2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

Motion vector estimation of textureless objects exploiting reaction-diffusion cellular automata

Miho Ushida†, Kazuyoshi Ishimura†, Alexandre Schmid‡, Tetsuya Asai†, and Masato Motomura†

†Graduate School of Information Science and Technology, Hokkaido University
Kita 14 Nishi 9, Kita-ku, Sapporo 060-0814, Japan
‡Microelectronic Systems Laboratory, École Polytechnique Fédérale de Lausanne CH-1015, Lausanne, Switzerland
Email: ushida@lalsie.ist.hokudai.ac.jp, ishimura@lalsie.ist.hokudai.ac.jp

Abstract—It is difficult to detect motion vectors for textureless objects, because variations in brightness are needed for motion vector estimation. To solve this problem, we proposed a reaction-diffusion (RD) texture generation algorithm. The RD model is a well-known method of spatial pattern generation. Using RD, we were able to generate texture on textureless objects. In our research, we have tried to generate texture on objects by processing the two-dimensional RD. However, this poses the problem of suppressing noise diffusion. Therefore, in this study, we attempt to inhibit noise diffusion by processing the one-dimensional RD model.

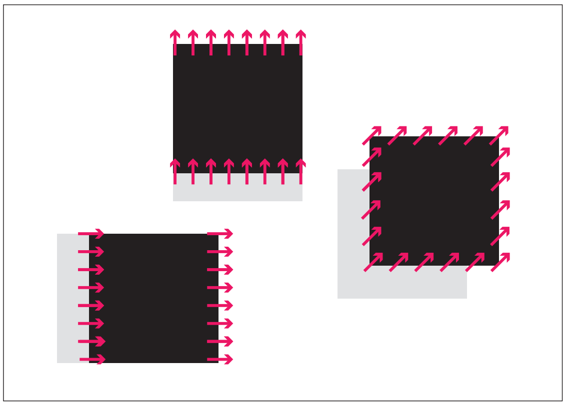

Figure 1: Motion estimation for a textureless object

1. Introduction

Motion estimation is used in various applications, such as anomaly detection, game interfaces, hand gesture user interfaces [1], and image stabilization [2]. This technology has been one of the hottest fields of study in recent years.

The block matching method is frequently used in motion vector estimation algorithms to determine the movement of an object in a video sequence. Block matching has a problem in that it requires variations in brightness. Thus, when we attempt to detect the motion of textureless objects, motion vectors cannot be detected, except at the object boundaries (Fig. 1). For example, when a completely black board moves, we cannot determine it as a board or a frame with motion vectors.

Therefore, we attempted to assign some texture to textureless objects. Texture that follows the object's movement assists the process of motion estimation. We assigned textures to textureless objects, and could detect motion vectors on both the boundary and the interior of an object. We focused on the reaction-diffusion (RD) model. The RD model is a well-known method for spatial pattern generation [3]. It can be used to simulate the diffusion of chemical activators and inhibitors, and the amplification of their difference. These dynamics can generate the stable striped or spotted patterns observed in nature on the bodies of animals, fish, and so on. Thus, we can generate texture from the boundary information using the RD model. The texture has a certain spatial frequency.

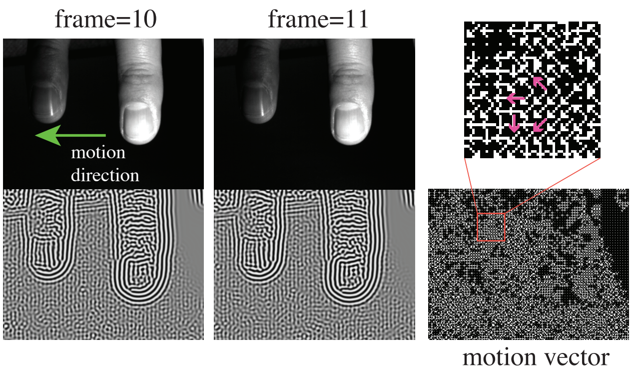

Figure 2: Snapshot of two-dimensional RD preprocessing and motion estimation

To date, we have tried to generate texture on objects by processing the two-dimensional RD, i.e., the diffusion field is two-dimensional (Fig. 2) [4]. However, this presents the problem of suppressing noise diffusion. Noise diffuses and is also amplified, so different textures are generated between neighboring frames, even when the objects have barely moved. This leads to errors in motion estimation.

