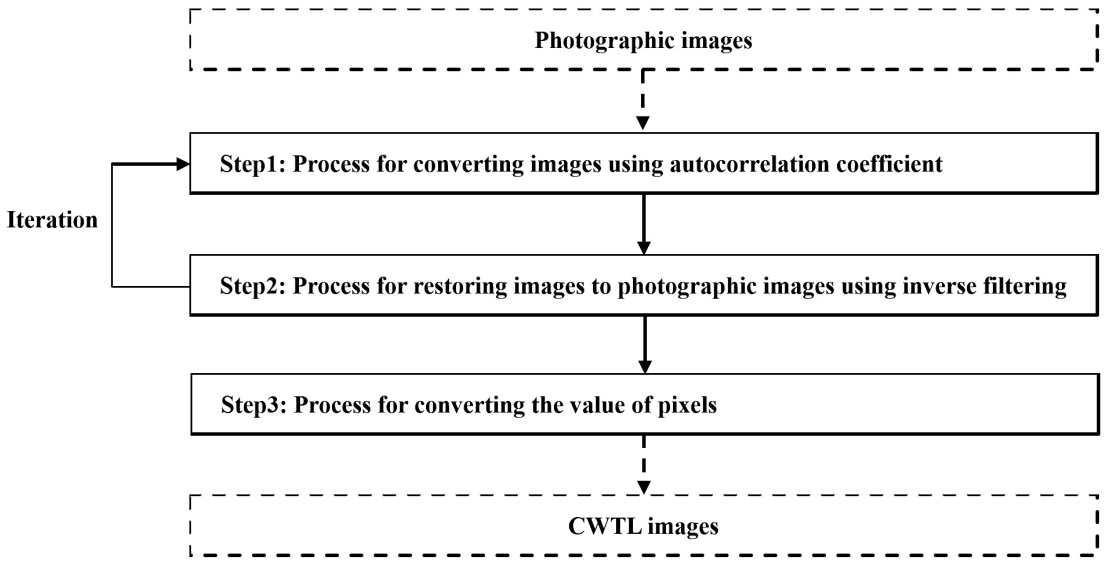


FIGURE 2. Flow chart of our method

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: Let the vectors that are created by arranging the pixel values in the window of $2W + 1$ centered on the pixel (i, j) be $\vec{v}_{i,j}^{(t)}$ as

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

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

where $t\ (= 1, 2, \ldots)$ is the number of iterations, $f_{i,j}^{(1)} = f_{i,j}$, and $a_{i,j}^{(t)}$ are the averages of the pixel values in the window. Autocorrelation coefficients $c_{i,j,k,l}^{(t)}$ are calculated from the area in the window of $2W + 1$ centered on the pixel (i, j) and the pixel $(i + k, j + l)$ that is (k, l) pixel away from the pixel (i, j) as follows.

$$c_{i,j,k,l}^{(t)} = \frac{\vec{v}_{i,j}^{(t)} \cdot \vec{v}_{i+k,j+l}^{(t)}}{\left|\vec{v}_{i,j}^{(t)}\right| \left|\vec{v}_{i+k,j+l}^{(t)}\right|} \tag{3}$$

Averages of autocorrelation coefficients $C_{i,j}^{(t)}$ in the window centered on the pixel (i, j) are calculated as

$$C_{i,j}^{(t)} = \frac{\sum_{k=-W}^{W}\sum_{l=-W}^{W} c_{i,j,k,l}^{(t)}}{(2W+1)^2 - 1} \tag{4}$$

where it is not included in the calculation when $k = 0$ and $l = 0$. The pixel values $AC\left(f_{i,j}^{(t)}\right)$ converted using correlation-coefficient averages $C_{i,j}^{(t)}$ are computed as

$$AC\left(f_{i,j}^{(t)}\right) = 127.5 + 127.5\frac{C_{i,j}^{(t)}}{C_{\max}^{(t)}} \tag{5}$$

where $C_{\max}^{(t)}$ is the maximum value of the absolute value of $C_{i,j}^{(t)}$.

Step 2: The pixel values $f_{i,j}^{(t+1)}$ using inverse filtering are computed as follows.

$$f_{i,j}^{(t+1)} = f_{i,j}^{(t)} - AC\left(f_{i,j}^{(t)}\right) + f_{i,j} \tag{6}$$

Step 3: The processes of Steps 1 and 2 are repeated T times. The pixel values $f_{i,j}^{(T)}$ are converted to a value from 0 to 255 as

$$g_{i,j}^{(T)} = 255\frac{f_{i,j}^{(T)} - f_{\min}^{(T)}}{f_{\max}^{(T)} - f_{\min}^{(T)}} \tag{7}$$

where $f_{\min}^{(T)}$ and $f_{\max}^{(T)}$ are the minimum and maximum values of $f_{i,j}^{(T)}$, respectively. The image of the pixels values $g_{i,j}^{(T)}$ is the CWTL image.

