ICIC Express Letters
Volume **14**, Number **12**, December **2020**

ICIC International ©2020 ISSN 1881-803X
pp. 1163–1167

GENERATION OF SAND-STYLE IMAGES USING HASH FUNCTION AND INVERSE 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 March 2020; accepted June 2020

Abstract. *Until now, non-photorealistic rendering methods for generating various sand-style images have been proposed. In this paper, we focus on sand-style images that are studded with sand, and aim to develop a method that is easy to process, considering the processing on the mobile terminal. Our method generates sand-style images from photographic images, and is executed by an iterative processing using hash function and inverse filter. Our method is simple in processing and easy to implement. We conducted experiments using various photographic images to verify the effectiveness of our method. The experimental results show that our method can generate impressive sand-style images. Potential applications of our method range from magazines, TVs, websites, to social networking services.*

Keywords: Non-photorealistic rendering, Sand-style image, Hash function, Inverse filter

1. **Introduction.** Many studies have been conducted on non-photorealistic rendering that converts photographic images and three-dimensional models to non-realistic images such as ink paintings, watercolors, and cartoons [1-7]. Similarly, a study on non-photorealistic rendering of sand paintings is also conducted [8-11]. Among the conventional methods, there are the methods [8, 9] to develop paint tools that allow users to create sand-style images from scratch, and the methods [10, 11] that automatically generate sand-style images from photographic images. In this paper, we focus on the latter method. The method [10] is executed by an iterative processing using bilateral filter and unsharp mask, and the method [11] is done by three main steps including image data field generating, multiscale field coupling, and additional details optimizing. These methods require many computational cost in order to use an exponential function. Among the expressions of sand-style images from photographic images, there are the expression [11] that is studded with sand, and the expression [10] that is traced sand with fingers and sticks. In this paper, we focus on the former expression.

We propose a new method for generating sand-style images from gray-scale photographic images. In sand-style images generated by our method, light areas contain less fine particles, and dark areas contain more fine particles. In addition, the edges of photographic images are also preserved in sand-style images. Our method is executed by an iterative processing using inverse filter [12, 13] and hash function. Images with fine patterns such as sand can be generated by performing processing using hash function, and the shades and edges of photographic images can be expressed by inverse filter. As the iterative processing proceeds, sand-style patterns are gradually expressed, and finally sand-style images are generated. Our method is simple in processing, and has low computational cost in order not to use the exponential function. To verify the effectiveness of our method, we conducted experiments to visually confirm sand-style patterns by using

DOI: 10.24507/icicel.14.12.1163

various photographic images. As a result of the experiments, we show that our method can generate desired sand-style images.

This paper is organized as follows: the second section describes our method for generating sand-style images, the third section shows experimental results and reveals the effectiveness of our method, and the conclusion of this paper is given in the fourth section.

2. **Our Method.** Our method can generate sand-style images from gray-scale photographic images by the iterative processing using hash function and inverse filter. Hash function calculates a representative numerical value from certain data. Inverse filter restores images transformed by a processing to original images. Our method is executed in two steps: the first step uses hash function to transform images, and the second step uses inverse filter to restore the transformed images to original photographic images. Sand-style patterns are generated by emphasizing the errors that occur by repeating the two steps. It turns out that our method does not use particularly difficult processing. The flow chart of our method is shown in Figure 1.