Figure 3: Sensitivity of two-dimensional RD; the picture at top-left is from an input without noise, the others are from input with different levels of noise

2. Algorithm

2.1. One-dimensional reaction-diffusion

In this study, we attempt to inhibit the noise diffusion by processing the one-dimensional RD model.

One-dimensional RD has two merits. First, it is hard for the noise to diffuse—there is nowhere for it to escape. In two-dimensional RD, noise has a four-way diffusion field (up, down, left, and right). Hence, two-dimensional RD is very sensitive to noise (Fig. 3). In contrast, one-dimensional RD presents a diffusion field with only two directions (up/down or left/right). Therefore, noise does not diffuse as much, and has less of an effect on texture generation. The second merit is simply that the hardware implementation is easier than for the two-dimensional case.

However, one-dimensional RD also has a limitation. Noise affects the generation of a regular pattern, because there is too little escape (Fig. 4). To solve this problem, we add a filter (diffusion) process on every third iteration of the RD process, starting on the fourth update. This diffusion has no subtraction or amplification process. As a result, we can inhibit the influence of noise and obtain a regular pattern (Fig. 5).

2.2. Proposed algorithm

In the algorithm proposed in this paper, a two-dimensional input image is first divided into a one-dimensional arrangement, x and y. These arrangements are then repeatedly processed by one-dimensional RD. Finally, they are multiplied together.

The simulation results given by this algorithm are shown

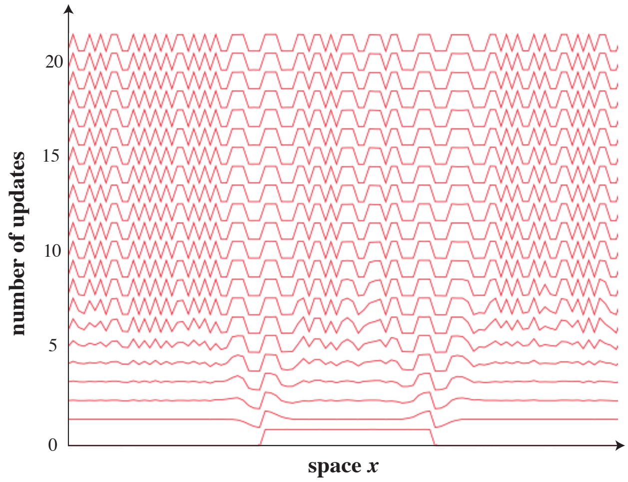

Figure 4: Waveform without additional filter processing

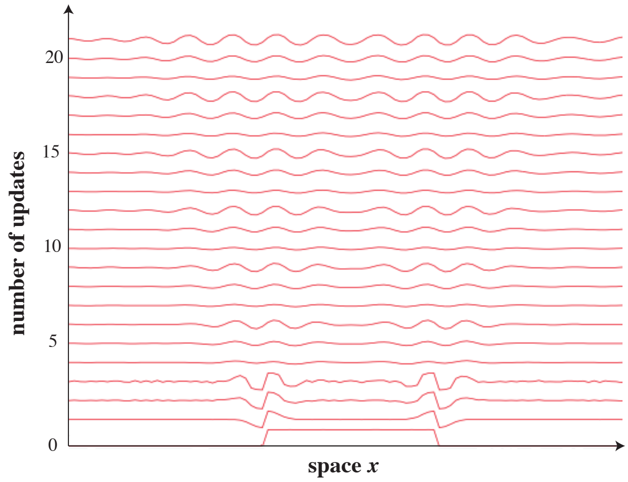

Figure 5: Waveform with additional filter processing every three updates

in Fig. 6. The input image includes noise. Fig. 6 (a) is the result without additional filter processing. As for the one-dimensional pattern, the generated texture is disordered by the influence of noise. Fig. 6 (b) is the result with additional filter processing. An ordered texture has been generated, unlike in Fig . 6 (a).

3. Simulation results

3.1. No background

The simulation results with no background are shown in Fig. 7. The moving object is a white square. Texture is generated from the boundary of the square. However, the texture not only expands inside the object, but also spills outside the boundaries. This leads to errors in motion detection. We had to confirm the texture generation with some background.

