2010 International Workshop on Nonlinear Circuits,
Communication and Signal Processing
NCSP'10, Waikiki, Hawaii, March 3-5, 2010

Color Texture Boundary Detection Using Three-Layer CNN Based on Hybrid-Feature

Takashi Inoue† and Yoshifumi Nishio†

†Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Phone:+81-88-656-7470, FAX:+81-88-656-7471
Email: : {takashi, nishio}@ee.tokushima-u.ac.jp

Abstract

In this paper, a color texture boundary detection using cellular neural networks (CNN) is presented. The approach is similar to the early vision system of the human brain. The proposed algorithm has been tested on synthetic color texture images. Additionally, we carry out the error evaluation between ideal boundaries and detected boundaries. From our simulations, we confirm that the boundaries can be detected like human's sensation.

1. Introduction

Researchers in recent decades have elucidated the signal transduction in the retina and the function of the visual cortex. The highly flexible nature of the neural circuits in the visual cortex, especially the early vision, has been an interesting subject for studying and developing the neural plasticity. Cellular Neural Networks (CNN) were introduced by Chua and Yang in 1988 [1] and are one of the neural circuits in the retina. The idea of CNN was inspired from the architecture of the cellular automata and the neural networks. Unlike the conventional neural networks, CNN has local connectivity property. CNN can be used for various applications of the image processing [2].

The early vision represents the first stage of the visual processing. These mechanisms are operated in parallel across the visual field and are believed to be used for detecting the most basic visual features, such as contrast, edge, grouping, brightness and lightness. A. K. Jain and F. Farrokhnia [3] confirmed as follows. There are two fundamental features in the human visual system: one is that the difference of light intensity projects into the retina, the other is that the behavior of the retina is similar to the band-pass filtering. These facts implicate that the abrupt changes on the light intensity will cause the stronger stimulation in the visual cortex and those abrupt changes are called first-order feature. K. Obermayer and G. G. Blasde confirmed an orientation-selective neuron which is sensitive to the areas of lines and boundaries, called simple cells. The receptive field of the simple cells can be modeled by the Gabor function, and it has been widely used for the information extraction, which is the so-called second-order feature [4].

C.T. Lin et al. have proposed the texture segregation as the early vision using CNN [5]. They have confirmed the texture boundary detection with high accuracy by using the first- and second-order features. However, the first- and second-order features are created respectively, unlike the early vision system of the human brain. Additionally, in their research, a texture input image is the gray-scale image unlike the human visual system, and we think that the color information is one of the basic visual features for understanding the visual field.

In this paper, a color texture boundary detection using CNN is presented. The approach is similar to the early vision system of the human brain. The proposed algorithm for the texture boundary detection consists of the proposed and the conventional CNN. The proposed CNN can treat the color information and can consider the first- and second order features at the same time like the early vision system. The proposed algorithm has been tested on the synthetic color texture images. Additionally, we carry out the error evaluation between the ideal boundaries and the detected boundaries. From our simulations, we can confirm that the boundaries can be detected based on the first- and second-order features like the human visual system.

2. Proposed algorithm for texture boundary detection

In this section, we present the proposed algorithm for the color texture boundary detection using CNN. Figure 1 shows the flowchart of the proposed algorithm. Following subsections are going to introduce each block of the proposed algorithm.

2.1. Conversion method using HSB model

In the first step of the proposed algorithm, the synthetic color texture image is converted to three gray-scale images using HSB model. HSB model is similar to the perception of human, hence it is suitable for a treatment of the color information. Therefore, HSB model is used for the color texture

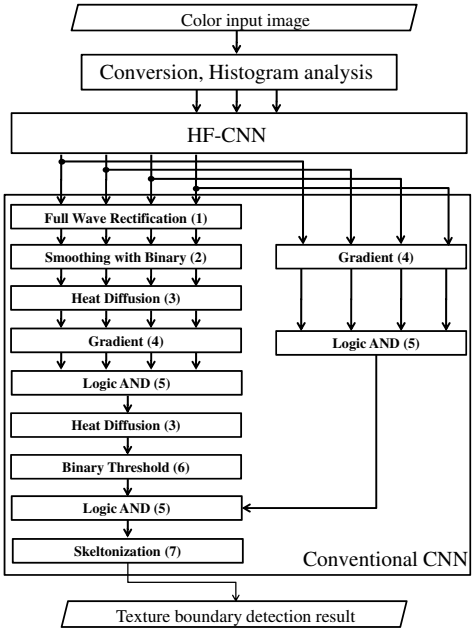


Figure 1: Proposed algorithm.

boundary detection in the proposed algorithm. From this conversion step, we can obtain the three gray-scale images.

