# A Method for Patterns of Cell-Like Images Based on Distance Transformation

**Toru Hiraoka**
*Department of Information Systems*
*University of Nagasaki*
*1-1-1, Manabino, Nagayo-chou, Nishisonogi-gun, Nagasaki-ken, 851-2195, Japan*
*E-mail: hiraoka@sun.ac.jp*

**Kohei Maeda**
*Department of Information Systems*
*University of Nagasaki*
*1-1-1, Manabino, Nagayo-chou, Nishisonogi-gun, Nagasaki-ken, 851-2195, Japan*
*E-mail: bs118031@sun.ac.jp*

**Abstract**

A non-photorealistic rendering method has been proposed to generate cell-like images in which cell-like patterns are represented in photographic images. Cell-like patterns are automatically generated by the change of density of photographic images. However, cell-like patterns are irregularly arranged. In this paper, we propose a method to arrange cell-like patterns along the edges of photographic images. We improves the conventional method by using Euclidean distance from the edges. We show that appealing cell-like images can be generated by our method through experiments using various photographic images.

*Keywords*: Non-photorealistic rendering, Cell, Euclidean distance, Edge, Automatic generation

## 1. Introduction

Non-photorealistic rendering is a field of computer graphics that can generate effective illustrations and attractive artistic images. Some researchers have proposed non-photorealistic rendering methods to simulate art expression techniques and to develop unprecedented artistic approaches [1,2,3,4,5,6,7]. One of the unprecedented artistic approaches is a method to generating cell-like images by an iterative calculation using inverse iris filter from photographic images [8]. Cell-like patterns are composed of cell membrane and cell nucleus. Cell-like images are overlaid with cell-like patterns in photographic images. Cell patterns are automatically generated by the change of density of photographic images. However, cell-like patterns are irregularly arranged. Therefore, a method for aligning cell-like patterns in cell-like images has been proposed [9]. The conventional method[9] is implemented by synthesizing sine and cosine waves into photographic images. However, in the conventional method [9], cell-like patterns are mainly arranged in a grid pattern, and therefore the edge preservation of cell-like images is not high. Also, the actual cells are not arranged in the grid pattern.

In this paper, we propose a method to arrange cell-like patterns along the edges of photographic images. We improve the conventional method [8] by using Euclidean distance from the edges. Cell-like images of our method can give an impression different from cell-like images of the conventional method [8,9]. By conducting experiments using various photographic images, it is visually confirmed that cell-like patterns can be generated along the edges. In addition, it is visually confirmed that the size of cell-like patterns can be changed by changing the values of the parameters in our method.

This paper is organized as follows: the second section describes our method to arrange cell-like patterns along the edges, 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. In the first step, distance-transformed images are created by calculating Euclidean distance from the edges of photographic images. In the second step, cell-like images are generated using distance-transformed images and inverse iris filter. Inverse iris filter is calculated by the procedure that restores the transformed image using iris filter [10] to the original image using inverse filter [11]. The flow chart of our method is shown in Fig. 1.

We explain details of the procedure in Fig. 1 below.

Step 0: Let input pixel values of RGB on coordinates $(i, j)$ in a color photographic image be $f_{R,i,j}$ , $f_{G,i,j}$ and $f_{B,i,j}$ ($i$ = 1, 2, 3, ⋯ , $I$; $j$ = 1, 2, 3, ⋯ , $J$). The pixel values $f_{R,i,j}$ , $f_{G,i,j}$ and $f_{B,i,j}$ have value of $U$ gradation from 0 to $U-1$.

Step 1: Gray-scale pixel values $f_{i,j}$ are calculated as follows.

$$f_{i,j} = \frac{f_{R,i,j}+f_{G,i,j}+f_{B,i,j}}{3} \qquad (1)$$

Edges are extracted from the gray-scale image using EDISON [12] that is a feature extraction tool that integrates edge detection and image segmentation. Shortest Euclidean distances $d_{1,i,j}$ from the edge pixels are calculated at each pixel.

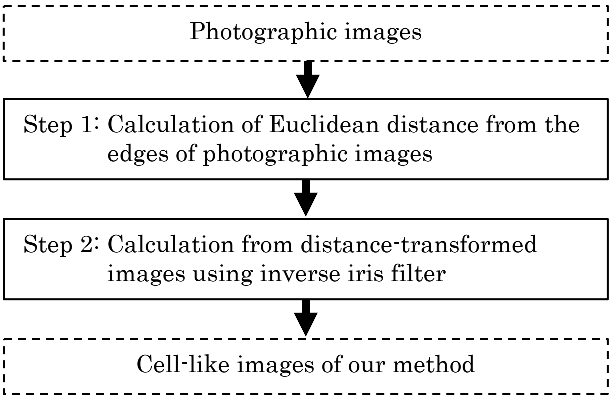

Photographic images
Step 1: Calculation of Euclidean distance from the edges of photographic images
Step 2: Calculation from distance-transformed images using inverse iris filter
Cell-like images of our method

