SUGISAKA MASANORI

Journal of Advances in Artificial Life Robotics
Vol. 2(1); June (2021), pp. 17–21
ON LINE ISSN 2435-8061; ISSN-L 2435-8061
https://alife-robotics.org/jallr.html

Research Article

Extension of Striped Image by Inverse Line Convergence Index Filter to Video

Toru Hiraoka[1], Ryosuke Takaki[2]
[1]*Department of Information Systems, University of Nagasaki, 1-1-1, Manabino Nagayo-chou, Nishisonogi-gun, Nagasaki-ken, 815—2195, Japan*
[2]*Division of Computer Science, Graduate School of Regional Design and Creation, University of Nagasaki, 1-1-1, Manabino, Nagayo-chou, Nishisonogi-gun, Nagasaki-ken, 815-2195, Japan*

ARTICLE INFO

Article History

Received 06 November 2020
Accepted 10 May 2021

Keywords

Non-photorealistic rendering
Video
Striped pattern
Inverse line convergence index filter

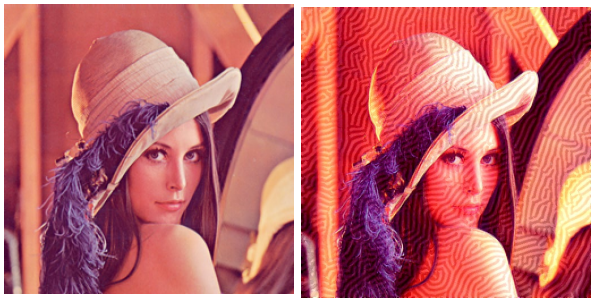
ABSTRACT

A non-photorealistic rendering method has been proposed for generating a striped image which is overlaid striped patterns in a photograph. The conventional method generates the striped image by an iterative process using an inverse line convergence index filter from the photograph. In this paper, we propose a method to extend the method of generating the striped image so that it can be applied to a video. In the proposed method, it is possible to suppress flicker due to the striped patterns. To verify the effectiveness of the proposed method, an experiment was conducted to visually and quantitatively evaluate the degree of flicker using Yuzenzome video. As a result of the experiment, it was found that the proposed method can suppress flicker.

1. Introduction

Various non-photorealistic rendering (NPR) methods for a photograph have been proposed [1],[2],[3],[4],[5],[6]. These methods are used in a wide range of applications, for examples, applications embedded in a personal computer and a portable terminal. Applying these methods to a video is to improve the visual appearance. A study on NPR of the video has also been conducted[7], but the conventional method [7] was been known that flickering occurs in the NPR video. Therefore, a method for suppressing flicker for a checkered pattern has also been proposed [8].

In this paper, we focus on the NPR method for generating a striped image [9] from the photograph. We also apply the conventional method [9] to the video. The striped image is overlaid the striped patterns in the photograph as shown in Fig. 1. The conventional method[9] generates the striped image by an iterative process using an inverse line convergence index filter, and is characterized by the ability to automatically generate the striped patterns in accordance with the shading and the edge of the photograph. When a striped video is generated by converting each frame of the video by the conventional method[9], flickering occurs in the generated



(a) Lena image (b) Striped image
Fig. 1. Lena image and striped image

Corresponding author's E-mail: hiraoka@sun.ac.jp. mc220002@sun.ac.jp

striped video. Hereinafter, the striped video generated by applying the conventional method [9] to each frame of the video is referred to as the conventional striped video. Therefore, we suppress flicker by using the forward and backward frames of the video. The conventional method [8] also uses the forward and backward frames. In the conventional method [8], NPR images are generated for each frame, and new NPR images are generated from the NPR images of the forward and backward frames. On the other hand, the proposed method directly generates NPR images using the forward and backward frames. The effectiveness of the proposed method is investigated experimentally. By visually and quantitatively comparing the proposed striped video to the conventional striped video, we show that the proposed method can suppress flicker better than the conventional method.

The rest of this paper is organized as follows. Section 2 describes the proposed method for generating the striped video. Section 3 shows experimental results, and reveals the effectiveness of the proposed method. Finally, Section 4 concludes this paper.

2. Method

Let input pixel values of RGB on coordinates (i, j) in o-th frame of the video be $f_{R,i,j,o}$, $f_{G,i,j,o}$ and $f_{B,i,j,o}$ $(i = 1,2,\dots,I; j = 1,2,\dots,J; o = 1,2,\dots,O)$, and let output pixel values after processing with a line convergence index filter [10] be $LF(f_{R,i,j,o})$, $LF(f_{G,i,j,o})$ and $LF(f_{B,i,j,o})$.

