ISSN: 1978 - 8282

# EDGE DETECTION USING CELLULAR NEURAL NETWORK AND TEMPLATE OPTIMIZATION

Widodo Budiharto<sup>1</sup> Djoko Purwanto<sup>2</sup> Mauridhi Hery Purnomo<sup>3</sup>

E-mail: widodo@widodo.com, djoko@ee.its.ac.id, hery@ee.its.ac.id

Diterima:12 Juli 2010/Disetujui:23 Juli 2010

#### ABSTRACT

Result of edge detection using CNN could be not optimal, because the optimal result is based on template applied to the images. During the first years after the introduction of the CNN, many templates were designed by cut and try techniques. Today, several methods are available for generating CNN templates or algorithms. In this paper, we presented a method to make the optimal result of edge detection by using TEMPO (Template Optimization). Result shown that template optimization improves the image quality of the edges and noise are reduced. Simulation for edge detection uses CANDY Simulator, then we implementing the program and optimized template using MATLAB. Comparing to Canny and Sobel operators, image shapes result from CNN edge detector also show more realistic and effective to user.

Keywords: CNN, edge detection, TEMPO, Template optimization.

#### I. INTRODUCTION

A cellular neural network (CNN) is a 2 dimensional rectangular structure, composed by identical analogical non-linear processors, named cells [1]. CNN can be used in many scientific applications, such as in signal processing, image processing and

- 1. Dosen Program Studi Teknik Elektro Institute Teknology Surabaya Jl. Raya ITS, Sukolilo, Surabaya 60111, Indonesia
- Kaya 115, Sukolilo, Surabaya 00111, Indonesia
   Dosen Program Studi Teknik Elektro Institute Teknology Surabaya
- J. Raya ITS, Sukolilo, Surabaya 60111, Indonesia
   3. Dosen Program Studi Teknik Elektro Institute Teknology Surabaya
- 3. Dosen Program Studi Teknik Elektro Institute Teknology Surabaya Jl. Raya ITS, Sukolilo, Surabaya 60111, Indonesia

Vol.4 No.1 - September 2010

analyzing 3D complex surfaces [9]. In this paper, we implement edge detection program based on CNN and optimized using TEMPO provided by CNN Simulator called CANDY Simulator [7].

The basic circuit unit of CNNs contains linear and nonlinear circuit elements, which typically are linear capacitors, linear resistors, linear and nonlinear controlled sources, and independent sources.

The structure of cellular neural networks is similar to that found in cellular automata; namely, any cell in a cellular neural network is connected only to its neighbor cells. All the cells of a CNN have the same circuit structure and element values. Theoretically, we can define a cellular neural network of any dimension, but in this paper we will focus our attention on the two dimensional for image processing. A typical circuit of a single cell is shown in the figure 1 below.



Each cell contains one independent voltage source  $E u_{ij}$  (Input), one independent current source I (Bias), several voltage controlled current sources  $I_n uij, I_n yij$ , and one voltage controlled voltage source  $E_{yij}$  (Output). The controlled current sources  $I_n uij$  are coupled to neighbor cells via the control input voltage of each neighbor cell. Similarly, the controlled current sources  $I_n yij$  are coupled to their neighbor cells via the feedback from the output voltage of each neighbor cell [2].

The CNN allows fully parallel image processing, a given processing being executed simultaneously for the entire image. An example of 2 dimensional cellular neural networks is shown in Fig 2:

102

Vol.4 No.1 - September 2010

ISSN: 1978 - 8282



Figure 2. A two-dimensional cellular neural network. This circuit size is 4x4.

The squares are the circuit units called cells. The links between the cells indicate that there are interactions between the linked cells [2].

The state equation of a cell is [2]:

$$s_{ij} = -\kappa_{ij} + \sum_{e \mid k, \forall a_{ij}^{e} \mid (j)} A(i, j) k, \forall y_{ki} + \sum_{e \mid k, \forall a_{ij}^{e} \mid (ij)} B(i, j) k, \forall u_{ki} + z_{ij}$$
(1)

Where  $x_{ij}$  represent the state,  $u_{kl}$  is the input,  $y_{kl}$  is the output and  $z_{ij}$  is the threshold of the cell C(i,j). A(i,j;k,l) is the feedback operator and B (i,j;k,l) is the input synaptic operator. The ensemble (A,B,z) is named template.  $C_{lk}$  are the cells from a r-order neighborhood S<sub>r</sub> of the cell (i,j).

$$y_{ij} = f(x_{ij}) = \frac{1}{2} |x_{ij} + 1| - \frac{1}{2} |x_{ij} - 1|$$
(2)



Figure 3. Signal flow structure of a CNN with a 3x3 neighborhood.

Vol.4 No.1 - September 2010

The system structure of a center cell is represented in Figure 4:

Figure 4. Structure for cell  $C_{ij}$ , arrow printed in bold mark parallel data path from the input and the output of the surround cell  $u_{kl}$  and  $y_{kl}$ . Arrows with thinner lines denote the threshold, input, state and output, z,  $u_{ij}$ ,  $x_{ii}$  and  $y_{ij}$  respectively.

### **II. LITERATURES**

#### **Edge Detection**

In general, edge detection defines as boundary between 2 region (two adjacent pixel) that have very high different intensity [3]. Some of the others operator are Sobel, Prewitt, Roberts and Canny [3]. In this research we will compare the result with Sobel and Canny edge detector as another important methods [6].