Fig. 1. Flow chart of our method

If the Euclidean distances $d_{1,i,j}$ are within $mD + D/2 \pm 0.5$($m$ = 0, 1, 2, ⋯), the Euclidean distances $d_{2,i,j}$ are set to 0, where $D$ is a positive constant. Otherwise, the Euclidean distances $d_{2,i,j}$ are set to ∞. When the positions where the Euclidean distances $d_{2,i,j}$ are 0 are expressed on the image, it becomes lines of the interval $D$ along the edges.

Scan from the upper left to the lower right of the image, and in the case that the Euclidean distance $d_{2,i,j}$ are 0 at the target pixel $(i, j)$, the Euclidean distances $d_{2,k,l}$ in the range where the Euclidean distance from the target pixel $(i, j)$ is smaller than $D$ are updated to ∞. Where the pixels $(k, l)$ have the Euclidean distances from the target pixel $(i, j)$ smaller than $D$, and the Euclidean distance $d_{2,i,j}$ of the target pixel $(i, j)$ is not updated. When the positions where the Euclidean distances $d_{2,i,j}$ are 0 are expressed on the image, it becomes dotted lines with the spacing $D$.

Scan from the upper left to the lower right of the image, and in the case that the Euclidean distance $d_{2,i,j}$ are ∞ at the target pixel $(i, j)$, the Euclidean distance $d_{2,i,j}$ of the target pixel $(i, j)$ is updated to 0 if there is no pixel where the Euclidean distances $d_{2,k,l}$ are 0 in the range where the Euclidean distance from the target pixel $(i, j)$ is smaller than $D$. In all pixels $(i, j)$, there are pixels that the Euclidean distances $d_{2,k,l}$ are 0 within the radius $D$. Shortest Euclidean distances $d_{i,j}$ from the pixels that the Euclidean distances $d_{2,k,l}$ are 0 are calculated at each pixel. An image composed of the shortest Euclidean distances $d_{i,j}$ is called a distance-transformed image.

Step 2: Let output pixel values after processing with iris filter on $d_{i,j}$ be $IF(d_{i,j})$. Iris filter is executed with the $2r$+1 peripheral pixels $(k, l)$ in the window of size $r$. Angles $\theta_{i,j,k,l}$ between a vector $(i-k, j-l)$ from the peripheral pixels $(k, l)$ to the target pixel $(i, j)$ and a vector $\Big(\big(d_{k+2,l+2}+d_{k+2,l+1}+d_{k+2,l}+d_{k+2,l-1}+d_{k+2,l-2}\big)-\big(d_{k-2,l+2}+d_{k-2,l+1}+d_{k-2,l}+d_{k-2,l-1}+d_{k-2,l-2}\ \big)$ , $\big(d_{k+2,l+2}+d_{k+1,l+2}+d_{k,l+2}+d_{k-1,l+2}+d_{k-2,l+2}\big)-\big(d_{k+2,l-2}+d_{k+1,l-2}+d_{k,l-2}+d_{k-1,l-2}+d_{k-2,l-2}\big)\Big)$ are computed. Let convergence indices of the target pixel $(i, j)$ be

$c_{i,j}$. The convergence indices $c_{i,j}$ are calculated as follows.

$$c_{i,j} = \frac{1}{(2r+1)^2} \left| \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} \cos\theta_{i,j,k,l} \right| \qquad (2)$$

Let minimum and maximum values of $c_{i,j}$ in all pixels be $c_{min}$ and $c_{max}$, respectively. The convergence indices $c_{i,j}$ are converted to $C_{i,j}$ as follows.

$$C_{i,j} = 255 \left( \frac{c_{i,j} - c_{min}}{c_{max} - c_{min}} \right) \qquad (3)$$

The values $IF(d_{i,j})$ and $C_{i,j}$ are the same value. Pixel values $g_{R,i,j}$, $g_{G,i,j}$ and $g_{B,i,j}$ are computed by using inverse iris filter as follows.

$$\begin{aligned} g_{R,i,j} &= a\left( f_{,i,j} - C_{i,j} \right) + f_{R,i,j} \\ g_{G,i,j} &= a\left( f_{,i,j} - C_{i,j} \right) + f_{G,i,j} \\ g_{B,i,j} &= a\left( f_{,i,j} - C_{i,j} \right) + f_{B,i,j} \end{aligned} \qquad (4)$$

where $a$ is positive constant. If $g_{R,i,j}$, $g_{G,i,j}$ and $g_{B,i,j}$ are less than 0, then these values must be set to 0, respectively. If $g_{R,i,j}$, $g_{G,i,j}$ and $g_{B,i,j}$ are greater than $U - 1$, then these values must be set to $U - 1$, respectively. The image composed of the pixel values $g_{R,i,j}$, $g_{G,i,j}$ and $g_{B,i,j}$ is a cell-like image of our method.