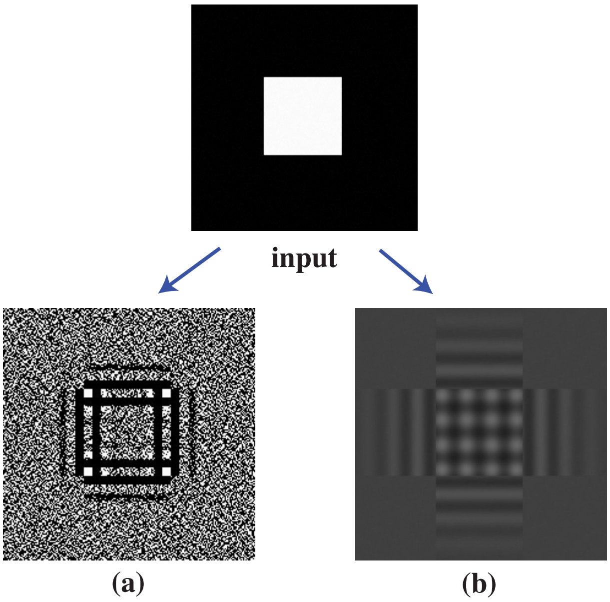

Figure 6: Difference in the texture generated by the proposed algorithm: (a) without additional filter processing and (b) with additional filter processing

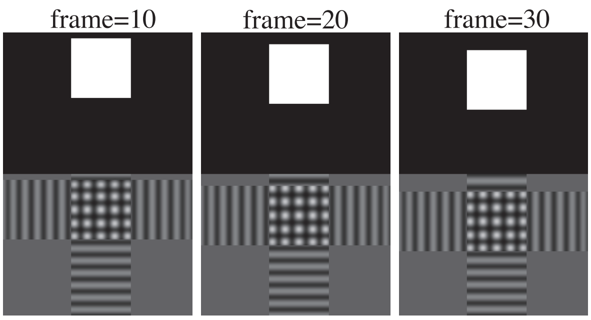

Figure 7: Snapshot in the case without background

3.2. Ideal background

Simulation results with an ideal background are shown in Fig. 8. As an ideal background, we processed a cross-stripe image with the proposed algorithm until the generated texture became stable. As can be seen, the outside expansion of texture has been inhibited. Thus, the proposed algorithm will theoretically work well.

3.3. Real background

To confirm the practicality of the algorithm, we simulated the texture generation with a real background. Simulation results with a bookshelf as the real background are shown in Fig. 9. In this case, the outside expansion of texture is again inhibited, and a stable texture is generated inside the object boundaries.

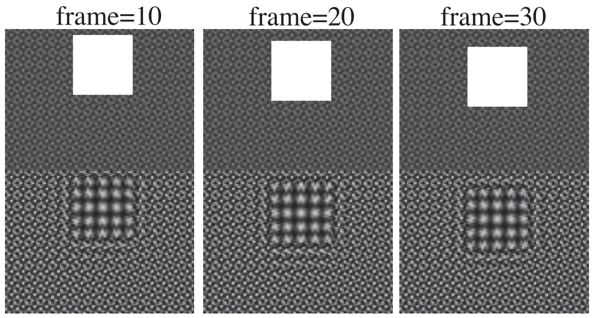

Figure 8: Snapshot in the case of an ideal stable background

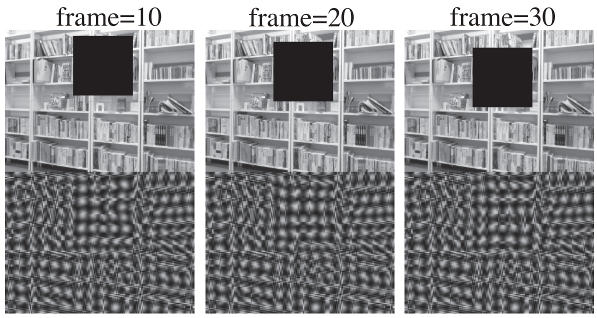

Figure 9: Snapshot in the case with a real background

3.4. Motion estimation

In the motion estimation stage, we detect the motion vector in images from the generated texture. As a motion estimation method, we used the optical flow function in OpenCV, an image processing library. The results of motion estimation with a real background image are shown in Fig. 10. In the upper image without RD processing, the motion vector is only detected on the boundary of the square. However, in the lower image with RD processing, the motion vector is detected both on and inside the boundary. Furthermore, the direction of the motion vector is almost correct.

4. Hardware

In prior research, the two-dimensional RD algorithm [5] has been implemented on digital hardware [6]. However, the RD hardware was based on two-dimensional RD, and required a considerable amount of memory. Based on this, we are working on a hardware implementation for the one-dimensional RD model. The current RD texture generation module is shown in Fig. 11.