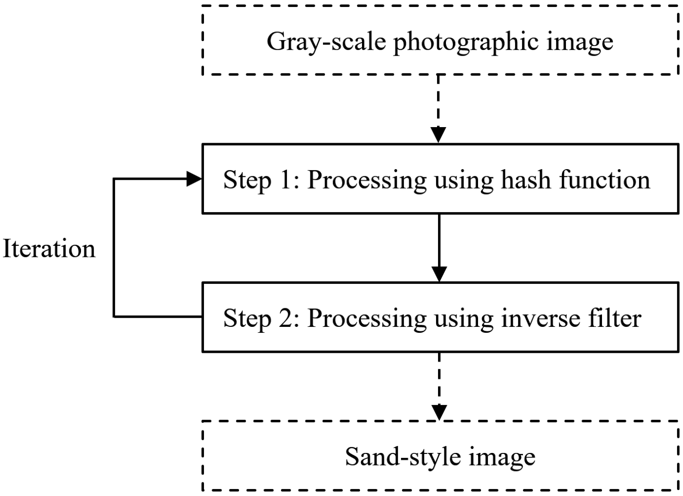

Gray-scale photographic image
Step 1: Processing using hash function
Iteration
Step 2: Processing using inverse filter
Sand-style image

FIGURE 1. Flow chart of our method

We explain the details of the procedure of our method.

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

Step 1: The output pixel values $h_{i,j}^{(t)}$ using hash function are calculated by the following equation.

$$h_{i,j}^{(t)} = \sum_{k=i-W}^{i+W} \sum_{l=j-W}^{j+W} f_{k,l}^{(t-1)} \bmod N \tag{1}$$

where W is the window size, and k and l are the positions in the window. By using Equation (1), it is possible to generate blackish points in light areas and white points in dark areas.

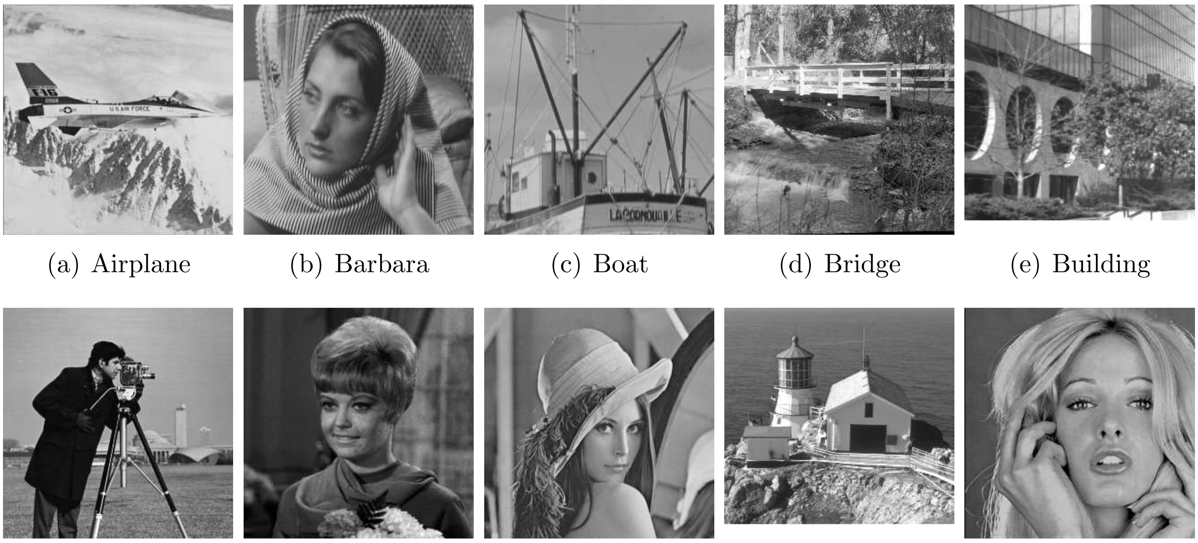
Step 2: The output pixel values $f_{i,j}^{(t+1)}$ using inverse filter are calculated from the pixel values $h_{i,j}^{(t)}$ by the following equation.

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

If $f_{i,j}^{(t+1)}$ is less than 0, then $f_{i,j}^{(t+1)}$ must be set to 0. If $f_{i,j}^{(t+1)}$ is greater than $N-1$, then $f_{i,j}^{(t+1)}$ must be set to $N-1$. By using Equation (2), it is possible to preserve the shades and edges of photographic image.

