ICIC Express Letters ICIC International ©2018 ISSN 1881-803X
Volume **12**, Number **1**, January **2018** pp. **23–28**

GENERATION OF CELL-LIKE COLOR ANIMATION BY INVERSE IRIS FILTER

Toru Hiraoka[1] and Masaharu Hirota[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]Department of Information Science
Okayama University of Science
1-1, Ridai-cho, Okayama-shi, Okayama-ken 700-0005, Japan
hirota@mis.ous.ac.jp

Received July 2017; accepted October 2017

Abstract. *A non-photorealistic rendering method has been proposed for generating a cell-like color image (CLCI) by an inverse iris filter. Non-photorealistic rendering is used for special visual effects in a personal computer, a portable terminal, and a television. CLCI is overlaid with cell patterns in a photographic color image. However, in some cases, the visual appearance of cell-like color animation (CLCA) generated from a video is improved more than those of CLCI. When CLCI is generated from each frame of the video and CLCA is generated from these CLCIs, flickering occurs in the generated CLCA. Such CLCA is designated as the usual CLCA. This letter presents a method for generating CLCA that suppresses flicker from the video. To validate the effectiveness of the proposed method, we compare the proposed CLCA and the usual CLCA visually and quantitatively. As a result, we show that the proposed method suppressed flicker better than the usual method.*
Keywords: Non-photorealistic rendering, Animation, Cell, Inverse filter, Iris filter

1. **Introduction.** A non-photorealistic rendering method has been proposed for generating a cell-like color image (CLCI) from a photographic color image by using an inverse iris filter [1]. The inverse iris filter is an inverse filter [2] that restores an image converted by some image processing procedures to an original image. The inverse iris filter performs calculations by using a procedure that restores an image converted by an iris filter [3] to the original image. CLCI includes cell patterns that are formed from a cell membrane and a cell nucleus, and is characterized to easily recognize a contour region because black cell patterns are arranged along the contour of the photographic color image. Although we use CLCI as special visual effects in a personal computer, a portable terminal, and a television, in some cases, the visual appearances of the cell-like color animation (CLCA) generated from a video are improved more than those of CLCI. When CLCI is generated from each frame of the video and CLCA is generated from these CLCIs, flickering occurs in the generated CLCA. Such CLCA is designated for this study as the usual CLCA. Flickering is known to occur similarly in animations generated by another non-photorealistic rendering method [4, 5, 6].

This letter proposes a CLCA generation method that suppresses flicker from the video. The proposed method suppresses flicker by using previous and subsequent frames of the video. The conceptual diagram of the proposed method is shown in Figure 1. To validate the effectiveness of the proposed method, we compare the proposed CLCA and the usual CLCA visually and quantitatively. As a result, we show that the proposed method suppressed flicker better than the usual method.

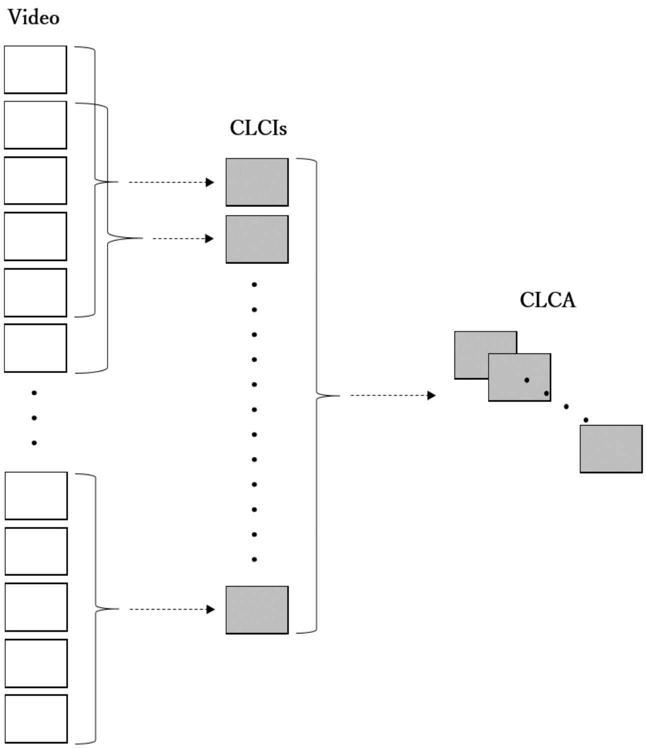


FIGURE 1. The conceptual diagram of the proposed method

The rest of this letter is organized as follows. Section 2 describes the proposed method for generating CLCA that suppresses flicker from the video. Section 3 shows the effectiveness of the proposed method through the experiments based on the visual appearance and the quantitative evaluation. Finally, Section 4 concludes this letter.

