ICIC Express Letters
Volume **18**, Number **5**, May **2024**

 ISSN 1881-803X
pp. **455–460**

GENERATION OF TRIPPY-PATTERN IMAGES USING VARIANCE-COVARIANCE MATRIX

Toru Hiraoka[1] and Seiichi Hamasaki[2]

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

[2]Nagasaki Prefectural Government
3-1, Onoue-machi, Nagasaki-shi, Nagasaki-ken 850-0058, Japan
hamasaki_s970@pref.nagasaki.lg.jp

Received June 2023; accepted September 2023

Abstract. *We propose a new non-photorealistic rendering method for automatically generating trippy-pattern (TP) images from photographic images. TP images are non-realistic images represented by overlapping colorful and curved areas on photographic images. Our method is performed by an iterative calculation using variance-covariance matrices derived from the pixel values in the window. To verify the effectiveness of our method, we visually checked TP images generated from various photographic images. Additionally, we visually confirmed how TP images generated by changing the values of the parameters in our method changed.*
Keywords: Non-photorealistic rendering, Trippy pattern, Variance-covariance matrix, Automatic generation

1. **Introduction.** Numerous studies [1, 2, 3, 4, 5, 6] have been conducted on non-photorealistic rendering (NPR), which transforms photographic images into non-photorealistic images through image processing [7, 8]. NPR is suitable for cases such as paintings and illustrations where only part of the information in a scene is to be conveyed, or where the author's intention or sensitivity is to be incorporated. Recently, NPR has been studied not only to imitate existing artistic expressions, but also to create unprecedented expressions [9, 10, 11, 12, 13, 14]. As NPRs with the unprecedented expressions, oil-film-like images [9] generated using upper and lower smoothing filters, bubble images [10] generated using absolute difference in window, checkered pattern images [11] generated using Prewitt filter with expanded window size, fingerprint-pattern images [12] generated using cosine-wave-weight smoothing filter, cell-like images [13] generated by inverse iris filter, and Islamic-pattern-like images [14] generated by weight-shifted bilateral filter have been proposed. In the natural world, oil-film-like images and bubble images imitated patterns related to physical phenomena, and fingerprint-pattern images and cell-like images imitated patterns related to living things. On the other hand, in the human world, checkered pattern images and Islamic-pattern-like images incorporated human-created patterns to achieve unprecedented expressions. The studies exploring the generations of non-photorealistic images of unprecedented expressions represent a leap into uncharted artistic territory. The development of new NPRs with unprecedented expressions is one important theme of NPRs.

We propose a novel method to generate trippy-pattern (TP) images from photographic images as an NPR with an unprecedented expression. TP images are non-realistic images

DOI: 10.24507/icicel.18.05.455

represented by overlapping colorful and curved areas on photographic images, and are classified as non-photorealistic images that incorporate human-created patterns in the human world. However, TP images are non-photorealistic images with expressions completely different from checkered pattern images and Islamic-pattern-like images, and as far as the authors have investigated, non-photorealistic images with expressions like TP images do not exist. Our method is performed by an iterative calculation using variance-covariance matrices derived from the pixel values in the window. Our method can automatically generate TP images according to the change of the edges and the shading in photographic images. Through an experiment that our method was applied to various photographic images, it was visually confirmed that TP images were generated. Additionally, through an experiment of changing the values of the parameters in our method, how to change TP images was clarified.

This paper is organized as follows: the second section describes our method for generating TP images from photographic images using variance-covariance matrices, 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 is implemented in two steps: Step 1 performs a calculation using the variance-covariance matrices derived from the pixel values in the window, and Step 2 performs a process of converting the images using variance-covariance matrices. By repeating Steps 1 and 2, TP images are generated. The flow chart of our method is shown in Figure 1.