## **EDGEGRAY CNN**

We use EDGEGRAY CNN for edge detection gray scale input images that accepting gray-scale input images and always converging to a binary output image. One application of this CNN template is to convert gray –scale images into binary images, which can then be used as inputs to many image-processing tasks which require a binary input image. Here is gray scale edge detection template with z/bias used are -0.5:

Tabi	le 1	:	Templ	late foi	r gray	scale	e ed	'ge a	letection
------	------	---	-------	----------	--------	-------	------	-------	-----------

A=	0	0	0	B=	-1	-1	-1	a foica
	0	2	0		-1	8	-1	= 0.5
	0	0	0		-1	-1	-1	0.5

#### **III. SYSTEM ARCHITECTURE**

We use MATLAB and webcam for capturing images, and CANDY (CNN Simulator) [7] for testing the color images edge detection. For optimizing template, we use TEMPO provided by CANDY. Diagram block that show the order of the process of the system shown in figure below :

Vol.4 No.1 - September 2010



Figure 5. Diagram block of Edge Detection using CNN using template optimization

First, the original edgegray template given to TEMPO program. Using some features of its program, we can optimizing the template. As the result, template below show optimized template for edgegray edge detection :

neighborhood: 1 feedback: 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 control: -1.0000 -1.0000 -1.0000 -1.0000 8.0000 -1.0000 -1.0000 -1.0000 -1.0000 current: -1.0000 Then, the script below show program generated by TEMPO to optimize the edgegray detection using CNN by modifying Iteration. {START: temrun} Initialize SIMCNN CSD TimeStep 0.100000 IterNum 80 OutputSampling 1 **Boundary ZEROFLUX** SendTo INPUT PicFill STATE 0.0 Vol.4 No.1 - September 2010

ISSN: 1978 - 8282

TemplatePath C:\Candy\temlib\ CommunicationPath C:\CANDY\ TemLoad o\_edgegray.tem RunTem

Terminate {STOP: temrun}

The template value and script above will be used by CANDY for simulation.

# **IV. RESULTS**

In this section the experimental results obtained by CANDY and MATLAB are presented. Let us consider an image figure 6,



Figure 6. Edge detection using CANDY Simulator

Figure above is the result of edge detection without template optimization, it shown that many noises arised (see image detail below).

106

Vol.4 No.1 - September 2010



Figure 7. Result of edge detection without template optimization

Using template and script above, we try to implement edge detection based on CNN using CANDY, the result shown below:



Figure 8. Result of edge detection with template optimization

Figure above shows that using template optimization, some noises are reduced. This indicated that template optimization succesfully implemented. Figure below is a program developed by MATLAB to implementing the edge detection using CNN and template optimization.

Vol.4 No.1 - September 2010



Figure 9. Implementation using MATLAB for template optimization.

To evaluate the effectiveness of CNN to any operators, we Compared to Sobel and Canny operator, from the figure below, indicated that image processed using CNN edge detector show more realistic and easy to understand.



Figure 10. Original image (a), Comparing to Sobel (c) and Canny edge detector (d), CNN edge detector with z=0.8 optimized with closing operation (b) show more realistic to user.

Vol.4 No.1 - September 2010

# V. CONCLUSION

In this paper, we have investigated the implementation of CNN and template optimization for edge detection. Based on the experiment, template optimization proved able to improves the quality of images for edge detection. Template optimization also reduced noises, but it makes some important lines disconnected. To solve this problem, we can use closing operation.

# **VI. FUTURE WORK**

CNN very important method for image processing. We propose this system can be used for system that need high speed image processing such as robotics system for tracking object and image processing in medical application. We will continue working on CNN for development of high speed image tracking in servant robot.

# .VII. REFERENCES

- [1] Chua LO, Yang L, "Cellular Neural Networks: Theory", IEEE Transactions on Circuit and System, vol 35, 1998, pp.1257-72.
- [2] Chua LO, Roska T, Cellular Neural Networks and Visual Computing, Cambridge University Press, 2002.
- [3] Gonzales, Rafael C. and Richard E. Woods. Digital Image Processing. 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.
- [4] Alper Basturk and Enis Gunay, "Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm", Expert System with Applciation 35, 2009, pp 2645-2650.
- [5] Koki Nishizono and Yoshifumi Nishio, "Image Processing of Gray Scale Images by Fuzzy Cellular Neural Network", International workshop on Nonlinear Circuits and Signal Processing, Hawaii, USA, 2006.
- [6] Febriani, Lusiana, "Analisis Penelusuran Tepi Citra menggunakan detector tepi Sobel dan Canny", Proceeding National Seminar on Computer and Intelligent System, University of Gunadarma, 2008.
- [7] CANDY Simulator, www.cnn-technology.itk.ppke.hu\
- [8] CadetWin CNN applicaton development \ environment and toolkit under Windows. Version 3.0, Analogical and Neural Computing Laboratory, Hungarian Academy of Sciences, Budapest, 1999.
- [9] Yoshida, T. Kawata, J. Tada, T. Ushida, A. Morimoto, Edge Detection Method with CNN, SICE 2004 Annual Conference, 2004, pp.1721-1724.

Vol.4 No.1 - September 2010