2. **Proposed Method.** We propose a method for generating CLCA by using an iterative process with an inverse iris filter from a video. The proposed method suppresses flicker by using previous and subsequent frames of the video. The procedure of the proposed method is as the following.

We define input pixel values of RGB on coordinates (i, j) in the m-th frame of the video as $f_{R,m,i,j}$, $f_{G,m,i,j}$, and $f_{B,m,i,j}$ $(i = 1, 2, \ldots, I;\ j = 1, 2, \ldots, J;\ m = 1, 2, \ldots, M)$. The pixel values $f_{R,m,i,j}$, $f_{G,m,i,j}$, and $f_{B,m,i,j}$ have values of 256 gradations from 0 to 255. The gray-scale-pixel value $f_{m,i,j}$ is calculated by using $2N + 1$ (N: positive constant) frames before and after the m-th frame as the following equation.

$$g_{m,i,j} = \frac{f_{R,m,i,j} + f_{G,m,i,j} + f_{B,m,i,j}}{3} \tag{1}$$

$$f_{m,i,j} = \frac{\sum_{n=-N}^{N} \frac{1}{1+|n|} g_{m+n,i,j}}{\sum_{n=-N}^{N} \frac{1}{1+|n|}} \tag{2}$$

We define the output pixel value after processing with an iris filter on $f_{m,i,j}$ as $IF(f_{m,i,j})$. The iris filter calculates the peripheral pixels (k, l) in the radius R pixels from the target pixel (i, j), and compute angle $\theta_{m,i,j,k,l}$ between a vector $(i-k, j-l)$ from peripheral pixels (k, l) to the target pixel (i, j) and a vector $((f_{m,k+2,l+2} + f_{m,k+2,l+1} + f_{m,k+2,l} + f_{m,k+2,l-1} + f_{m,k+2,l-2}) - (f_{m,k-2,l+2} + f_{m,k-2,l+1} + f_{m,k-2,l} + f_{m,k-2,l-1} + f_{m,k-2,l-2}), (f_{m,k+2,l+2} + f_{m,k+1,l+2} + f_{m,k,l+2} + f_{m,k-1,l+2} + f_{m,k-2,l+2}) - (f_{m,k+2,l-2} + f_{m,k+1,l-2} + f_{m,k,l-2} + f_{m,k-1,l-2} + f_{m,k-2,l-2}))$. We define the convergence index of the target pixel (i, j) as $c_{m,i,j}$. The convergence index

$c_{m,i,j}$ is calculated as,

$$c_{m,i,j} = \frac{1}{S} \left| \sum_{k=i-R}^{i+R} \sum_{l=j-R}^{j+R} \cos\theta_{m,i,j,k,l} \right| \tag{3}$$

where $\cos\theta_{m,i,j,k,l}$ is calculated as 0 if $\sqrt{(i-k)^2+(j-l)^2}$ is greater than R, and S is the number of pixels included in the radius R from the target pixel (i,j). We define the minimum and maximum values of $c_{m,i,j}$ in all pixels as $c_{\min,m}$ and $c_{\max,m}$, respectively. We convert $c_{m,i,j}$ to $C_{m,i,j}$ as the following equation.

$$C_{m,i,j} = 255\left(\frac{c_{m,i,j} - c_{\min,m}}{c_{\max,m} - c_{\min,m}}\right) \tag{4}$$

The values $IF(f_{m,i,j})$ and $C_{m,i,j}$ are the same value.

