2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

Cellular Automaton Based Pixel Level Snakes Using Active Contour Curvature

FUJITA Tomohiro†, SAWADA Sho†, KISHIMOTO Koki†, KUMAKI Takeshi† and OGURA Takeshi†

†Department of Electronic and Computer Engineering, Ritsumeikan University
1-1-1 Noji-higashi, Kusatsu, Shiga, Japan
Email: tfujita@se.ritsumei.ac.jp, kumaki@fc.ritsumei.ac.jp, togura@se.ritsumei.ac.jp

Abstract—One of image segmentation methods, the snake algorithm, is realized using Cellular Automaton (CA). In this study, we propose CA based Pixel Level Snakes (PLS) which utilizes curvature of active contour. In the original snake algorithm, contractile force is determined by curvature of active contour, so our proposed method follows the original concept. For the calculation of curvature, the local curvature counting algorithm is used. This algorithm is based on CA, and therefore this method can be easily implemented CA dedicated hardware, CAM^2. The experimental result shows that our proposed method can obtain equal or better performance than the previous one. The number of the CA rule for the proposed method is less than previous one, so it makes the implementation more easier and processing speed faster. That is advantage of our method.

1. Introduction

Image segmentation is one of key issues in the computer vision discipline. The active contour method or snake algorithm is one of the image segmentation methods [1]. Pixel Level Snakes (PLS) [2, 3, 4, 5] is a kind of snake algorithm. In PLS, the development of the active contours are governed by a balance of two forces which are from an internal energy and an external potential in each pixel. This PLS algorithm is based on Cellular Neural Network (CNN) and can be processed in parallel. So it can be implemented on parallel processing machines. In Ref. [2, 3, 4, 5] PLS was implemented on Cellular Neural Network (CNN) dedicated machine.

We implemented PLS on our parallel processing machine, Cellular AutoMata on Content-Addressable Memory(CAM^2) [6], in previous works [7, 9]. CAM^2 is basically developed as a hardware dedicated engine for Cellular Automaton (CA), so we modified the PLS algorithm to process with CA. CA based approach consists of a sires of simple CA rules. CAM^2 was developed as a CA dedicated hardware engine. So, it is possible that the CA based PLS realizes real-time image processing on CAM^2.

The CA rules for PLS, however, has no physical background, so that is a drawback of this method. Especially, the force from curvature of the contour is empirically defined. In this paper we propose the PLS algorithm based on the curvature of the contour. For the calculation of the curvature, the local curvature counting algorithms [10] is utilized. This algorithm is CA based method, so this is also easily implemented on CAM^2. The force which required for evolution of the active contour is calculated by the curvature. It makes physical meanings more clear, so improvement of the performance can be expected.

This paper is structured as follows. In Sec. 2, the CA based PLS algorithm is illustrated. The local curvature counting algorithms and the proposed algorithm utilizing it is introduced in Sec. 3. The experiments are demonstrated in Sec. 4. Finally, this paper is concluded in Sec. 5.

2. CA based PLS

Snake algorithm is an image segmentation method in image processing [1]. In snake algorithm, a closed curve successively deforms according to an external potential from an image and an internal energy of the active contour, and finally, the curve stops at the object contour of the image. In the motion of the snake, internal energy is calculated from the curvature and the stretchiness of the curve.

Pixel level snakes (PLS) is evolved version of original snake algorithm [2, 3, 4, 5]. In PLS, two forces from the internal energy and the external potential are calculated, and taking balance between these two forces, the motion of the curve is determined in the pixel level.

The PLS algorithm [2, 3, 4, 5] is based on Cellular Neural Network (CNN) [8]. CNN is a kind of neural network whose connectivity is only restricted among adjacent cells. In Ref. [2, 3, 4, 5], PLS was implemented on CNN dedicated hardware, and this implementation achieved high-speed parallel processing.

The CA based PLS algorithm is realized by three state CA. The three states correspond to the three types of cells, and first, second, and third state is assigned to the cells on contour, the cells in inner region, and the cells in outer region, respectively. A closed curve moves taking balance between two forces caused by internal energy and external potential.