2.2. Histogram analysis

In the second step of the proposed algorithm, three types of density histogram are created from each gray-scale image in order to decide a parameter in the proposed CNN. The parameter in the proposed CNN represent the input method of the three gray-scale images. Because the proposed CNN has three single CNNs, the three gray-scale images in the previous step are inputted to each layer of the proposed CNN. We decide the inputted gray-scale image from the three types of density histogram, because we do not know which gray-scale image should be inputted to each layer of the proposed CNN, in this input step.

2.3. CNN based on hybrid-feature (HF-CNN)

Figure 2 shows the structure of CNN based on the first- and second-order features, so-called the hybrid-feature CNN (HF-CNN). HF-CNN consists of three single CNNs. In particular, one of the three layers is called the isolate layer and the other two layers are called the two-layer. Hence, the isolate layer influences one layer of the two-layer, and the two-layer influence each other. Namely, the isolate layer is not influenced by other layers.

The dynamical system equations associated with HF-CNN are described by the following equations:

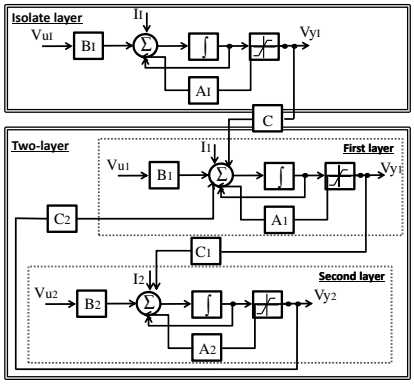


Figure 2: Structure of HF-CNN.

(1) Three first-order differential equations:
Equation of isolate layer:

$$\frac{dv_{xI,ij}}{dt} = -v_{xI,ij} + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} A_{I(i,j;k,l)}v_{yI,kl}(t) + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} B_{I(i,j;k,l)}v_{uI,kl}(t) + I_I, \quad (1)$$

Equations of two-layer:

$$\frac{dv_{x1,ij}}{dt} = -v_{x1,ij} + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} A_{1(i,j;k,l)}v_{y1,kl}(t) + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} B_{1(i,j;k,l)}v_{u1,kl}(t) + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} C_{(i,j;k,l)}v_{yI,kl}(t) + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} C_{2(i,j;k,l)}v_{y2,kl}(t) + I_1, \quad (2)$$

$$\frac{dv_{x2,ij}}{dt} = -v_{x2,ij} + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} A_{2(i,j;k,l)}v_{y2,kl}(t) + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} B_{2(i,j;k,l)}v_{u2,kl}(t) + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} C_{1(i,j;k,l)}v_{y1,kl}(t) + I_2, \quad (3)$$

(2) Three output equations:

$$v_{yI,ij}(t) = \frac{1}{2}(|v_{xI,ij}(t) + 1| - |v_{xI,ij}(t) - 1|), \quad (4)$$

$$v_{y1,ij}(t) = \frac{1}{2}(|v_{x1,ij}(t) + 1| - |v_{x1,ij}(t) - 1|), \quad (5)$$

$$v_{y1,ij}(t) = \frac{1}{2}(|v_{x2,ij}(t) + 1| - |v_{x2,ij}(t) - 1|), \quad (6)$$

where v_x, v_y, and v_u represent a state, an output and an input of the cell, respectively. In Eqs. (2) and (3), C, C_1 and C_2 are the coupling templates introduced to couple each layer.

The templates of each layer in HF-CNN are shown as follows.
Template of isolate layer:

$$A_i = \begin{bmatrix} 0.1 & 0.15 & 0.1 \\ 0.15 & 0 & 0.15 \\ 0.1 & 0.15 & 0.1 \end{bmatrix}, \; B_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; I_i = 0. \quad (7)$$

Templates of two-layer:

$$A_1 = \begin{bmatrix} 0 & \cos\Omega y & 0 \\ \cos\Omega x & -(3+\lambda)^2 & \cos\Omega x \\ 0 & \cos\Omega y & 0 \end{bmatrix}, \; B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; I_1 = 0. \tag{8}$$

$$A_2 = \begin{bmatrix} 0 & \cos\Omega x & 0 \\ \cos\Omega y & -(3+\lambda)^2 & \cos\Omega y \\ 0 & \cos\Omega x & 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I_2 = 0. \tag{9}$$