The line convergence index filter is executed in the following manner. Consider the straight line $l_{m,o}(m = 1,2,\dots,M)$ inclined $\theta_{m,o}(= 0, \pi/M, 2\pi/M, \dots, (M-1)\pi/M)$ degrees from the x-direction of the target pixel (i, j). Let the length of the straight line $l_{m,o}$ in each direction around the target pixel (i, j) be W_1 pixels. Consider the inclined rectangle which has the center at (i, j) and the length of sides are W_1, W_2. Let N be the number of pixels (k, l) inside the rectangle (Conceptual diagram for W_1, W_2 and N are shown in Fig. 2). Compute a cosine of the angle between a vector perpendicular to a straight line $l_{m,o}$ from a neighboring pixel (k, l) and a vector $((s_{k+2,l+2,o} + s_{k+2,l+1,o} + s_{k+2,l,o} + s_{k+2,l-1,o} + s_{k+2,l-2,o}) - (s_{k-2,l+2,o} + s_{k-2,l+1,o} + s_{k-2,l,o} + s_{k-2,l-1,o} + s_{k-2,l-2,o}), (s_{k+2,l+2,o} + s_{k+1,l+2,o} + s_{k,l+2,o} + s_{k-1,l+2,o} + s_{k-2,l+2,o}) - (s_{k+2,l-2,o} + s_{k+1,l-2,o} + s_{k,l-2,o} + s_{k-1,l-2,o} + s_{k-2,l-2,o}))$ calculated from the density variation. Let the cosine of the angle be $c_{m,n,o}(n = 1,2,\dots,N)$. The term $s_{i,j,o}$ is calculated as the following equation.

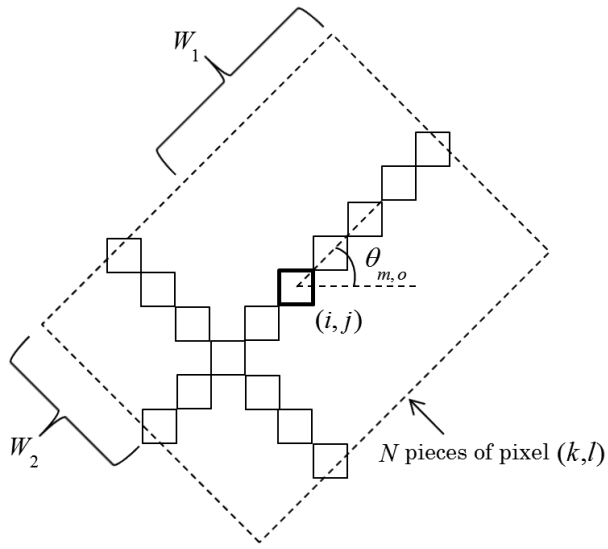


Fig. 2. Conceptual diagram for W_1, W_2 and N

$$d_{i,j,o} = \frac{f_{R,i,j,o} + f_{G,i,j,o} + f_{B,i,j,o}}{3}. \quad (1)$$

$$s_{i,j,o} = \frac{\sum_{p=-P}^{P} \frac{1}{1+|p|} d_{i,j,o+p}}{\sum_{p=-P}^{P} \frac{1}{1+|p|}}. \quad (2)$$

Where P is a positive constant. Next, compute $C_{m,o}$ that is the average of the absolute value of $c_{m,n,o}$ of N neighboring pixels (k, l) in each straight line $l_{m,o}$ as the following equation.

$$C_{m,o} = \frac{1}{N} \sum_{n=1}^{N} |c_{m,n,o}|. \quad (3)$$

With the maximum value of $C_{m,o}$ in all straight lines $l_{m,o}$ in each pixel (i, j), denote as $g_{i,j,o}$. With the minimum and maximum values of $g_{i,j}$ in all pixels, denoted as $g_{min,o}$ and $g_{max,o}$, respectively. Transform $g_{i,j,o}$ to $h_{i,j,o}$ and thus to the pixel values had from 0 to 255 by the following equation.

$$h_{i,j,o} = 255 \frac{g_{i,j,o} - g_{min,o}}{g_{max,o} - g_{min,o}}. \quad (4)$$

$LF(f_{R,i,j,o})$, $LF(f_{R,i,j,o})$ and $LF(f_{R,i,j,o})$ are the same value as $h_{i,j,o}$.