In PLS process, the two forces are compared by the fol-

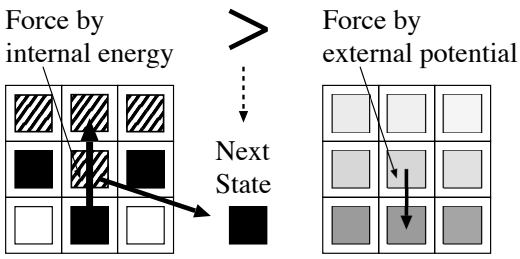


Figure 1: Conceptual scheme of CA based PLS

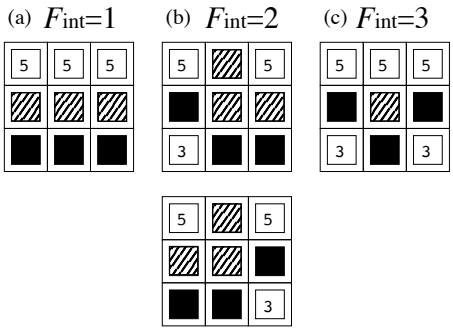


Figure 2: Forces for north

lowing equation,

$$a \cdot F_{\text{int}} - F_{\text{ext}}, \tag{1}$$

where F_{int} is force by internal energy of the contour, F_{ext} is force caused by external potential, and a is a scale parameter. The value of F_{int} is given by CA rule, and F_{ext} is calculated by taking absolute value of the gradient of the processing image. The conceptual scheme of CA based PLS is depicted in Fig. 1. In Fig. 1, F_{int} is compared with the force F_{ext}, and if Eq. 1 is greater than 0, then the center cell in Fig. 1 changes into contour cell.

2.1. CA rules for PLS

In Ref. [9] we proposed CA based PLS. This consists of rules which define the F_{int} and thinning and connectivity rules for active contour. The rule for F_{int} contains two types: forces in vertical and horizontal directions and diagonal direction. We illustrate these rules in following part of this section.

Forces in vertical and horizontal directions

The forces for vertical and horizontal directions are shown in Fig. 2. In Fig. 2, the labels of the cells mean the states of the cells. The labels are assigned to the states as shown in Tab. 1. In Fig. 2, only north direction is shown, and the other directions can obtain by rotating these rules.

Figure 2 (a) shows the force caused by the membrane energy. This force represents the contract force to resist the stretch of the contour. We assign a weak force ($F_{\text{int}} = 1$) to this kind of force. Figure 2 (b) and (c) show the force caused by the thin plate energy, which weighs its resistance to bending. The bending in Fig. 2 (c) is sharp, thus the force F_{int} is stronger than (b).

Forces in diagonal direction

The force for northwest is shown in Fig. 3. The meanings of the labels are shown in Tab. 1. The forces of the other diagonal direction can be obtained by rotating this rule. This force is from the membrane energy.

Thinning rule

When the contour evolve by the forces, the line width of the contour broadens. The excess contour must be removed to keep the line width thin. Thinning rule is shown in Fig. 4. As shown in Fig. 4, the contour cell which is not next to the inner cell transits to the outer cell.

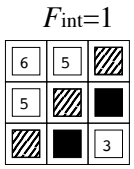


Figure 3: Forces for northwest

Table 1: Label assignment to CA state

Label	Assgned state
□	outer cell
■	contour cell
▨	inner cell
3	outer or contour cell
4	outer or inner cell
5	contour or inner cell
6	all kinds of cell

Connectivity rule

In PLS, a continuity of the contour pixels must be guaranteed. For this purpose, if an discontinuous point is encountered, the algorithm have to modify it to maintain the connectivity. The rule for connectivity is shown in Fig. 5. This rule means if a outer cell is next to a inner cell in horizontal or vertical direction, this cell is replaced with a contour cell.

3. Local curvature counting algorithms

The drawback of the above mentioned method is that it has no physical background, especially the force F_{int} is empirically defined. In original snake algorithm [1] this value is determined from curvature of the contour, so our algorithm should be modified to be based on the curvature of active contours. In this study, we utilize local curvature counting algorithms [10] to estimate curvature of the contour. The local curvature counting algorithm is a curvature

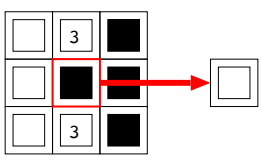


Figure 4: Thinning rule. Self-state is contour cell and no inner cells in neighbor cells.

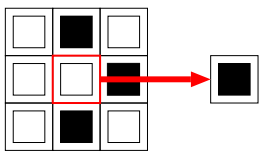


Figure 5: Connectivity rule. Self-state is outer cell and more than one inner cell in north, east, south, and west cells

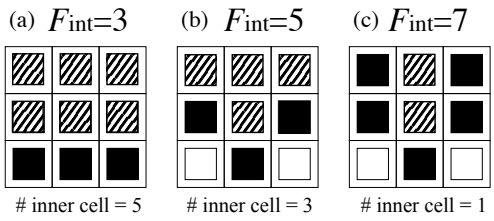


Figure 6: Curvature estimation

estimate method with CA. In next section, we will introduce the local curvature counting algorithms.