Coupling template :

$$C = \begin{bmatrix} a & a & a \\ a & 0 & a \\ a & a & a \end{bmatrix}, \quad a = |V_{yIij} - V_{yIkl}|/8. \tag{10}$$

$$C_1 = \begin{bmatrix} 0 & \sin\Omega y & 0 \\ -\sin\Omega x & 0 & \sin\Omega x \\ 0 & -\sin\Omega y & 0 \end{bmatrix}. \tag{11}$$

$$C_2 = \begin{bmatrix} 0 & -\sin\Omega x & 0 \\ \sin\Omega y & 0 & -\sin\Omega y \\ 0 & \sin\Omega x & 0 \end{bmatrix}. \tag{12}$$

First-Order Feature Extraction: As the introduction in the Section 1, the first-order feature represents the boundary in which the intensity of light suddenly changes. This fact implies that the first-order feature can be obtained by using the Gaussian operation and the Gradient operation. Therefore, in HF-CNN, the template of the isolate layer as Eq. (7) and the coupling template as Eq. (10) have an ability of the Gaussian operation and of the Gradient operation, respectively. By the effect of this setting, the first-order feature can be created.

Second-Order Feature Extraction: As the introduction in the Section 1, the simple cells can be modeled by the Gabor filter operation Eqs. (8), (9), (11) and (12). In the Gabor filter template, there are two parameters as Ω_x and Ω_y. We can control the 4 types orientations ($-45°$, $0°$, $45°$ and $90°$) of the Gabor filter template by changing Ω_x and Ω_y. Namely, we can obtain 4 output images corresponding to each orientation from HF-CNN.

Hybrid-Feature: The first-order feature is created in the isolate layer and the coupling part from the isolate layer to two-layer. Additionally, in the two-layer, the second-order feature is created with considering the first-order feature obtained by the isolate layer. Therefore, the output image from HF-CNN has the hybrid-feature combining the first- and the second-order features.

2.4. Creation of texture boundaries

In the fourth step of the proposed algorithm, the output images using HF-CNN are processed by using templates in the conventional CNN. The templates in the proposed algorithm are founded in [5] and [6].

Template 1: Full Wave Rectification Template [5].
Template 2: Smoothing with Binary Template [6].
Template 3: Heat Diffusion Template [6].
Template 4: Gradient Template [6].
Template 5: Logic AND Template [6].
Template 6: Binary Threshold Template [6].
Template 7: Skeltonization Template [6].

Finally, we can obtain the result of the texture boundary detection through a series of steps as Fig. 1.

3. Color texture boundary detection results

In this section, we show results of the color texture boundary detection using the proposed algorithm. All input images have 4 textures with 256×256 pixels.

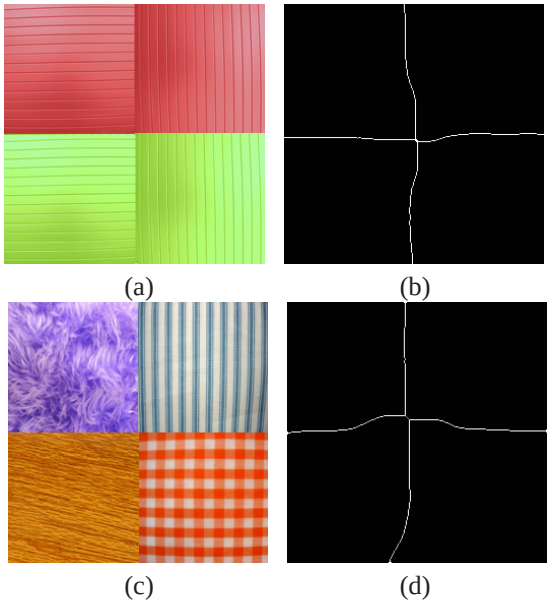


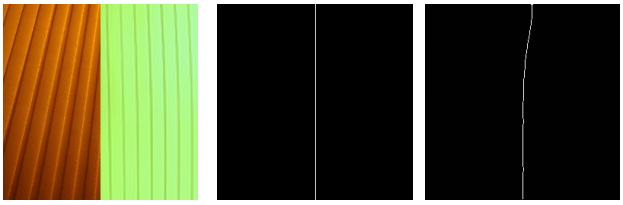
Figure 3: Simulation results for synthetic color texture input image using the proposed algorithm. (a) Input image. (b) Detection result for Fig. 3(a). (c) Input image. (d) Detection result for Fig. 3(c).

Figures 3(a) and (c) show the input images which is the synthetic color texture. Figures 3(b) and (d) show results of the texture boundary detection using the proposed algorithm. In Fig. 3(b), the boundaries of each texture can be detected well. On the other hands, in Fig. 3(d), we think that it is difficult to detect the accurate boundaries of the some textures including the nonperiodic feature.