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

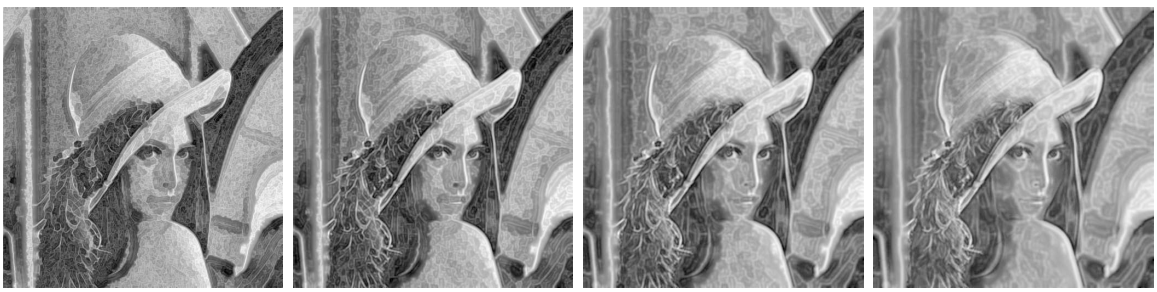
FIGURE 3. Lenna image

3.1. **Experiments with changing parameters.** We visually confirm CWTL images generated by changing the value of the iteration number T using Lenna image. The value of T is set to 1, 3, 5, and 7. The value of the parameter W is set to 2. The results of the experiment are shown in Figure 4. As the value of T is larger, the CWTL patterns become clearer and are expressed finely. As the value of T is around 5, the CWTL images are generated on the whole image. We think that the value of T is around 5.

We visually confirm CWTL images generated by changing the value of the window size W using Lenna image. The value of W is set to 1, 2, 3, and 4. The value of the parameter T is set to 5. The results of the experiment are shown in Figure 5. As the value of W is larger, the CWTL patterns become bigger. However, as the value of W is larger, CWTL

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

FIGURE 4. CWTL images generated by changing the value of the iterative number T

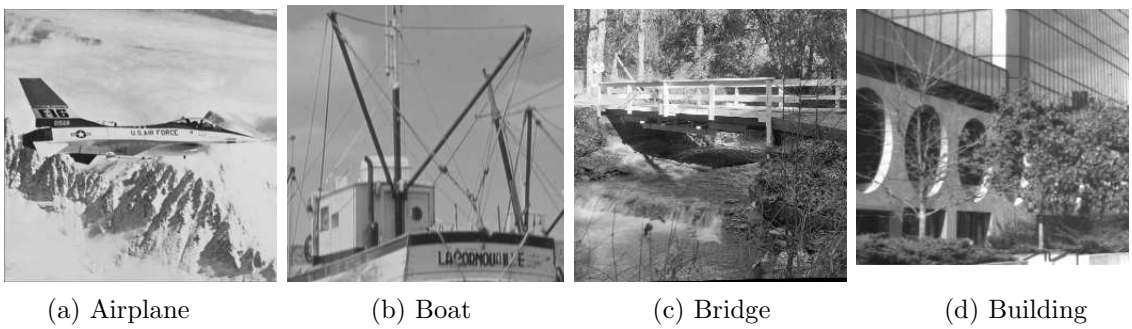


(a) $W = 1$ (b) $W = 2$ (c) $W = 3$ (d) $W = 4$

FIGURE 5. CWTL images generated by changing the value of the window size W

images and Lenna image increase in difference. We think that the value of T should be 1 or 2.

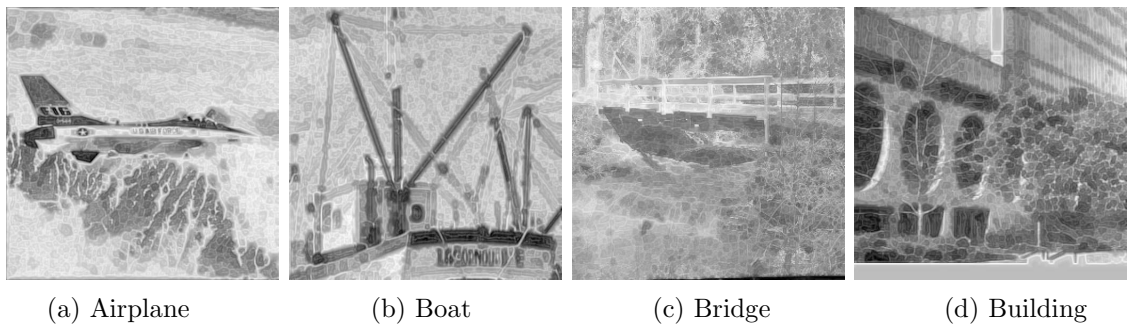
3.2. **Experiment using various photographic images.** We apply our method to eight photographic images shown in Figure 6. The values of the parameters T and W are set to 5 and 2, respectively. The results of the experiment are shown in Figure 7. The CWTL patterns are automatically generated in accordance with the luminance, shading, and edge of photographic images throughout entire regions for all CWTL images, and the edges of photographic images are preserved in all CWTL images. Therefore, if the values of the parameters T and W are set to 5 and 2, we think that our method can generate CWTL images for various photographic images.