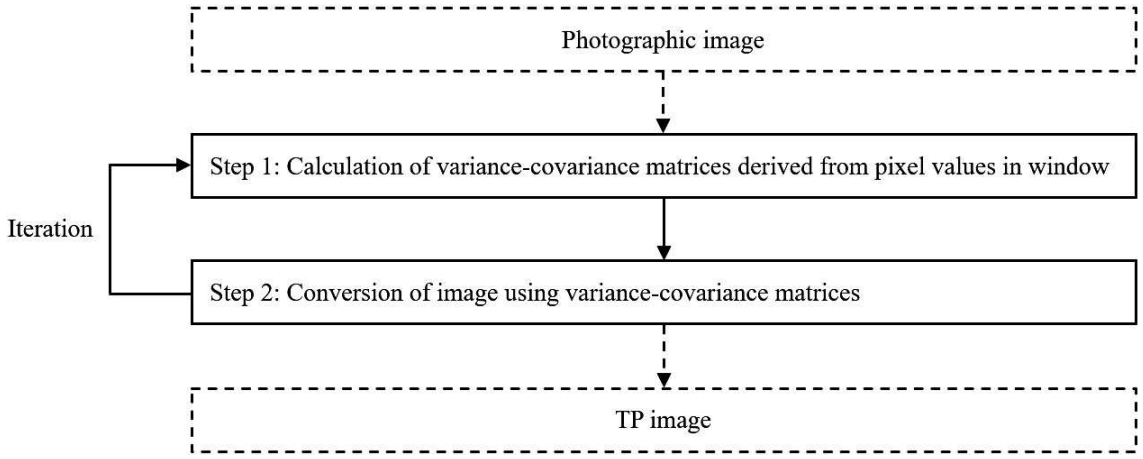


FIGURE 1. Flow chart of our method

Details of the procedure in Figure 1 are explained below.

Step 0: The input pixel values of RGB for spatial coordinates (i, j) of a photographic image are defined as $f_{R,i,j}$, $f_{G,i,j}$ and $f_{B,i,j}$. The pixel values of the image at the t-th iteration number are defined as $f^{(t)}_{R,i,j}$, $f^{(t)}_{G,i,j}$ and $f^{(t)}_{B,i,j}$, where $f^{(0)}_{R,i,j} = f_{R,i,j}$, $f^{(0)}_{G,i,j} = f_{G,i,j}$ and $f^{(0)}_{B,i,j} = f_{B,i,j}$. The pixel values $f^{(t)}_{R,i,j}$, $f^{(t)}_{G,i,j}$ and $f^{(t)}_{B,i,j}$ have value of U gradation from 0 to $U - 1$.

Step 1: The averages $a^{(t)}_{R,i,j}$, $a^{(t)}_{G,i,j}$ and $a^{(t)}_{B,i,j}$ of the pixel values in the window centered on the spatial coordinate (i, j) are calculated in each of RGB as follows:

$$a^{(t)}_{R,i,j} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} f^{(t-1)}_{R,k,l} \tag{1}$$

$$a^{(t)}_{G,i,j} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} f^{(t-1)}_{G,k,l} \tag{2}$$

$$a_{B,i,j}^{(t)} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} f_{B,k,l}^{(t-1)} \tag{3}$$

where W is the window size, and k and l are the positions in the window. The variances $v_{R,i,j}^{(t)}$, $v_{G,i,j}^{(t)}$ and $v_{B,i,j}^{(t)}$ are calculated in each of RGB as follows:

$$v_{R,i,j}^{(t)} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(f_{R,k,l}^{(t-1)} - a_{R,i,j}^{(t)} \right)^2 \tag{4}$$

$$v_{G,i,j}^{(t)} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(f_{G,k,l}^{(t-1)} - a_{G,i,j}^{(t)} \right)^2 \tag{5}$$

$$v_{B,i,j}^{(t)} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(f_{B,k,l}^{(t-1)} - a_{B,i,j}^{(t)} \right)^2 \tag{6}$$

The covariances $c_{RG,i,j}^{(t)}$, $c_{GB,i,j}^{(t)}$ and $c_{BR,i,j}^{(t)}$ between RGB are calculated as follows:

$$c_{RG,i,j}^{(t)} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(f_{R,k,l}^{(t-1)} - a_{R,i,j}^{(t)} \right) \left(f_{G,k,l}^{(t-1)} - a_{G,i,j}^{(t)} \right) \tag{7}$$

$$c_{GB,i,j}^{(t)} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(f_{G,k,l}^{(t-1)} - a_{G,i,j}^{(t)} \right) \left(f_{B,k,l}^{(t-1)} - a_{B,i,j}^{(t)} \right) \tag{8}$$

$$c_{BR,i,j}^{(t)} = \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} \left(f_{B,k,l}^{(t-1)} - a_{B,i,j}^{(t)} \right) \left(f_{R,k,l}^{(t-1)} - a_{R,i,j}^{(t)} \right) \tag{9}$$