Local curvature counting algorithms [10] is a method which estimates curvature with CA. Using the estimated curvature, the internal force F_{int} can be determine and the active contour evolve with the above mentioned scheme. Curvature is approximated with the number of the inner cells which contains in 8 neighbor cells. Figure 6 shows examples of calculating curvature using this algorithm. In Fig. 6, the labels of the cells mean the states of the cells. The labels are assigned to the states as shown in Tab. 1. In Fig. 6 the number of the inner cells are also shown.

The force by internal energy F_{int} is defined by following equation.

$$F_{\text{int}} = N_{\text{neigh}} - N_{\text{i}},$$

where N_{neigh} is the number of neighbor cells and N_{i} is the number of inner cell among neighbor cells. In this paper N_{neigh} is assumed to be 8 in all cases. In Fig. 6 the calculated values of F_{int} using Eq. 3 are also depicted in each case. Note that the value defined here is not an absolute value but a relative value. In our algorithm, F_{int} is multiplied by scaling parameter a to adjust the balance of force with F_{ext}.

The number of rules of the curvature based method is less than the previous method. So it can be easily implemented on CAM^2, and the processing speed will be faster.

4. Experiments

The performance evaluation of CA based PLS was conducted. We picked up 7 images from the Berkeley Segmentation Dataset and Benchmark (BSDS) [11]. BSDS supplies 200 images for training and 100 images for test with answers by manual segmentation. In this paper the answer data is referred to as FG data. In the evaluation, we try two method, namely, the method proposed in previous work [9] and the one based on the local curvature counting algorithm.

After segmentation process, matching between processed data and FG data is quantitatively evaluated. For this purpose we use F-measure as a figure. We define the two figure 'Precision' and 'Recall' to define the F-measure.

$$\text{Precision} = \frac{T_{\text{P}}}{T_{\text{P}} + F_{\text{P}}} \tag{2}$$

$$\text{Recall} = \frac{T_{\text{P}}}{T_{\text{P}} + F_{\text{N}}} \tag{3}$$

Here, T_{P} is a number of pixels which is identical to FG data among contour pixels, F_{P} is a number of pixels which is not identical to FG data among contour pixels, and F_{N} is a number of pixels which is not identical to contour pixels among segment pixels in FG data. Precision indicates the ratio of the correct pixels to contour pixels, and Recall indicates the ratio of correct pixels to segment pixels in FG data. F-measure is defined as a harmonic mean of these two figures, and its formula is as follows.

$$\text{F} - \text{measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

The F-measure of 7 images from BSDS is illustrated in Fig. 7. In Fig. 7 the data processed by both the previous method and the curvature counting method is shown. The experimental result depends on the parameter a, so we conducted the experiments with the value of 0.1, 0.2, 0.3 and 0.4 for a. In Fig. 7, the maximum values of F-measure among various values of a are shown.

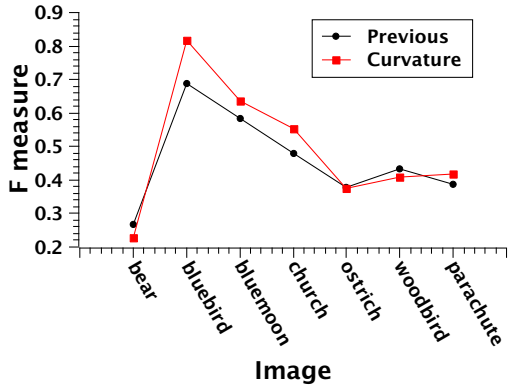
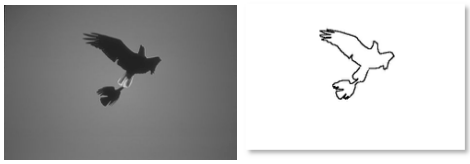


Figure 7: The maximum value of F-measure



(a) Image (b) FG

Figure 8: Test image of *bluebird*

From the experimental result, it can be seen that the proposed method can obtain equal or better performance than the previous one.

Next, the details of experimental result is shown. The image '*bluebird*' and its FG data is shown in 8 (a) (b), respectively. The CA based PLS was applied to this image with the value of a as 0.1, 0.2, 0.3 and 0.4. The processed images with the previous and the curvature based method are shown in Figs. 9 and 10, respectively. The values of F-measure are shown in Fig. 11.