4. Evaluation of detection performance

In this section, we evaluate the performance of texture boundary detection using the proposed algorithm. The per-

formance of the texture boundary detection depends on the feature of the texture as the periodicity. Therefore, we focus on three types of synthetic texture as "Uniform Texture", "Nonuniform Texture" and "Combining Texture" in this evaluation. "Uniform Texture" is the image combining the periodic textures. "Nonuniform Texture" is the image combining the nonperiodic textures. "Combining Texture" is the image combining the periodic texture and nonperiodic texture. Hence, each performance of the texture boundary detection using the three types of synthetic texture is compared. In this evaluation, we consider only the synthetic texture composed of two textures because it is hard to judge the accuracy of the texture containing multi boundaries.



(a) (b) (c)

Figure 4: Example of "Uniform Texture". (a) Input image. (b) Ideal output image. (c) Detection result for Fig. 4(a).

Figure 4(a) shows "Uniform Texture" which is the input image containing the two textures. Figures 4(b) and (c) show the ideal boundary and the result of the boundary detection using the proposed algorithm, respectively.

The error value between the ideal boundary and the detected boundary is calculated by

$$Error = \frac{\sum_{i=0}^{255} D_i \ /256}{128} \times 100, \tag{13}$$

where D_i is the distance between the horizontal positions of the ideal boundary Ans_i and of the detected boundary De_i calculated by

$$D_i\ [pixels] \ = \ |Ans_i - De_i|, \quad (i = 0, 1 \cdots, 255). \tag{14}$$

For this evaluation, we prepare 10 input images per synthetic texture. All the original texture images used in the synthetic texture are found in [7] and [8].

Table 1 shows the error evaluation of the three types of synthetic texture. In the case of "Uniform Texture", all error values are the smallest value among the three types of synthetic texture. On the other hands, in the case of "Nonuniform Texture", all error values are the biggest value. However, we think that the minimum error is sufficiently-small although it is difficult to detect the boundary of two textures including the nonperiodic feature. Because the color information is considered in the proposed algorithm. Namely, if the two textures have the different colors, the boundary of the two textures including the nonperiodic feature can be detected well. In the case of "Combining Texture", the all error values are middle between the case of "Uniform Texture" and of "Nonuniform Texture". In other words, if one of the two textures has the periodic feature, the performance of the texture boundary detection is better than the case of "Nonuniform Texture". From the comparison of the three types of synthetic texture, we think that these differences of the performance may occur in the case of the human visual system. Therefore, we feel that the detected boundaries using the proposed algorithm are similar to the human's sensation.

Table 1: Error Evaluation

	Ave. [%]	Min. [%]	Max. [%]
Uniform Texture	4.0	0.5	7.7
Nonuniform Texture	15.1	5.2	36.3
Combining Texture	8.3	2.4	13.2
	8.2	0.5	36.3

5. Conclusions

In this study, we have presented a color texture boundary detection using cellular neural networks. The proposed CNN is inspired from the early vision system of the human brain and can create two features at the same time like the early vision system. From our simulations, we have confirmed that boundaries can be detected by using the two features like the human's sensation. However, in the early vision system, there are higher order features that can be utilized. Therefore, we consider that the proposed algorithm can be extended by considering the higher order features.

References

[1] L.O. Chua and L. Yang, "Cellular Neural Networks:Theory," IEEE Trans. Circuits Syst., vol. 32, pp. 1257-1272, Oct. 1988.

[2] F. Dirk and T. Ronald, "Coding of Binary Image Data using Cellular Neural Networks and Iterative Annealing," Proc. of ECCTD'03, vol. 1, pp. 229-232, Sep. 2003.

[3] A. K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," Pattern. Recogn., vol. 24, pp. 1167-1186, 1991.

[4] K. Obermayer and G. G. Blasde "Geometry of Orientation and Ocular Dominance Columns in Monkey Striate Cortex" J. Comput. Neurol., vol. 13, pp. 4114-4129, Oct. 1993.

[5] C.T. Lin, C.H. Huang and S.A. Chen, "CNN-Based Hybrid-Order Texture Segregation as Early Vision Processing and Its Implementation on CNN-UM," IEEE Trans. Circuits Syst., vol. 54, no. 10, pp. 2277-2287, Oct. 2007.

[6] Cellular Sensory Wave Computers Laboratory Computer and Automation Research Institute Hungarian Academy of Sciences, "Cellular wave Computing Library (Template, Algorithms, and Programs) Version 2.1"

[7] http://textureking.com/

[8] http://textures.forrest.cz/