ICIC Express Letters
Volume **17**, Number **12**, December **2023**

ICIC International ©2023 ISSN 1881-803X
pp. 1347–1353

GENERATION OF CHECKERED-PATTERN VIDEOS FROM RGB-D VIDEOS

Toru Hiraoka[1] and Kanya Goto[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]Movenext Corporation
Ushioda-building 5F, 2-30-8, Minamiotsuka, Toshima-ku, Tokyo 170-0005, Japan
kanya.goto@mvnxt.co.jp

Received February 2023; accepted April 2023

Abstract. *A non-photorealistic rendering method to automatically generate checkered-pattern images from photographic images using Prewitt filter with an expanded window size has been proposed. Extensions to the conventional method have also been proposed to generate checkered-pattern images from RGB-D images. In this paper, extending the conventional methods, a method is proposed to generate checkered-pattern videos from RGB-D videos. The proposed method uses the forward and backward frames of the depth and RGB in RGB-D videos separately. Flickering is a problem in non-photorealistic rendering videos, but the proposed method can suppress flicker. Through experiments using an RGB-D video taken by the authors, the flicker of checkered-pattern videos generated by the proposed method was evaluated visually and quantitatively.*

Keywords: Non-photorealistic rendering, Checkered pattern, RGB-D video, Prewitt filter, Suppression of flicker

1. **Introduction.** One of the typical purposes of non-photorealistic rendering (NPR) method [1, 2, 3, 4, 5, 6, 7, 8] is to simulate drawing techniques such as oil paintings and watercolors on a computer. There are two advantages to simulating the drawing techniques on the computer: even people who are not good at drawing can easily draw, and drawings that are difficult by hand can be done easily and in a short time. The drawings that are difficult by hand include maze-like images [9], moire-like images [10], and checkered-pattern images [11]. Maze-like images were generated on the basis of halftone images calculated using minimum spanning trees, moire-like images were generated by emphasizing the staircasing effect [12, 13] of bilateral filter [14, 15], and checkered-pattern images were generated by iterative calculation using Prewitt filter with an expanded window size.

This paper focuses on checkered-pattern images that are difficult to draw by hand. Checkered-pattern images are expressed by superimposing checkered patterns like a checkered flag on photographic images. Checkered patterns are automatically generated, are changed the shade of checkered patterns, and are curved according to changes in edges and shade of photographic images. The conventional method [11] uses photographic images, and a method of extending it to RGB-D images has also been proposed [16]. By using RGB-D images instead of photographic images, it has become possible to change the size of checkered patterns according to the depth.

This paper proposes an NPR method to automatically generate checkered-pattern videos from RGB-D videos. Flickering is generally a problem in NPR videos [17]. When

DOI: 10.24507/icicel.17.12.1347

the conventional method [16] is applied to each frame of RGB-D videos to generate checkered-pattern videos, flicker occurs in the same way. Therefore, when calculating each frame of checkered-pattern videos, flicker is suppressed by using the forward and backward frames of RGB-D videos. A method has been proposed to suppress flicker by using the forward and backward frames of RGB videos [18, 19]. Extending the concept of the conventional method [18, 19], the proposed method uses the forward and backward frames of the depth and RGB in RGB-D videos separately. This paper can contribute not only to the field of computer graphics related to NPR but also to the field of image processing related to filtering. To confirm that the proposed method can generate checkered-pattern videos with suppressed flicker, experiments were conducted using RGB-D videos taken by the authors.

This paper is organized as follows: the second section describes the proposed method for generating checkered-pattern videos from RGB-D videos, the third section shows experimental results and reveals the effectiveness of the proposed method, and the conclusion of this paper is given in the fourth section.

2. **Proposed Method.** The proposed method is implemented in three steps: Step 1 calculates the window sizes, which determine the size of checkered patterns according to the depth, using the forward and backward frames of the depth, Step 2 calculates the gradients of the gray-scale pixel values obtained from the RGB pixel values of the forward and backward frames and the window sizes using Prewitt filter with the expanded window size, and Step 3 converts the RGB videos using the gradients. By repeating Steps 2 and 3, checkered-pattern videos of the proposed method are generated. A flow chart of the proposed method is shown in Figure 1.