Compute the pixel value $f_{R,i,j,o}^{(t)}$, $f_{G,i,j,o}^{(t)}$ and $f_{B,i,j,o}^{(t)}$ by using the inverse line convergence index filter as

$$f_{R,i,j,o}^{(t)} = a(f_{R,i,j,o}^{(t-1)} - LF(f_{R,i,j,o}^{(t-1)})) + f_{R,i,j,o}. \quad (5)$$

$$f_{G,i,j,o}^{(t)} = a(f_{G,i,j,o}^{(t-1)} - LF(f_{G,i,j,o}^{(t-1)})) + f_{G,i,j,o}. \quad (6)$$

$$f_{B,i,j,o}^{(t)} = a(f_{B,i,j,o}^{(t-1)} - LF(f_{B,i,j,o}^{(t-1)})) + f_{B,i,j,o}. \quad (7)$$

Where a is a positive constant and t is the number of iterations. Let the initial value $f_{R,i,j,o}^{(0)}$, $f_{G,i,j,o}^{(0)}$ and $f_{B,i,j,o}^{(0)}$ be $f_{R,i,j,o}$, $f_{G,i,j,o}$ and $f_{B,i,j,o}$, respectively. $f_{R,i,j,o}^{(t)}$, $f_{G,i,j,o}^{(t)}$ and $f_{B,i,j,o}^{(t)}$ set to 0 if their values are less than 0, and set to 255 if their values are greater than 255.

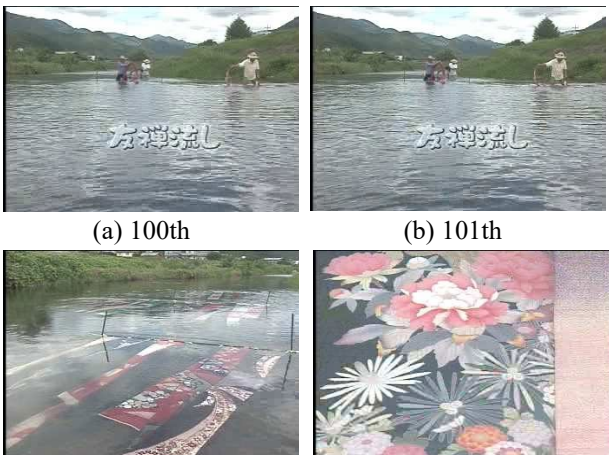
Finally, the N striped images are obtained after processing of the inverse line convergence index filter of T generated from these striped images.

3. Experiments

We applied the proposed method to the Yuzenzome video [11] which consists of 703 frames, 30 frames / second, $352 * 240$ pixels and 256 tone. The 100, 101, 387 and 388 th frames of the Yuzenzome video are shown in Fig. 3. The 387 and 388th ftames are the scene change frames. In the following experiments, referring to the literature 9), we set the values of the parameters M, W_1, W_2, a and T to 8, 4, 4, 0.4 and 30, respectively. The computing environment for all experiments was a Windows 7 Professional operating system on a computer with a 3.40 GHz CPU and a 8.00 GB of memory. The programming language used in all experiments was VC++.

3.1 Visual experiment

We visually compared the proposed striped video to the conventional striped video. The 100, 101, 387 and 388th frames of the conventional striped video and the proposed striped video are shown in Fig. 4 and Fig. 5, respectively. The larger the value of the parameter P, the less flicker, while the quality of the proposed striped video is different from the original video. When the value of the parameter P was around 3, the difference between the proposed striped video and Yuzenzome video could not be visually recognized, so we set the value of the parameter P to 3. Observing these videos, the proposed striped video were suppressed flicker more than the conventional striped video. Observing Fig. 4 and Fig. 5, the proposed striped video had less changes in the striped patterns than the conventional striped video.



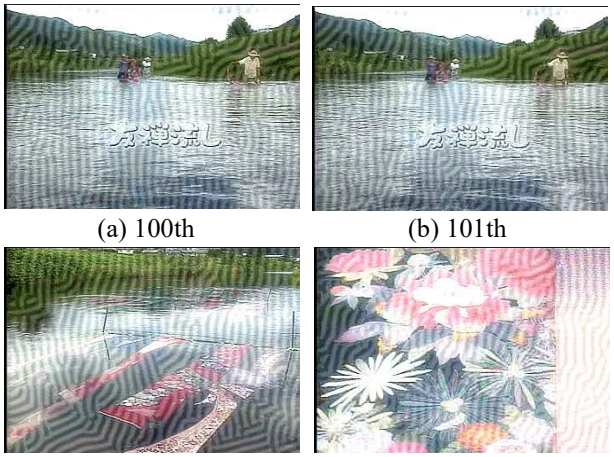
(a) 100th (b) 101th

(c) 387th (d) 388th

Fig. 3. The frames of the Yuzenzome video

3.2 Quantitative experiment

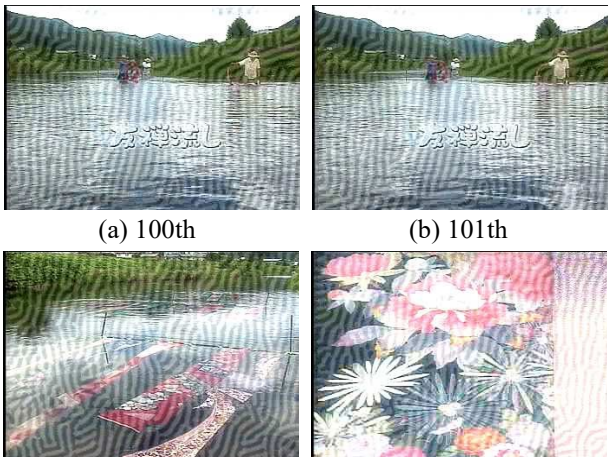
We quantitatively compared the proposed striped video to the conventional striped video. We calculated the