The variance-covariance matrix $A_{i,j}^{(t)}$ are as follows:

$$A_{i,j}^{(t)} = \begin{pmatrix} v_{R,i,j}^{(t)} & c_{RG,i,j}^{(t)} & c_{BR,i,j}^{(t)} \\ c_{RG,i,j}^{(t)} & v_{G,i,j}^{(t)} & c_{GB,i,j}^{(t)} \\ c_{BR,i,j}^{(t)} & c_{GB,i,j}^{(t)} & v_{B,i,j}^{(t)} \end{pmatrix} \tag{10}$$

Step 2: The values $b_{R,i,j}^{(t)}$, $b_{G,i,j}^{(t)}$ and $b_{B,i,j}^{(t)}$ for converting the image are calculated in each of RGB as follows:

$$\begin{aligned} &\left(b_{R,i,j}^{(t)}, b_{G,i,j}^{(t)}, b_{B,i,j}^{(t)} \right) \\ &= \left(f_{R,i,j}^{(t-1)} - a_{R,i,j}^{(t)}, f_{G,i,j}^{(t-1)} - a_{G,i,j}^{(t)}, f_{B,i,j}^{(t-1)} - a_{B,i,j}^{(t)} \right) \begin{pmatrix} v_{R,i,j}^{(t)} & c_{RG,i,j}^{(t)} & c_{BR,i,j}^{(t)} \\ c_{RG,i,j}^{(t)} & v_{G,i,j}^{(t)} & c_{GB,i,j}^{(t)} \\ c_{BR,i,j}^{(t)} & c_{GB,i,j}^{(t)} & v_{B,i,j}^{(t)} \end{pmatrix}^{-1} \end{aligned} \tag{11}$$

The pixel values $f_{R,i,j}^{(t)}$, $f_{G,i,j}^{(t)}$ and $f_{B,i,j}^{(t)}$ are calculated using the values $b_{R,i,j}^{(t)}$, $b_{G,i,j}^{(t)}$ and $b_{B,i,j}^{(t)}$ as follows:

$$f_{R,i,j}^{(t)} = f_{R,i,j}^{(t-1)} + \alpha b_{R,i,j}^{(t)} \tag{12}$$

$$f_{G,i,j}^{(t)} = f_{G,i,j}^{(t-1)} + \alpha b_{G,i,j}^{(t)} \tag{13}$$

$$f_{B,i,j}^{(t)} = f_{B,i,j}^{(t-1)} + \alpha b_{B,i,j}^{(t)} \tag{14}$$

where α is a positive constant. In case $f^{(t)}_{R,i,j}$, $f^{(t)}_{G,i,j}$ and $f^{(t)}_{B,i,j}$ are less than 0, then $f^{(t)}_{R,i,j}$, $f^{(t)}_{G,i,j}$ and $f^{(t)}_{B,i,j}$ must be set to 0, respectively. In case $f^{(t)}_{R,i,j}$, $f^{(t)}_{G,i,j}$ and $f^{(t)}_{B,i,j}$ are greater than $U-1$, then $f^{(t)}_{R,i,j}$, $f^{(t)}_{G,i,j}$ and $f^{(t)}_{B,i,j}$ must be set to $U-1$, respectively.

Steps 1 and 2 are repeated T times. An image composed of the pixel values $f^{(T)}_{R,i,j}$, $f^{(T)}_{G,i,j}$ and $f^{(T)}_{B,i,j}$ is TP image.

3. **Experiments.** We conducted two experiments: the first experiment visually checked the changes of TP images when the values of the parameters in our method were changed using Parrots image shown in Figure 2, and the second experiment applied our method to four photographic images shown in Figure 3. All photographic images used in the experiments were $512 * 512$ pixels and 256 gradation.

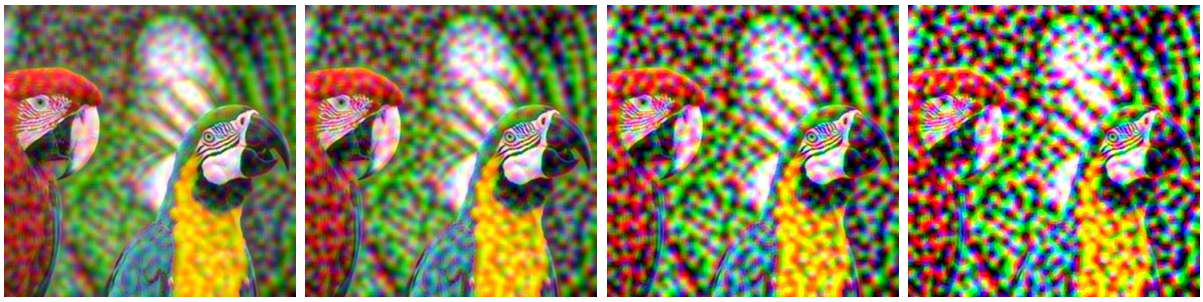
FIGURE 2. Parrots image

FIGURE 3. Various photographic images

3.1. **Experiment with changing parameters.** We visually confirmed TP images by changing the value of the iteration number T using Parrots image. The value of the iteration number T was set to 10, 20, 50 and 100. The values of the parameters W and α were set to 7 and 50, respectively. The results of the experiment are shown in Figure 4. As the value of the iteration number T became larger, TP images were clear.