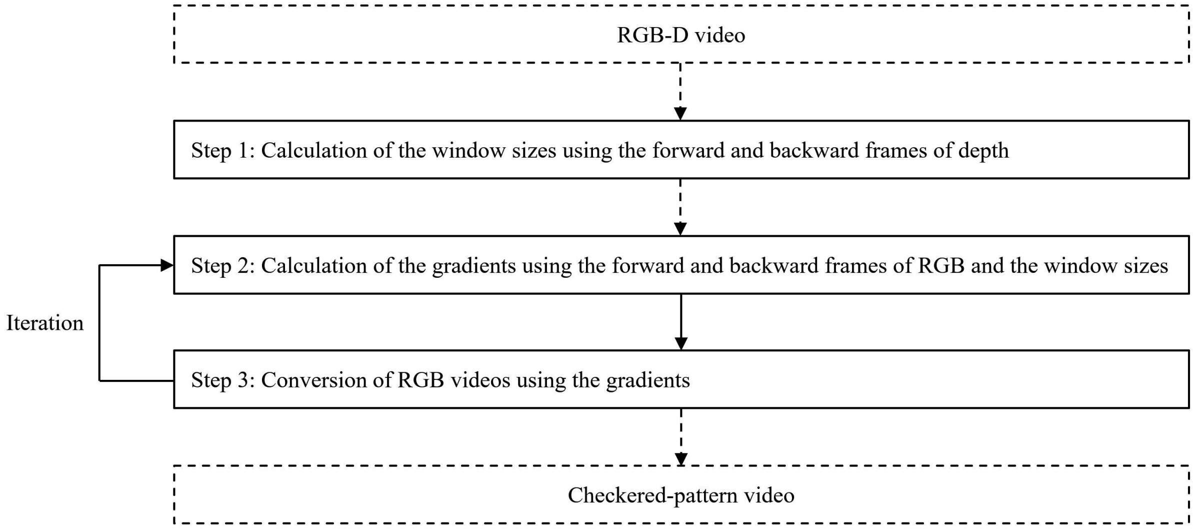

RGB-D video
Step 1: Calculation of the window sizes using the forward and backward frames of depth
Step 2: Calculation of the gradients using the forward and backward frames of RGB and the window sizes
Iteration
Step 3: Conversion of RGB videos using the gradients
Checkered-pattern video

FIGURE 1. Flow chart of the proposed method

Details of the procedure in Figure 1 are explained below.

Step 0: The input pixel values (R, G, B) and the depths for spatial coordinates (i, j) in the kth frame of an RGB-D video are defined as $f_{R,i,j,k}$, $f_{G,i,j,k}$, $f_{B,i,j,k}$, and $f_{D,i,j,k}$, respectively, where $i = 1, 2, \dots, I$, $j = 1, 2, \dots, J$, and $k = 1, 2, \dots, K$. The pixel values $f_{R,i,j,k}$, $f_{G,i,j,k}$, $f_{B,i,j,k}$, and $f_{D,i,j,k}$ have value of U gradation from 0 to $U - 1$. The depths $f_{D,i,j,k}$ are linearly transformed so that the minimum distance becomes 0 and the maximum distance becomes $U - 1$. The forward and backward O frames are used.

Step 1: At each pixel, the window sizes $W_{i,j,k}$ that determine the size of checkered patterns according to the depth are calculated using the forward and backward frames of the depth in the following equations:

$$\bar{f}_{D,i,j,k} = \frac{\sum_{n=k-O}^{k+O} f_{D,i,j,n}}{2O+1} \tag{1}$$

$$W_{i,j,k} = W_{\min} + \frac{(W_{\max} - W_{\min})\left(\bar{f}_{D,i,j,k} - \bar{f}_{D,k,\min}\right)}{\bar{f}_{D,k,\max} - \bar{f}_{D,k,\min}} \tag{2}$$

where n is the position in forward and backward frames, $\bar{f}_{D,k,\min}$ and $\bar{f}_{D,k,\max}$ are respectively the minimum and maximum values in the average depths $\bar{f}_{D,i,j,k}$, and $W_{\min}$ and $W_{\max}$ are respectively the minimum and maximum window sizes set by the user. The larger the values of the depths $f_{D,i,j,k}$, the smaller the size of checkered patterns.