Next, we compute the pixel values $f_{R,m,i,j}^{(t)}$, $f_{G,m,i,j}^{(t)}$, and $f_{B,m,i,j}^{(t)}$ by using an inverse iris filter as

$$f_{R,m,i,j}^{(t)} = a\left(f_{m,i,j}^{(t-1)} - IF\left(f_{m,i,j}^{(t-1)}\right)\right) + f_{R,m,i,j} \tag{5}$$

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

$$f_{B,m,i,j}^{(t)} = a\left(f_{m,i,j}^{(t-1)} - IF\left(f_{m,i,j}^{(t-1)}\right)\right) + f_{B,m,i,j} \tag{7}$$

$$g_{m,i,j}^{(t-1)} = \frac{f_{R,m,i,j}^{(t-1)} + f_{G,m,i,j}^{(t-1)} + f_{B,m,i,j}^{(t-1)}}{3} \tag{8}$$

$$f_{m,i,j}^{(t-1)} = \frac{\displaystyle\sum_{n=-N}^{N} \frac{1}{1+|n|} g_{m+n,i,j}^{(t-1)}}{\displaystyle\sum_{n=-N}^{N} \frac{1}{1+|n|}} \tag{9}$$

where a is a positive constant and t is the number of iterations. We define the initial values $f_{R,m,i,j}^{(0)}$, $f_{G,m,i,j}^{(0)}$, and $f_{B,m,i,j}^{(0)}$ as $f_{R,m,i,j}$, $f_{G,m,i,j}$, and $f_{B,m,i,j}$, respectively. The pixel values $f_{R,m,i,j}^{(t)}$, $f_{G,m,i,j}^{(t)}$, and $f_{B,m,i,j}^{(t)}$ are set to 0 if their values are less than 0, and are set to 255 if their values are greater than 255.

Finally, the proposed CLCI is obtained after processing of the inverse iris filter of T times iteration. Then, the proposed CLCA is generated from these proposed CLCIs.

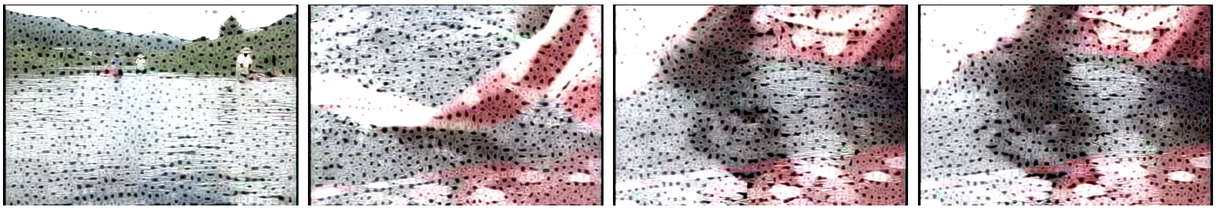
3. **Experiments.** To validate the effectiveness of the proposed method, we used the Yuzenzome video which has been developed by the video-database-working group affiliated with the technical group of PRMU [7]. The Yuzenzome video consists of 703 frames, 30 frames/second, 352 × 240 size, and 256 tones. The 126th, 127th, 200th, and 201th frames of the Yuzenzome video are portrayed in Figure 2. The 126th and 127th frames are the scene change frames.

We first visually compared the proposed CLCA and the usual CLCA. Next, we examined the visual appearances of the proposed CLCA when the value of the parameter N is varied, and quantitatively compared the proposed CLCA and the usual CLCA. Finally, we examined the optimal value of the parameter N. In the following experiments, referring to [1], the values of parameters R, a, and T were set to 3, 0.4, and 20, respectively.

We visually compared the proposed CLCA and the usual CLCA. In this case, the value of parameter N was set to 3. In the questionnaire survey, the proposed CLCA and the usual CLCA were presented to subjects, and the subjects selected one of two CLCA with