In Fig. 11 we can see that the F-measure largely changes, depending on the value of a. The value of a which gives the maximum value of F-measure depends on images, so it is required to determine the value of a for each image. It was also observed that relatively high value of F-measure

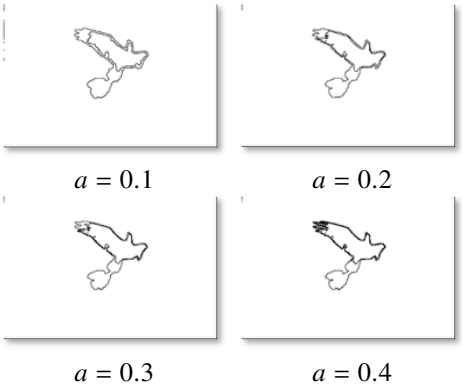


Figure 9: Processed image with the previous method

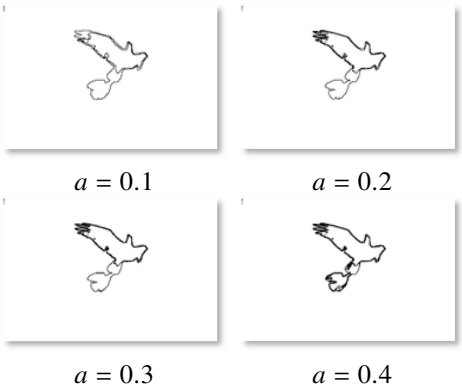


Figure 10: Processed image with the proposed method

is taken for the sharp contrast image such as the image *bluebird*.

5. Conclusion

A CA based PLS algorithm based on curvature of active contour is proposed. For the estimation of curvature the local curvature counting algorithm is utilized. This algorithm is also based on CA, so the proposed method can be easily implemented on CAM^2. Comparing the proposed method with the previous CA based PLS algorithm, the proposed method can obtain equal or better performance than the previous one. Taking into account of the easiness of implementation on CAM^2, the proposed method has advantage over the previous one. The implementation of the proposed method on CAM^2 is a future work.

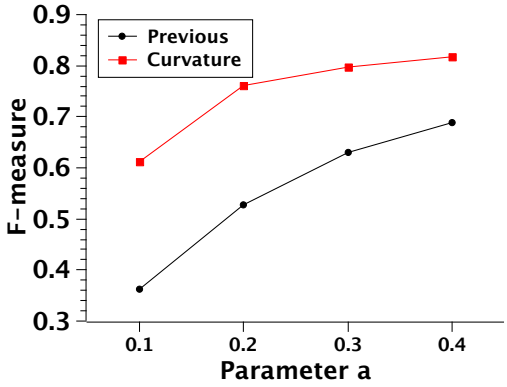


Figure 11: The F-measure value of image *bluebird* with various *a*

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988, 10.1007/BF00133570. [Online]. Available: http://dx.doi.org/10.1007/BF00133570

[2] D. Vilariño, D. Cabello, A. Mosquera, and J. Pardo, "Application of a multilayer discrete-time cnn to deformable models," in *Biological and Artificial Computation: From Neuroscience to Technology*, ser. Lecture Notes in Computer Science, J. Mira, R. Moreno-Díaz, and J. Cabestany, Eds. Springer Berlin Heidelberg, 1997, vol. 1240, pp. 1193–1202. [Online]. Available: http://dx.doi.org/10.1007/BFb0032579

[3] D. Vilarino, D. Cabello, X. Pardo, and V. Brea, "Cellular neural networks and active contours: a tool for image segmentation," *Image and Vision Computing*, vol. 21, no. 2, pp. 189 – 204, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885602001531

[4] D. Vilarino and C. Rekeczky, "Implementation of a pixel-level snake algorithm on a CNNUM-based chip set architecture," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 5, pp. 885 – 891, may 2004.

[5] D. L. Vilariño and C. Rekeczky, "Pixel-level snakes on the CNNUM: algorithm design, on-chip implementation and applications," *Int. J. Circuit Theory Appl.*, vol. 33, no. 1, pp. 17–51, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1002/cta.v33:1

[6] T. Ikenaga and T. Ogura, "CAM^2: A highly-parallel two-dimensional cellular automaton architecture," *IEEE Trans. Comput.*, vol. 47, no. 7, pp. 788–801, 1998.

[7] T. Matsui, T. Fujita, Y. Tsuji, T. Kumaki, M. Nakanishi, and T. Ogura, "Evaluation of advanced pixel-level snakes on cellular hardware platform," in *New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International*, 2013, pp. 1–4.

[8] L. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Trans. Circuits Syst.*, vol. 37, no. 12, pp. 1257–1272, Oct. 1988.

[9] S. Sawada, T. Fujita, K. Iwanaga, T. Kumaki, M. Nakanishi, and T. Ogura, "A method using ca-based pixel-level snakes for multiple-contour expansion," *Journal of Signal Processing*, vol. 18, no. 4, pp. 221–224, 2014.

[10] P. J. Pimienta, E. J. Garboczi, and W. C. Carter, "Cellular automaton algorithm for surface mass transport due to curvature gradients simulations of sintering," *Computational Materials Science*, vol. 1, no. 1, pp. 63 – 77, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/092702569290008W

[11] Berkeley Computer Vision Group, "The Berkeley segmentation dataset and benchmark." [Online]. Available: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html