The processing of Steps 1 and 2 is repeated T times, and then an image composed of the pixel values $f_{i,j}^{(T)}$ is the sand-style image.

3. **Experiments.** We applied our method to ten gray-scale photographic images shown in Figure 2. All photographic images used in the following experiments were $512 * 512$ pixels and 256 gradation. Even if the value of the window size W was changed, there was no significant difference in sand-style images. However, by increasing the value of the window size W, sand-style patterns were more finely expressed in the areas where the change in the pixel values of photographic images were small. In the following experiments, we set the value of the window size W to 2. As the value of the iteration number T was increased, sand-style patterns were generated. In the following experiments, we set the value of the iteration number T to 100 with a margin so that sand-style images could be obtained. The computing environment for all experiments was a Windows 10 Enterprise 2016 LTSB operating system on a computer with a 3.20 GHz CPU and a 8.00 GB of memory. The programming language used in all experiments was VC++.



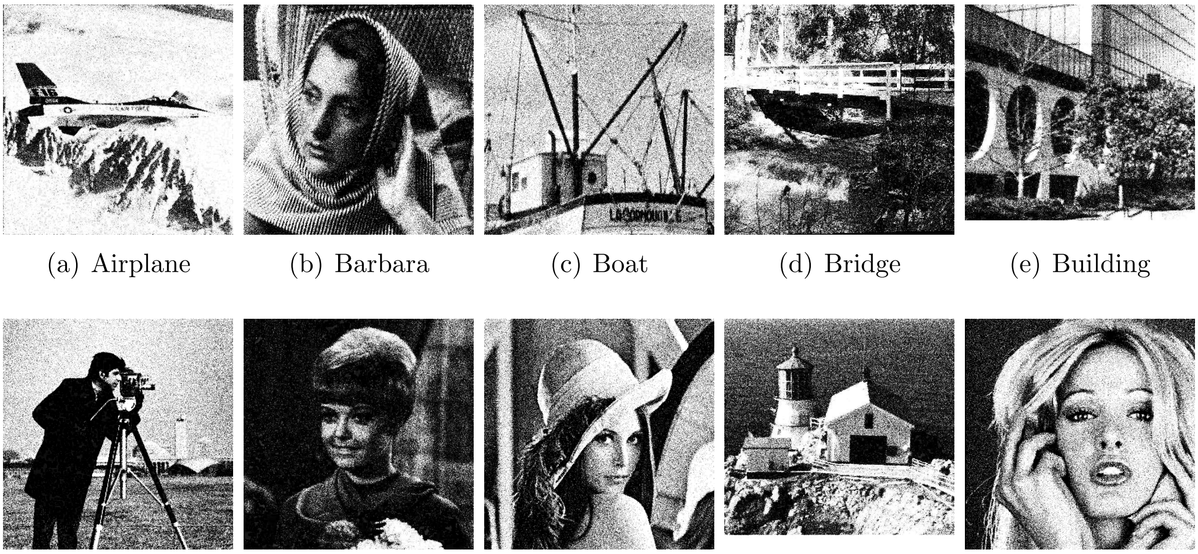
(a) Airplane (b) Barbara (c) Boat (d) Bridge (e) Building

(f) Cameraman (g) Girl (h) Lenna (i) Lighthouse (j) Woman

Figure 2. Various photographic images

The results of the experiments are shown in Figure 3, and an enlarged view of Figure 3(h) is shown in Figure 4. In all sand-style images, the edges of photographic images were preserved. Then, light areas contained less sand-style patterns, and dark areas contained more sand-style patterns. However, sand-style patterns were less likely to occur in the white and black areas such as the background in Figure 3(a) and the clothe in Figure 3(f). It is considered that sand-style patterns can be generated by adding Gaussian noise or impulse noise to the white and black areas and applying our method. However, since it is necessary to consider how to set the conditions for extracting the white and black areas, we want to make it a future study. In addition, such as the lower right area in Figure 3(d) and the left and lower sides in Figure 3(g), fine changes in shading could not be expressed and were expressed as blackish. It is considered that if the size of photographic images is increased and the proposed method is applied, it is possible to express fine changes in shading.