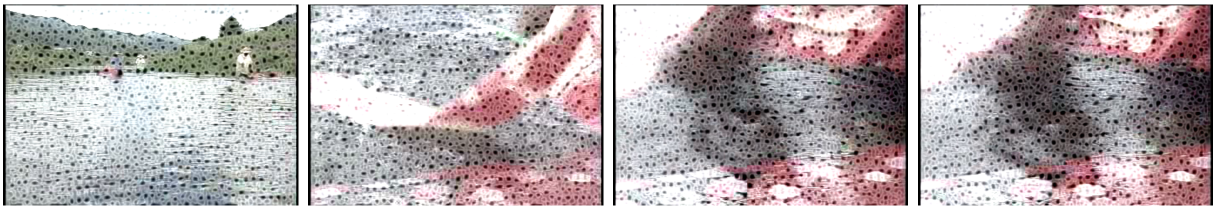
(a) 126th (b) 127th (c) 200th (d) 201th

FIGURE 2. The frames of the Yuzenzome video



(a) 126th (b) 127th (c) 200th (d) 201th

FIGURE 3. The frames of the the usual CLCA



(a) 126th (b) 127th (c) 200th (d) 201th

FIGURE 4. The frames of the the proposed CLCA

less flicker. The number of subjects was 10. As a result of the questionnaire survey, the numbers of subjects who selected the proposed CLCA and the usual CLCA were 10 and 0, respectively. Therefore, the proposed method could be suppressed flicker more than the usual method. Figure 3 and Figure 4 respectively depict the usual CLCA and the proposed CLCA of the 126, 127, 200, and 201th frames. Observing the lower right corner of these figures, the proposed CLCA had similar cell patterns between adjacent frames compared with usual CLCA.

We examined the visual appearances of the proposed CLCA quantitatively while varying the value of the parameter N. We calculated the average absolute value of the difference of pixel values between the front and rear frames of the proposed CLCA. The average (AV) and the maximum values (MV) of all frames are presented in Table 1, when the value of the parameter N was varied. As AV and MV become smaller, flickering lessens, because the difference in pixel value between adjacent frames increases greatly when the black part of cell patterns changes. As N became larger, AV and MV became smaller. The changes of AV and MV become small after $N = 4$. In other words, as N became larger, flickering lessened. MV in all cases was the scene change between 126th and 127th frames. On the other hand, AV and MV of the usual CLCA were 28.737 and 74.148, respectively (Table 2). Consequently, the proposed CLCA shows less flicker than the usual CLCA.

We calculated the average absolute value of the difference of pixel values between frames of the proposed CLCA and the original video. Table 3 presents the average (AV2) of all frames when the value of the parameter N is varied. As AV2 is small, the proposed CLCA is more similar to the original video. AV2 was the minimum at $N = 3$, and increased as

Table 1. AV and MV of the proposed CLCA when the value of the parameter N is varied

N	AV	MV
1	17.363	55.212
2	14.890	51.749
3	13.806	50.212
4	13.200	49.462
5	12.828	48.928
6	12.583	48.546
7	12.413	48.320
8	12.287	48.134
9	12.197	48.029
10	12.125	47.930

Table 2. AV and MV of the usual CLCA

AV	MV
28.737	74.148

Table 3. AV2 of the proposed CLCA when the value of the parameter N is varied

N	AV2
1	44.320
2	44.211
3	44.202
4	44.243
5	44.291
6	44.325
7	44.362
8	44.413
9	44.415
10	44.459

N became larger than 3. In other words, the proposed CLCA was most similar to the original video at $N = 3$, and loses similarity as N becomes larger than 3. Table 1 and Table 3 showed that the optimal value of parameter N was approximately 3.

4. **Conclusions.** We proposed a method for generating CLCA that suppresses flicker from a video. To validate the effectiveness of the proposed method, we compared the proposed CLCA and the usual CLCA visually quantitatively. As a result, we showed that the proposed method suppressed flicker better than the usual method. A future task for this research is to apply the proposed method to other videos.

REFERENCES

[1] 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.

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

[3] H. Kobatake and S. Hashimoto, Convergence index filter for vector fields, *IEEE Trans. Image Processing*, vol.8, no.8, pp.1029-1038, 1999.

[4] B. J. Meier, Painterly rendering for animation, *Proc. of ACM SIGGRAPH*, pp.477-484, 1996.

[5] J. Hays and I. Essa, Image and video-based painterly animation, *Proc. of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*, pp.113-120, 2004.
[6] C. Lu, Y. Xiao and C.-K. Tang, Real-time video stylization using object flows, *IEEE Trans. Visualization and Computer Graphics*, 2017.
[7] The Video-Database-Working Group of PRMU, *Video Database for Evaluating Video Processing*, http://www.nii.ac.jp/dsc/idr/video/VDB/documentary.html, Yuzenzome, 2002.