Step 2: The pixel values of the image at the tth iteration number are defined as $f_{R,i,j,k}^{(t)}$, $f_{G,i,j,k}^{(t)}$, and $f_{B,i,j,k}^{(t)}$, where $f_{R,i,j,k}^{(0)} = f_{R,i,j,k}$, $f_{G,i,j,k}^{(0)} = f_{G,i,j,k}$, and $f_{B,i,j,k}^{(0)} = f_{B,i,j,k}$. The gray-scale pixel values $f_{i,j,k}^{(t)}$ are calculated in the following equation:

$$f_{i,j,k}^{(t)} = \frac{\sum_{n=k-O}^{k+O}\left(f_{R,i,j,n}^{(t-1)} + f_{G,i,j,n}^{(t-1)} + f_{B,i,j,n}^{(t-1)}\right)}{6O+3} \tag{3}$$

The gradients of the pixel values $g_{x,i,j,k}^{(t)}$ and $g_{y,i,j,k}^{(t)}$ are calculated using Prewitt filter with the expanded window in the following equations:

$$g_{x',i,j,k}^{(t)} = \sum_{m=j-W_{i,j,k}}^{j+W_{i,j,k}} \left(f_{i-W_{i,j,k},m,k}^{(t)} - f_{i+W_{i,j,k},m,k}^{(t)}\right) \tag{4}$$

$$g_{y',i,j,k}^{(t)} = \sum_{l=i-W_{i,j,k}}^{i+W_{i,j,k}} \left(f_{l,j-W_{i,j,k},k}^{(t)} - f_{l,j+W_{i,j,k},k}^{(t)}\right) \tag{5}$$

$$g_{i,j,k}^{(t)} = \sqrt{{g_{x',i,j,k}^{(t)}}^2 + {g_{y',i,j,k}^{(t)}}^2} \tag{6}$$

$$g_{x,i,j,k}^{(t)} = \frac{g_{x',i,j,k}^{(t)}}{g_{i,j,k}^{(t)}} \tag{7}$$

$$g_{y,i,j,k}^{(t)} = \frac{g_{y',i,j,k}^{(t)}}{g_{i,j,k}^{(t)}} \tag{8}$$

where l and m are the positions in the window.

Step 3: The output pixel values $f_{R,i,j,k}^{(t)}$, $f_{G,i,j,k}^{(t)}$, and $f_{B,i,j,k}^{(t)}$ are calculated using the gradients of the pixel values $g_{x,i,j,k}^{(t)}$ and $g_{y,i,j,k}^{(t)}$ in the following equations:

$$f_{R,i,j,k}^{(t)} = \begin{cases} f_{R,i,j,k} + a g_{x,i,j,k}^{(t)} & (t\%2 = 0) \\ f_{R,i,j,k} + a g_{y,i,j,k}^{(t)} & (t\%2 = 1) \end{cases} \tag{9}$$