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

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

FIGURE 6. Various photographic images



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

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

FIGURE 7. CWTL images generated from various photographic images

4. **Conclusions.** We proposed an NPR method for generating CWTL images from photographic images. Our method was executed in three processes. The first process converted photographic images using autocorrelation coefficient. The second process restored the converted images to photographic images using inverse filtering. The third process converted the value of the pixels. In experiments, we visually confirmed CWTL images by changing the value of parameters using Lenna image, and verified CWTL images generated from various photographic images. As a result of the experiments, we found the values of the optimal parameters to generate CWTL images. And we found that CWTL patterns can be automatically generated in accordance with the luminance, shading, and edge of photographic images, and the edges of photographic images can be preserved in CWTL images.

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

REFERENCES

[1] B. J. Meier, Painterly rendering for animation, *Proc. of ACM SIGGRAPH*, pp.477-484, 1996.
[2] Y. Tsunematsu, N. Kawai, T. Sato and N. Yokoya, Texture transfer based on energy minimization for painterly rendering, *Journal of Information Processing*, vol.24, no.2, pp.897-907, 2016.
[3] S. Kodama, P. Poulin, T. Moriya and T. Takahashi, Creativity enhancement of painterly rendering using a suggestive interface, *Computers & Graphics*, vol.71, pp.42-54, 2018.
[4] M. C. Sousa and J. W. Buchanan, Computer-generated graphite pencil rendering of 3D polygonal models, *Computer Graphics Forum*, vol.18, no.3, pp.195-208, 1999.
[5] S. X. Man, A multi-parameter generating method for colored pencil rendering, *American Journal of Information Science and Computer Engineering*, vol.2, no.5, pp.53-57, 2016.
[6] Z. Chen, Y. Jin, B. Sheng, P. Li and H. Sun, Parallel pencil drawing stylization via structure-aware optimization, *Proc. of the 31st International Conference on Computer Animation and Social Agents*, pp.32-37, 2018.
[7] A. Majumder and M. Gopi, Hardware accelerated real time charcoal rendering, *Proc. of the 2nd International Symposium on Non-Photorealistic Animation and Rendering*, pp.59-66, 2002.
[8] H. Johan, H. Matsui, T. Haga, Y. Dobashi and T. Nishita, A method for creating region-based and stroke-based artistic images, *IEICE Trans. Information and Systems*, vol.J88-D-II, no.2, pp.358-367, 2005.
[9] Q. Zhao and W. H. Lee, The application of traditional Chinese painting technique and stroke effect in digital ink painting, *Journal of Arts and Imaging Science*, vol.5, no.2, pp.35-42, 2018.
[10] J. Weickert, Coherence-enhancing shock filters, *Pattern Recognition Lecture Notes in Computer Science*, vol.2781, pp.1-8, 2003.
[11] T. Hiraoka and K. Urahama, Generation of line-pattern image by joint singular value decomposition and unsharp mask, *ICIC Express Letters*, vol.12, no.6, pp.511-517, 2018.
[12] 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.
[13] W. S. Chou, Rectangular maze construction by combining algorithms and designed graph patterns, *Journal on Computing*, vol.5, no.1, pp.35-39, 2016.
[14] 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.
[15] 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.
[16] C. W. Jho and W. H. Lee, Real-time tonal depiction method by reaction-diffusion mask, *Journal of Real-Time Image Processing*, vol.13, no.3, pp.591-598, 2017.
[17] 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.
[18] T. Hiraoka, Generation of interference-ripple images by inverse Sobel filter, *ICIC Express Letters*, vol.12, no.5, pp.409-415, 2018.
[19] J. M. Ortega and W. C. Rheinboldt, Iterative solutions of nonlinear equations in several variables, *Society for Industrial Mathematics*, 1987.
[20] Z. Yu and K. Urahama, Iterative method for inverse nonlinear image processing, *IEICE Trans. Fundamentals*, vol.E97-A, no.2, pp.719-721, 2014.