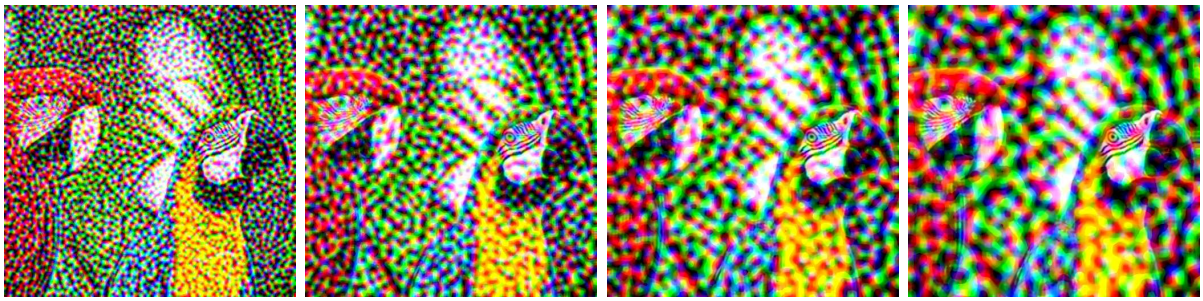
We visually confirmed TP images by changing the value of the window size W using Parrots image. The value of the window size W was set to 3, 5, 7 and 9. The values of the parameters T and α were set to 100 and 50, respectively. The results of the experiment are shown in Figure 5. As the value of the window size W became larger, trippy patterns were larger.

We visually confirmed TP images by changing the value of the parameter α using Parrots image. The value of the parameter α was set to 10, 30, 50 and 70. The values of the parameters T and W were set to 100 and 7, respectively. The results of the experiment are shown in Figure 6. As the value of the parameter α became larger, trippy patterns were clear. On the other hand, as the value of the parameter α became larger, it was harder to recall Parrots image.



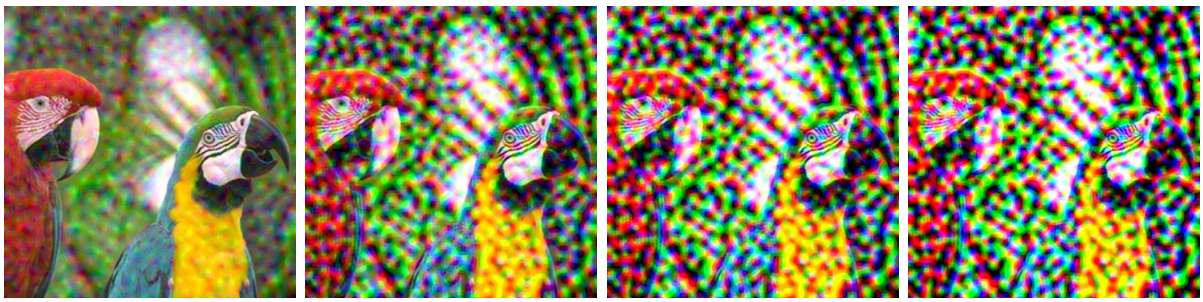
(a) $T = 10$ (b) $T = 20$ (c) $T = 50$ (d) $T = 100$

FIGURE 4. TP images in the case of the iteration number $T = 10$, 20, 50 and 100



(a) $W = 3$ (b) $W = 5$ (c) $W = 7$ (d) $W = 9$

FIGURE 5. TP images in the case of the window size $W = 3$, 5, 7 and 9



(a) $\alpha = 10$ (b) $\alpha = 30$ (c) $\alpha = 50$ (d) $\alpha = 70$

FIGURE 6. TP images in the case of the parameter $\alpha = 10$, 30, 50 and 70

3.2. **Experiment using various photographic images.** We applied our method to four photographic images shown in Figure 3. The values of the parameters T, W and α were set to 100, 7 and 50, respectively. The results of the experiment are shown in Figure 7. All TP images could be automatically generated according to the change of the edges and the shading in photographic images.