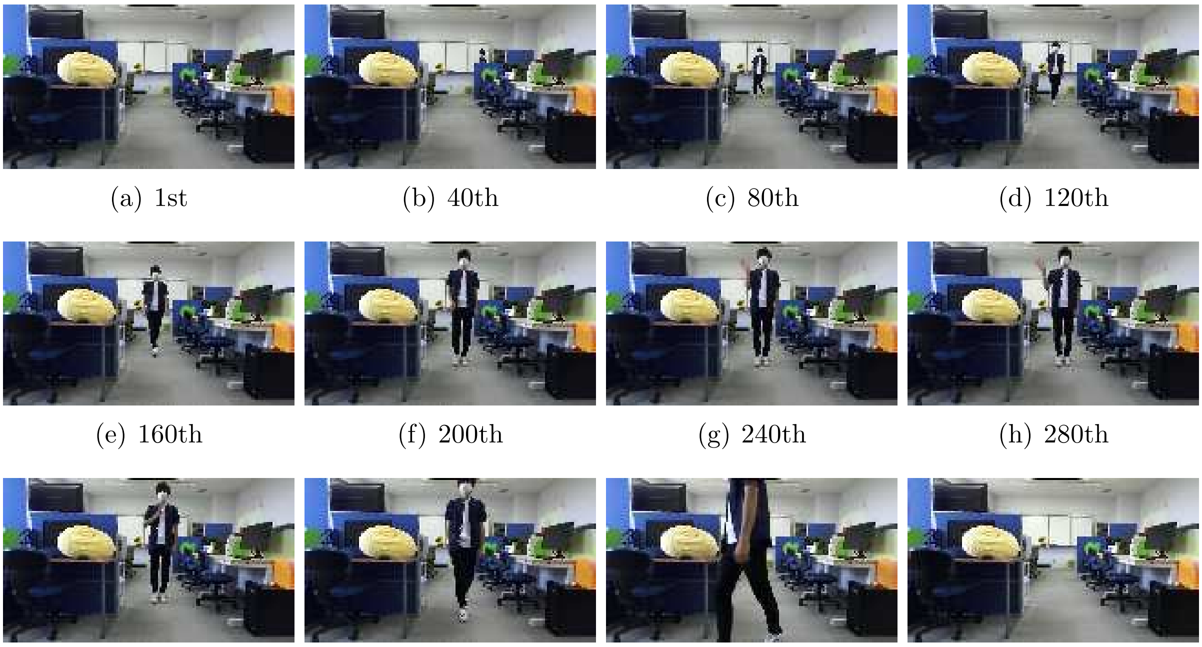
$$f_{G,i,j,k}^{(t)} = \begin{cases} f_{G,i,j,k} + a g_{x,i,j,k}^{(t)} & (t\%2 = 0) \\ f_{G,i,j,k} + a g_{y,i,j,k}^{(t)} & (t\%2 = 1) \end{cases} \tag{10}$$

$$f_{B,i,j,k}^{(t)} = \begin{cases} f_{B,i,j,k} + a g_{x,i,j,k}^{(t)} & (t\%2 = 0) \\ f_{B,i,j,k} + a g_{y,i,j,k}^{(t)} & (t\%2 = 1) \end{cases} \tag{11}$$

where a is a positive constant, and the notation % represents the remainder operation. If $f^{(t)}_{R,i,j,k}$ is less than 0, then $f^{(t)}_{R,i,j,k}$ must be set to 0, and if $f^{(t)}_{R,i,j,k}$ is greater than $U-1$, then $f^{(t)}_{R,i,j,k}$ must be set to $U-1$. If $f^{(t)}_{G,i,j,k}$ is less than 0, then $f^{(t)}_{G,i,j,k}$ must be set to 0, and if $f^{(t)}_{G,i,j,k}$ is greater than $U-1$, then $f^{(t)}_{G,i,j,k}$ must be set to $U-1$. If $f^{(t)}_{B,i,j,k}$ is less than 0, then $f^{(t)}_{B,i,j,k}$ must be set to 0, and if $f^{(t)}_{B,i,j,k}$ is greater than $U-1$, then $f^{(t)}_{B,i,j,k}$ must be set to $U-1$.

The process of Steps 2 and 3 is repeated T times, and a video composed of the pixel values $f^{(T)}_{R,i,j,k}$, $f^{(T)}_{G,i,j,k}$, and $f^{(T)}_{B,i,j,k}$ is the checkered-pattern video.

3. **Experiments.** Experiments were conducted using an RGB-D video which consists of 440 frames, 49 frames/second, $320*180$ pixels, and 256 gradations. The RGB-D video was shot using ZED stereo camera to capture a scene of a man moving indoor. The farthest distance from the camera was 5.987 meters. The RGB and depth images at the 1st, 40th, 80th, 120th, 160th, 200th, 240th, 280th, 320th, 360th, 400th, and 440th frames of the RGB-D videos are shown in Figures 2 and 3, respectively. In reference to [11, 16], the values of the parameters $W_{\min}$, $W_{\max}$, a, and T used in all experiments were set to 2, 4, 60, and 40, respectively. Hereinafter, the checkered-pattern video generated by the proposed method under the above conditions is referred to as a proposed checkered-pattern video, and the checkered-pattern video generated by applying the conventional method [16] to each frame of the RGB-D video under the above conditions is referred to as a conventional checkered-pattern video.