## 3. Experiments

We conducted two experiments. In the first experiment, we visually confirmed the appearance of cell-like images generated from Lenna image shown in Fig. 2 by changing the values of the parameters. In the second experiment, used in the experiments were 512 * 512 pixels and 256 gradation. For reference, the edge extracted image and the distance-transformed image ($D$ = 15) of Lenna image are show in Fig. 4 (a) and (b), respectively.

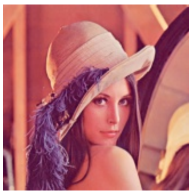


Figure 2. Lenna image.

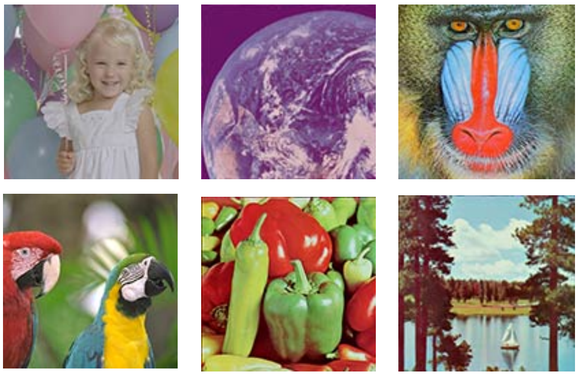
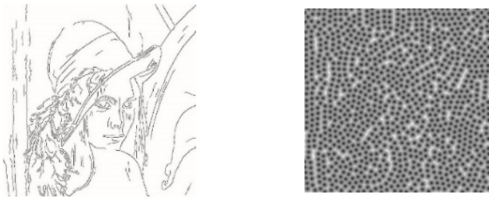


Figure 3. Various photographic images.



(a) Edge-extracted image (b) Distance-transformed image

Figure 4. Edge-extracted image and distance-transformed image.

### 3.1. Experiments with different parameter values

We visually assessed the change in appearance of cell-like images as the value of the parameter $D$ was changed using Lenna image. The value of the parameter $D$ was set to 10, 15 and 20, and the value of the parameters $r$ and $a$ were set to 3 and 0.4, respectively. Cell-like images in the case are shown in Fig. 5. The larger the value of the parameter $D$, the larger cell-like patterns were expressed.

We visually assessed the change in appearance of cell-like images as the value of the parameter $r$ was changed using Lenna image. The value of the parameter $r$ was set to 2, 3 and 4, and the value of the parameters $D$ and $a$ were set to 15 and 0.4, respectively. Cell-like images in the case are shown in Fig. 6. The larger the value of the parameter $r$, the larger the cell nuclei of cell-like patterns were expressed.

We visually assessed the change in appearance of cell-like images as the value of the parameter $a$ was changed using Lenna image. The value of the parameter

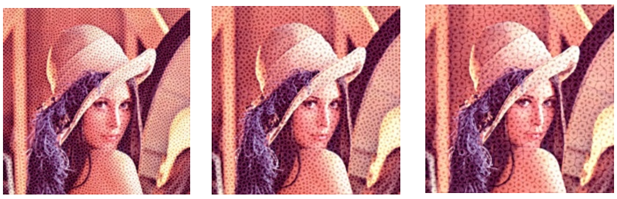

(a) $D = 10$ (b) $D = 15$ (c) $D = 20$

Figure 5. Cell-like images for $D = 10$, 15 and 20.

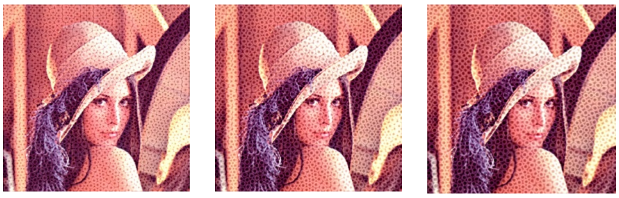

(a) $r = 2$ (b) $r = 3$ (c) $r = 4$

Figure 6. Cell-like images for $r = 2$, 3 and 4.

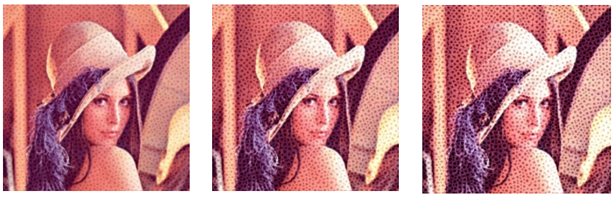

(a) $a = 0.2$ (b) $a = 0.4$ (c) $a = 0.6$

Figure 7. Cell-like images for $a = 0.2$, 0.4 and 0.6.