4. **Conclusions.** We proposed a new NPR method for generating TP images from photographic images. Our method was performed by an iterative calculation using variance-covariance matrices derived from the pixel values in the window. Through an experiment that our method was applied to various photographic images, it was found that our method can automatically generate TP images according to the change of the edges and the shading in photographic images. Additionally, through an experiment of changing the values of the parameters in our method, it was found how trippy patterns were changed.

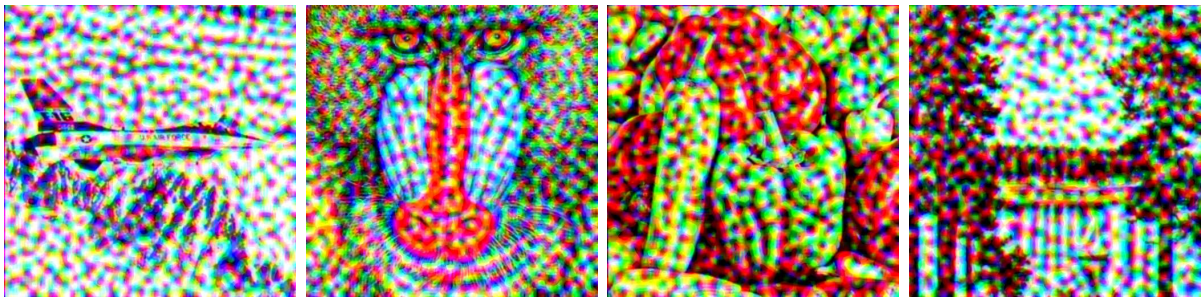


FIGURE 7. Various TP images

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

Acknowledgment. This work was supported by JSPS KAKENHI Grant Number JP23K 11727 and The Telecommunications Advancement Foundation Grant.

REFERENCES

[1] 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.

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

[3] P. L. Rosin, Y. K. Lai, D. Mould, R. Yi, I. Berger, L. Doyle, S. Lee, C. Li, Y. J. Liu, A. Semmo, A. Shamir, M. Son and H. Winnemoller, NPRportrait 1.0: A three-level benchmark for non-photorealistic rendering of portraits, *Computational Visual Media*, vol.8, no.3, pp.445-465, 2022.

[4] W. Ye, X. Zhu and Y. Liu, Multi-semantic preserving neural style transfer based on Y channel information of image, *The Visual Computer*, vol.39, no.2, pp.609-623, 2023.

[5] A. Karimov, E. Kopets, T. Shpilevaya, E. Katser, S. Leonov and D. Butusov, Comparing neural style fransfer and gradient-based algorithms in brushstroke rendering tasks, *Mathematics*, vol.11, no.10, 2255, pp.1-30, 2023.

[6] A. Ackerman, J. Auwaerter, E. Foulds, R. Page and E. Robinson, Cultural landscape visualization: The use of non-photorealistic 3D rendering as an analytical tool to convey change at statue of liberty national monument, *Journal of Cultural Heritage*, vol.62, pp.396-403, 2023.

[7] H. Chen and Z. M. Lu, Dynamic smoke detection by eliminating static targets in video, *International Journal of Innovative Computing, Information and Control*, vol.19, no.2, pp.355-364, 2023.

[8] C. Widjaya and A. Wicaksana, Liveness detection with randomized challenge-response for faceree recognition anti-spoofing, *International Journal of Innovative Computing, Information and Control*, vol.19, no.2, pp.419-430, 2023.

[9] T. Hiraoka and K. Urahama, Acceleration using upper and lower smoothing filters for generating oil-film-like images, *IEICE Transactions on Information and Systems*, vol.E102-D, no.12, pp.2642-2645, 2019.

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

[11] T. Hiraoka, Generation of checkered pattern images by iterative calculation using Prewitt filter with expanded window size, *IEICE Transactions on Information and Systems*, vol.E103-D, no.11, pp.2407-2410, 2020.

[12] T. Hiraoka, Generation of fingerprint-pattern images using cosine-wave-weight smoothing filter, *ICIC Express Letters*, vol.16, no.2, pp.153-158, 2022.

[13] T. Hiraoka, Generation of cell-like images with variable pattern size from RGB-D images, *ICIC Express Letters*, vol.16, no.7, pp.723-729, 2022.

[14] T. Hiraoka and M. Kumano, Quality improvement for generation of Islamic-pattern-like images using weight-shifted bilateral filter, *ICIC Express Letters*, vol.17, no.4, pp.457-462, 2023.