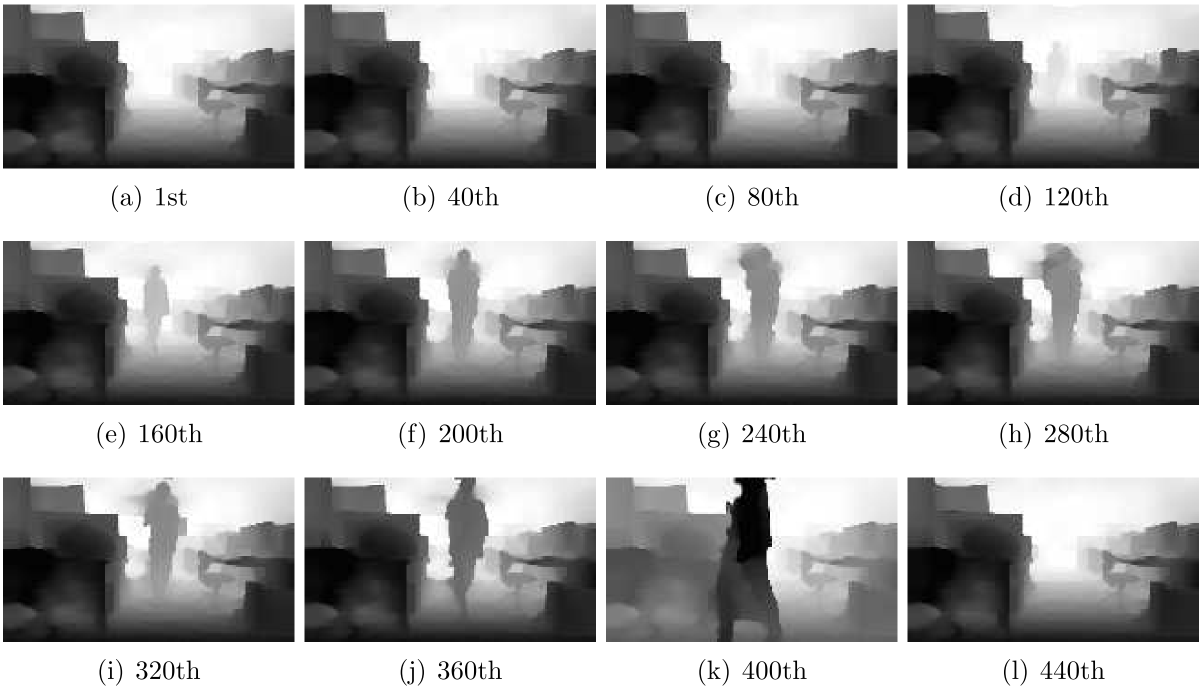

(a) 1st (b) 40th (c) 80th (d) 120th

(e) 160th (f) 200th (g) 240th (h) 280th

(i) 320th (j) 360th (k) 400th (l) 440th

Figure 2. RGB images

The proposed and conventional checkered-pattern videos were visually evaluated. The proposed checkered-pattern video was generated with the value of the parameter O set to 2. The conventional checkered-pattern video had many flickers. On the other hand, in the proposed video, flicker was suppressed, and the movement of checkered patterns was smooth. As an example, the proposed and conventional checkered-pattern images of the 350th and 351st adjacent frames are shown in Figures 4 and 5, respectively. From Figures 4 and 5 (especially, it is easy to understand if you look at the upper left and the