Fig. 11 (a) is the blur filter kernel from the nearest two pixels. The output around a_i is described as

$$\text{kernel out}_{a_i} = \frac{a_{i-1} + 2a_i + a_{i+1}}{4} \tag{1}$$

where i is the ith pixel in a row of an image. This kernel

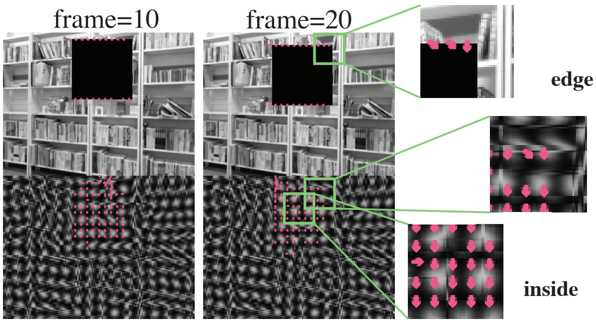

Figure 10: Snapshot of motion estimation with a real background

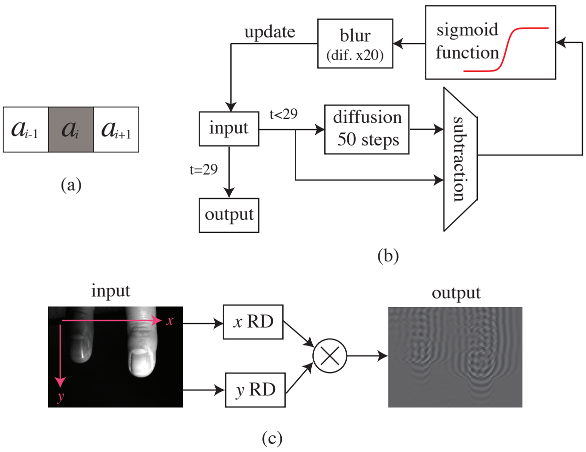

Figure 11: Current RD module: (a) one-dimensional filter kernel, (b) proposed one-dimensional RD algorithm, (c) system flow of a proposed hardware-oriented one-dimensional RD architecture

is used in the diffusion process and additional blur processing.

As shown in Fig. 11 (b), the diffusion process in the figure corresponds to the diffusion of inhibitors. Against this, non-diffusion corresponds to the diffusion of activators. The difference between these corresponds to the reaction of inhibitors and activators. The sigmoid function amplifies the output. As noted previously, the final blur processing corrects disorders introduced to the pattern by noise.

Fig. 11 (c) illustrates the system flow from an input image to the output image. Input images are processed in parallel in two direction, x and y. However, even with parallel processing, some delay is inevitable because the processing order is different. This issue will be considered in future work.

5. Summary

In this study, to enable motion estimation on the inside of the boundary of textureless objects, we proposed a texture generation algorithm that uses RD processing. To inhibit the diffusion of noise, we generated texture by combining one-dimensional RD processing. The simulation results showed that, without a suitable background, the texture expands outside the object; however, in the case with a background, the outside expansion of texture was inhibited. Furthermore, when RD texture was generated on the textureless object, we were able to detect the motion vector on the inside of the boundary. The direction of the detected motion vectors was almost correct, with only slight errors. In future work, we will seek to develop a hardware implementation that enables real-time processing.

References

[1] M. Tanaka, Japan patent Kokai JP2014-52934A, 2014.

[2] Y. Takagi, Japan patent Kokai JP2012-15959A, 2012.

[3] J.D. Murray, Mathematical Biology I: An Introduction (3rd Ed.), Chap. 7, p. 239, Springer, New York, 2002.

[4] M. Ushida, K. Ishimura, T. Asai, and M. Motomura, “A reaction-diffusion algorithm for texture generation towards motion-vector estimation of textureless objects,” 2015 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia, pp. 361-364, 2015.

[5] K. Ishimura, K. Komuro, A. Schmid, T. Asai, and M. Motomura, “Image steganography based on reaction diffusion models toward hardware implementation,” Nonlinear Theory and Its Applications, vol. 5, no. 4, pp. 456-465, 2014.

[6] K. Ishimura, K. Komuro, A. Schmid, T. Asai, and M. Motomura, “FPGA implementation of hardware-oriented reaction-diffusion cellular automata models,” Nonlinear Theory and Its Applications, vol. 6, no. 2, pp. 252-262, 2015.