(a) 100th (b) 101th

(c) 387th (d) 388th

Fig. 4. The frames of the conventional striped video



(a) 100th (b) 101th

(c) 387th (d) 388th

Fig. 5. The frames of the proposed video

Table 1. The average of the conventional striped video

Average
23.856

Table 2. The average of the proposed striped video when the value of the parameter P is changed

P	Average
1	20.403
2	20.264
3	20.190
4	20.151
5	20.118
6	20.100
7	20.087
8	20.075
9	20.067
10	20.062

average of the absolute difference value of pixel values between front and rear frames of these videos. As the average become smaller, flickering is less. The average of the conventional striped video is shown in Table 1. On the other hand, the average of the proposed striped video is shown in Table 2, when we changed the values of parameter P from 1 to 10. Observing Table 2, as the values of parameter P became bigger, the average became small. Observing Table 1 and Table 2, the average of the proposed striped video were smaller than the average of the conventional striped video. Thus, the proposed method can suppress flicker better than the conventional method.

3.3 Experiment of calculation time

We compared the calculation time to generate the proposed striped video to the calculation time to generate the conventional striped video. The calculation time to generate the conventional striped video is shown in Table 3. On the other hand, the calculation time to generate the proposed striped video is shown in Table 4, when we changed the values of parameter P from 1 to 10. The calculation time in Table 3 and Table 4 is one iteration calculation [second]. Observing Table 4, the calculation time became big as the values of parameter P became bigger, but there are no big differences in these calculation times. Observing Table 3 and Table 4, there was no big difference in the calculation time for Table 3. The calculation time to generate the conventional striped video [second]

Table 3. The calculation time to generate the conventional striped video [second]

Average
4.377

Table 4. The calculation time to generate the proposed striped video when the value of the parameter P is changed [second]

P	Average
1	4.381
2	4.390
3	4.394
4	4.399
5	4.401
6	4.405
7	4.410
8	4.413
9	4.420
10	4.426

generating the proposed striped video and the conventional striped video. Thus, compared with the case that the conventional method[9] was applied to each frame of the video, the proposed method was an effective method on the calculation time. However, considering real-time use on portable terminals, it is necessary to speed up the proposed method.

4. Conclusion

We proposed a method for generating a striped video that has characteristic with less flicker from a video. The effectiveness of the proposed method was investigated experimentally. As a result of the experiments, the proposed striped video had less flicker than the conventional striped video. And, compared with the case that the conventional method[9] was applied to each frame of the video, the proposed method was an effective method on the calculation time.

A subject for future study is to speed up the proposed method. Another future study is to apply the proposed method to other videos.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP19K12664.

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. B. J. Meier, Painterly Rendering for Animation, *Proceedings of ACM SIGGRAPH 1996*, 1996, pp. 477-484.
8. T. Hiraoka, Generating Checkered Pattern Animations Using Inverse Iris Filter, *Journal of the Institute of Industrial Applications Engineers*, 6(1), 2018, pp. 17-20.
9. T. Hiraoka and K. Urahama, Generating Striped Color Images by Inverse Line Convergence Index Filter, *IIEEJ Transactions on Image Electronics and Visual Computing*. **2**(2), 2014, pp. 190-194.
10. H. Kobatake and S. Hashimoto, Convergence Index Filter for Vector Fields, *IEEE Transactions Image Processing*. 8, 1999.
11. The Video-Database-Working Group of PRMU, *Video Database for Evaluating Video Processing*, http://www.nii.ac.jp/dsc/idr/video/VDB/documentary.html,Yuzenzo-me, 2002.

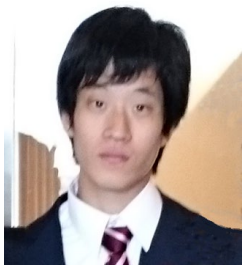
==========

Authors Introduction

Dr. Toru Hiraoka

He graduated Doctor course at Design in Kyushu Institute of Design. He is a Professor in the Faculty of Information Systems in University of Nagasaki. His current research interests are non-photorealistic rendering and disaster prevention.

Mr. Ryosuke Takaki



He is a student in the Division of Computer Science, Graduate School of Regional Design and Creation in University of Nagasaki. His current research interests are non-photorealistic rendering and image processing.

==========