(a) 1st (b) 40th (c) 80th (d) 120th

(e) 160th (f) 200th (g) 240th (h) 280th

(i) 320th (j) 360th (k) 400th (l) 440th

Figure 3. Depth images

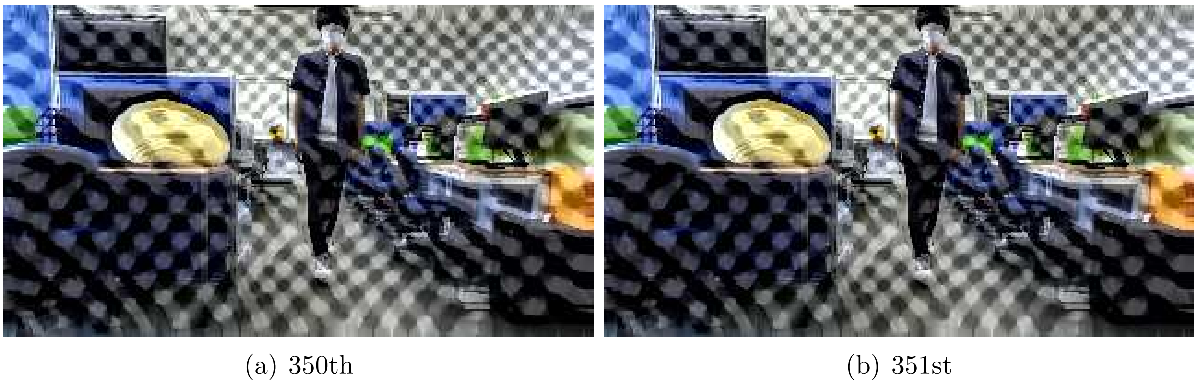

(a) 350th (b) 351st

Figure 4. Proposed checkered-pattern images

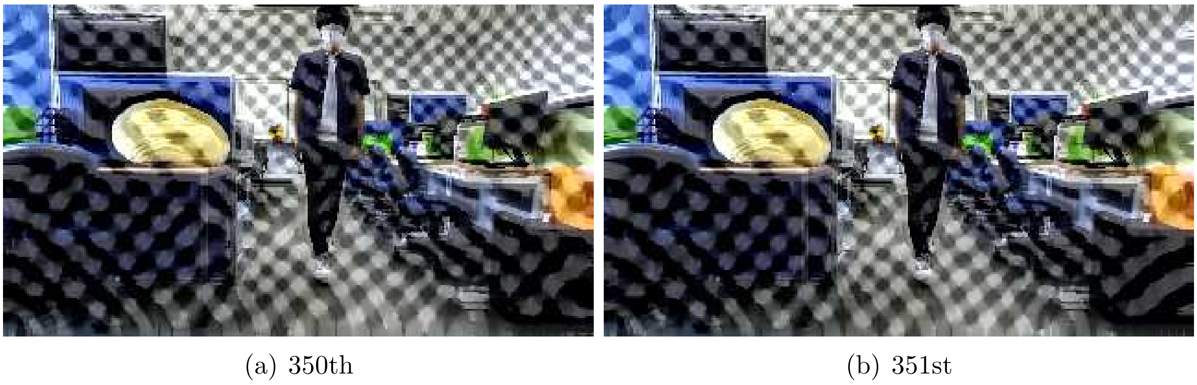

(a) 350th (b) 351st

Figure 5. Conventional checkered-pattern images

bottom right of the figure), the proposed checkered patterns changed less between the two adjacent frames than the conventional checkered patterns. This means that the proposed checkered-pattern video has less flicker than the conventional checkered-pattern video.

The proposed and conventional checkered-pattern videos were quantitatively evaluated. The averages of the absolute values of the differences between the pixel values between adjacent frames (hereinafter, frame difference averages) were calculated, and the average of the frame difference averages of all frames (hereinafter, all frame difference average) was calculated. The all frame difference average P is calculated by the following equation.

$$P = \frac{\sum_{k=1}^{K-1} \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{\left|f_{R,i,j,k}^{(T)} - f_{R,i,j,k+1}^{(T)}\right| + \left|f_{G,i,j,k}^{(T)} - f_{G,i,j,k+1}^{(T)}\right| + \left|f_{B,i,j,k}^{(T)} - f_{B,i,j,k+1}^{(T)}\right|}{3}}{K-1} \tag{12}$$

It is judged that the smaller the all frame difference average, the less the flicker of checkered-pattern videos. The all frame difference averages of the proposed and conventional checkered-pattern videos are shown in Tables 1 and 2, respectively. The proposed checkered-pattern videos were generated by changing the value of the parameter O to 1, 2, 3, and 4. The proposed checkered-pattern videos had smaller all frame difference averages than the conventional checkered-pattern video. More specifically, all frame difference averages of the proposed checkered-pattern videos were roughly 15% to 26% of that of the conventional checkered-pattern video. Therefore, the proposed checkered-pattern video had less flicker than the conventional checkered-pattern video. In addition, the larger the number of the forward and backward used, the smaller the all frame difference averages, and the changes in the all frame difference averages after the value of O was 2 were smaller than the change in the all frame difference averages when the values of O were 1 and 2. Therefore, it is considered that the value of O should be greater than 2 and need not be so large.