$a$ was set to 0.2, 0.4 and 0.6, and the value of the parameters $D$ and $r$ were set to 15 and 3, respectively. Cell-like images in the case are shown in Fig. 7. The larger the value of the parameter $a$, the more clearly cell-like patterns were emphasized.

### 3.2. Experiments with various photographic images

We applied our method to six photographic images shown in Fig. 3. Referring to the results of the experiments in the previous section, the values of the parameters $D$, $r$ and $a$ were set to 15, 3 and 0.4, respectively. Cell-like images generated from the six photographic images are show in Fig. 8. In all cell-like images, our method could express cell-like patterns along the edges of photographic images. In addition, the cell-like patterns generated by our method were expressed the cell membrane and cell nucleus more clearly than those of the conventional methods [8,9].

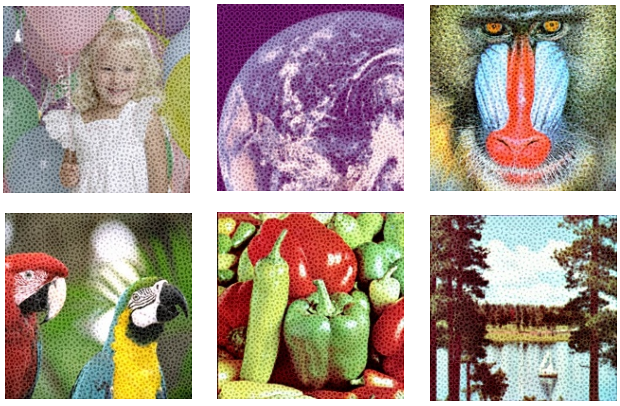


Figure 8. Cell-like images.

## 4. Conclusions

We proposed a method to arrange cell-like patterns along the edges of photographic images. We improved the conventional method [8] by using Euclidean distance from the edges of photographic images. We demonstrated the effectiveness of our method through experiments using various photographic images. The experimental results showed that our method can express cell-like patterns along the edges. In addition, the experimental results showed that the size of cell-like patterns can be changed by changing the values of the parameters in our method. However, cell-like patterns were hard to occur in the fine area and in the black and white areas.

In future work, we will try to make it possible to generate cell-like patterns in the areas where cell-like patterns do not easily occur. And, we will try to apply our method to videos and three-dimensional data.

**Acknowledgment.** This work was supported by JSPS KAKENHI Grant Number JP19-K12664.

## References

1. P. Haeberli, Paint by numbers: abstract image representations, *ACM SIGGRAPH Computer Graphics*, 24(4), 1990, pp.207-214.
2. D. D. Seligmann and S. Feiner, Automated generation of intent-based 3D illustrations, *ACM SIGGRAPH Computer Graphics*, 25(4), 1991, pp.123-132.
3. J. Lansdown and S. Schofield, Expressive rendering: a review of nonphotorealistic techniques, *IEEE Computer Graphics and Applications*, 15(3), 1995, pp.29-37.
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*, 76(13), 2017, pp.14559-14579.
5. D. Martin, G. Arroyo, A. Rodriguez and T. Isenberg, A survey of digital stippling, *Computers & Graphics*, 67, 2017, pp.24-44.

6. W. Qian, D. Xu, J. Cao, Z. Guan and Y. Pu, Aesthetic art simulation for embroidery style, *Multimedia Tools and Applications*, 78(1), 2019, pp.995-1016.
7. T. Hiraoka, Generation of edge-enhancing labyrinth images using inverse filter and two improved Laplacian filters, *ICIC Express Letters*, 14(2), 2020, pp.121-127.
8. T. Hiraoka, M. Hirota, K. Inoue and K. Urahama, Generating cell-like color images by inverse iris filter, *ICIC Express Letters*, 11(2), 2017, pp.399-404.
9. T. Hiraoka, A method for emphasizing and aligning patterns in cell-like images, *ICIC Express Letters*, 13(11), 2019, pp.1031-1037.
10. H. Kobatake and S. Hashimoto, Convergence index filter for vector fields, *IEEE Transactions on Image Processing*, 8(8), 1999, pp.1029-1038.
11. Z. Yu and K. Urahama, Iterative method for inverse nonlinear image processing, *IEICE Transactions on Fundamentals*, E97-A(2), 2014, pp.719-721.
12. The Robust Image Understanding Laboratory at Rutgers University, *Edge detection and image segmentation (EDISON) system*, http://rci.rutgers.edu/˜meer/RIUL/research/code/EDISON/doc/help.html, 2002.