The conventional method [10], which is simpler to process in the conventional methods [10] and [11], was taken up, and the calculation time was compared with the proposed method. The calculation time of the proposed method and the conventional method [10]



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

(f) Cameraman (g) Girl (h) Lenna (i) Lighthouse (j) Woman

Figure 3. Sand-style images

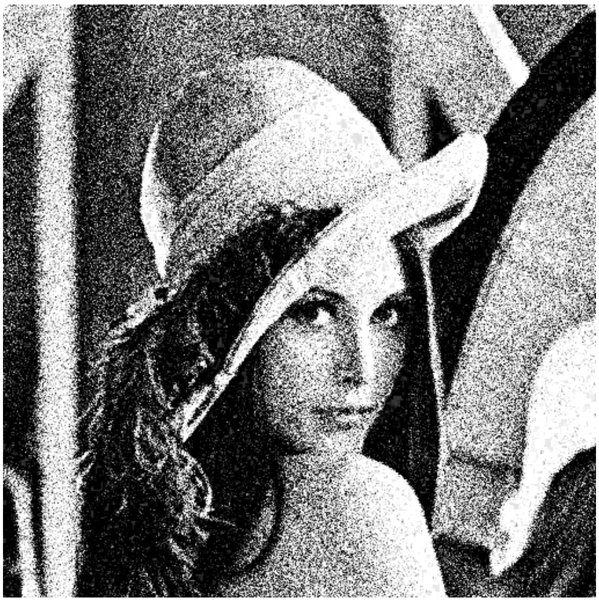


Figure 4. Enlarged view of Figure 3(h)

was 1.554 and 134.760 seconds, respectively. The proposed method could speed up the processing about 87 times faster than the conventional method [10].

4. **Conclusions.** We proposed a simple method for generating sand-style images from gray-scale photographic images by an iterative processing using hash function and inverse filter. Our method has the feature that it is expressed with less sand-style patterns in light areas and more sand-style patterns in dark areas. In addition, our method has the feature that can preserve the edges in photographic images. We demonstrated the effectiveness of our method through experiments using various photographic images. The experimental results showed that our method can realize the features.

Further study is to improve our method so that the sand-style patterns can be generated in the white and black areas. Another future challenge is to apply our method to color photographic images and movies.

Acknowledgment. This work was supported by JSPS KAKENHI Grant Number JP19K 12664.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] T. Hiraoka, A high-speed method for generating edge-preserving bubble images, *IEICE Trans. Information and Systems*, vol.E103-D, no.3, pp.724-727, 2020.

[8] R. Kazi, K. Chua, S. Zhao, R. Davis and K. Low, SandCanvas: A multi-touch art medium inspired by sand animation, *Proc. of International Conference on Human Factors in Computing Systems*, pp.1283-1292, 2011.

[9] P. Chen and S. Won, Real-time auto stylized sand art drawing, *Proc. of International Conference on Computer-Aided Design and Computer Graphics*, pp.439-440, 2013.

[10] T. Hiraoka, M. Kumano and K. Urahama, Generating sand picture-like images by using iterative enhancement bilateral filter, *The Journal of the Society for Art and Science*, vol.14, no.1, pp.20-25, 2013.

[11] T. Wu, X. Chen and L. Lu, Field coupling-based image filter for sand painting stylization, *Mathematical Problems in Engineering*, vol.2018, pp.1-18, 2018.

[12] J. M. Ortega and W. C. Rheinboldt, *Iterative Solutions of Nonlinear Equations in Several Variables*, Society for Industrial and Applied Mathematics, 1987.

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