Table 1. All frame difference averages of the proposed checkered-pattern videos

O	All frame difference average
1	201550.087
2	130646.095
3	125628.731
4	116024.617

Table 2. All frame difference average of the conventional checkered-pattern video

All frame difference average
771243.347

4. Conclusions.

This paper proposed an NPR method to automatically generate checkered-pattern videos from RGB-D videos. When calculating each frame of checkered-pattern videos, flicker was suppressed by using the forward and backward frames of RGB-D videos. Through experiments using an RGB-D video taken by the authors, the appearance of checkered-pattern videos generated by the proposed method was evaluated visually and quantitatively. As a result of the experiments, it was found that the proposed method can suppress flicker.

The future task is to apply the proposed method to more RGB-D videos, although this paper applied the proposed method to one type of RGB-D video. Another future task is to further elaborate the proposed method.

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

REFERENCES

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

[2] 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.

[3] D.-L. Way, M.-K. Yang, Z.-C. Shih and R.-R. Lee, A colored pencil non-photorealistic rendering for 2D images, *International Journal of Innovative Computing, Information and Control*, vol.10, no.1, pp.233-241, 2014.

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

[5] 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.

[6] Q. Sun, Y. Chen, W. Tao, H. Jiang, M. Zhang, K. Chen and M. Erdt, A GAN-based approach toward architectural line drawing colorization prototyping, *The Visual Computer*, vol.38, no.4, pp.1283-1300, 2022.

[7] Y. Zhao, D. Ren, Y. Chen, W. Jia, R. Wang and X. Liu, Cartoon image processing: A survey, *International Journal of Computer Vision*, vol.130, no.11, pp.2733-2769, 2022.

[8] 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.

[9] K. Inoue and K. Urahama, Halftoning with minimum spanning trees and its application to maze-like images, *Computers & Graphics*, vol.33, no.5, pp.638-647, 2009.

[10] T. Hiraoka and Y. Tsurunari, Generation of moire-like images using Bezier surfaces, *ICIC Express Letters*, vol.16, no.12, pp.1309-1314, 2022.

[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] A. Buades, B. Coll and J. M. Morel, The staircasing effect in neighborhood filters and its solution, *IEEE Transaction on Image Processing*, vol.15, no.6, pp.1499-1505, 2006.

[13] N. Pierazzo and G. Facciolo, Data adaptive dual domain denoising: A method to boost state of the art denoising algorithms, *Image Processing On Line*, vol.7, pp.93-114, 2017.

[14] C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images, *Proc. of the 6th International Conference on Computer Vision*, Bombay, India, pp.839-846, 1998.

[15] S. Paris, P. Kornprobst, J. Tumblin and F. Durand, Bilateral filtering: Theory and applications, *Foundations and Trends in Computer Graphics and Vision*, vol.4, no.1, pp.1-73, 2008.

[16] T. Hiraoka and R. Takaki, Applying method of generating checkered pattern images using Prewitt filter to RGB-D images, *Journal of Advances in Artificial Life Robotics*, vol.2, no.4, pp.384-387, 2022.

[17] B. J. Meier, Painterly rendering for animation, *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96)*, pp.477-484, 1996.

[18] T. Hiraoka and M. Hirota, Generation of cell-like color animation by inverse iris filter, *ICIC Express Letters*, vol.12, no.1, pp.23-28, 2018.

[19] T. Hiraoka and R. Takaki, Extension of striped image generated by inverse line convergence index filter to video, *Journal of Advances in Artificial Life Robotics*, vol.12, no.1, pp.234-238, 2021.