ISSN No. 0976-5697

Volume 3, No. 7, Nov-Dec 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Multi – Layer Raster Simulation for Color Image Processing Using CNN's Cloning Templates

R. Parimala
Project Consultant
InfoTech Technologies, Coimbatore
parimalaa6@gmail.com

K. Anuradha*
Ph.D Research Scholar,
Karpagam University, Coimbatore
k_anur@yahoo.com

Dr.K.Sankaranarayanan
Dean, Electrical Sciences
Easa College of Engineering and Technology, Coimbatore
kkd_sankar@yahoo.com

Abstract: This paper presents a raster simulator software capable of performing Image Processing applications using Cellular Neural Network (CNN). The CNN paradigm has rapidly evolved to cover a wide range of applications. The basic structure simulator is based on high – performance software capable of efficiently dealing with large images in the order of 10^5 pixels. The simulator operates in sequential mode. This provides an added flexibility to create individual templates that can be applied on single color layer to obtain full mix of colors. To perform any kind of color image processing, a color model is selected. Two color mapping schemes are used that effectively assign states to distinct color hues. Image processing with CNN may not always yield the desired visual results and post processing becomes necessary to enhance the visualization of the image. The simulator runs in an X-windows environment and uses standard graphics format as input.

Keywords: Cellular Neural Networks, X – Windows environment, Color Mapping, Raster simulator, Visualization.

I. INTRODUCTION

Cellular neural networks (CNN) [1] are a parallel computing paradigm similar to neural networks, with the difference that communication is allowed between neighboring units only. CNN is a multidimensional array of mainly identical dynamical systems called cells, which satisfies two properties [15]. 1) Most interactions are local within a finite radius T, and 2) All state variables are continuous valued signals. Simulation strategies based on Cellular Neural Network (CNN) multi-layer architectures are an ideal vehicle for color image processing. This is because each primary color is assigned to a unique layer. Each pixel's color can be handled as a triplet red, green and blue (RGB) whose combinations yield a secondary color [2]. Then it is possible to allocate a layer of CNN cells to each primary color component, to carry the processing independently, and then the triplet to give the results. In this case, it would be necessary to have an output function that would be based on Euclidean distance, or any other method of the three RGB values. It is a fact that the local interconnectivity properties of CNN make it very attractive for very large scale integration (VLSI) implementations. The basic structure simulator is based on a high – performance software capable of efficiently dealing with large images in the order of 105 pixels. The simulator operates in sequential mode. This provides an added flexibility to create individual templates that can be applied on single color layer to obtain full mix of colors.

In Section II, a related work is carried out. In Section III modules were designed. Section IV, V and VI deals with Results, Conclusion and References respectively.

II. RELATED WORK

V. Murugesh et al [3] presented a simulator for Cellular Neural Networks This simulator is capable of performing Multi-Layer Raster Simulation for any size of input image, thus a powerful tool for researchers investigating potential applications of CNN. This study reports an efficient algorithm exploiting the latency properties of Cellular Neural Networks along with popular numerical integration techniques; simulation results and comparison are also presented [3].

III. CELLULAR NEURAL NETWORK

The basic circuit unit of CNN is called a cell. It contains linear and nonlinear circuit elements. Any cell, C(i,j), is connected only to its neighbor cells i.e. adjacent cells interact directly with each other. This intuitive concept is called neighborhood and is denoted as N(i,j).

In multilayer CNN structure, each primary color is assigned to a layer of CNN. The color information is obtained from the Cell's state rather than from its output [4]. The local interconnectivity properties of CNN make it very attractive for VLSI implementation. A pixel's value is calculated based only on its neighbor pixels. Linear single step algorithms are used to filter pixel values. Designing color mapping techniques to assign states to distinct color hues. Designing templates to obtain difference effects on the input images.

A. Array Structure of CNN:

Cellular automata are automata [5] defined on the cellular space whose transition function is translation invariant:

$$f_j=f \text{ for any } Z \varepsilon Z^d \text{ with } f:Q^{|V|}Q, \qquad (1)$$

where Q is the set of states, Z is lattice and d is the dimension of lattices.

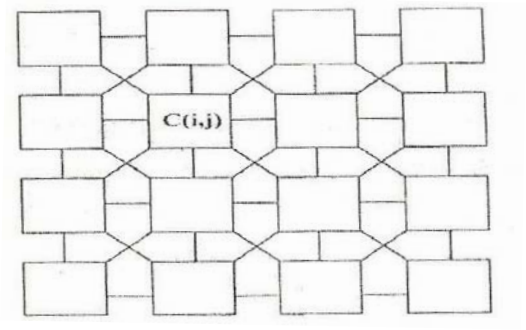

Figure 1. Array Structure of CNN

Consider an M x N CNN having M x N cells arranged in M rows and N columns as shown in Figure 1.

The basic unit of CNN is called a cell. Any cell on the ith row and jth column C(i,j) is connect only to its neighbor cells, i.e adjacent cells interact directly with each other. This neighborhood is denoted as N(i,j).

IV. MODULES

The following are the modules which are used to implement the algorithm designed for system "multi layer raster simulation for color image processing using CNN's cloning templates.

a. Color Mapping module
b. Designing templates module
c. Color Processing Module
d. Display Module.

The primary objective of the system is to perform color effects on the input images. The input images are stored in the database. When entering on the system, the image file name which one is to be processed is given as input. For the input image, the color mapping is done using any one of the mapping method. If in need, the pixel's value transformation is done using single step non linear filter procedure. Then any one of the color process is selected. According to that process, the CNN cloning templates are designed. The modified image is then stored into a new file. The required output image is displayed by using display module.

A. Color mapping techniques:

In color image processing, to handle a full range of color tone, two novel colors mapping schemes are derived. The two mapping schemes are such that a continuous mode and quantization mode. These mapping techniques are used to yield secondary colors by combining the primary color components.

a. Continuous mode mapping:

For the continuous mode mapping a linear transformation is employed. This transformation can be characterized by the function C:IR-> {-1,1}. In other words a set of real numbers is mapped to numbers bounded between (-1,1). A bound mapping can be applied to the bounded numbers so that their values can be scaled to the appropriate color range value between 0 and 255.

b. Quantized Mode mapping:

The quantized mode mapping is based on a linear mapping using the maximum and minimum layer colors as bounds to generate a discrete range of colors [6].

a) RGB Value: The Figure 2 displays the graphs of color mapping techniques. The plots show RGB values versus state values

The horizontal axis represents the State value and the vertical axis represents the RGB values. It can be seen that the continuous mapping approach presents a fairly good distribution of RGB values. The quantized mapping scheme is characterized by the abrupt step between state values of (-1, 1). Notice that for this approach most of the RGB values are concentrated in the upper and lower range [7]. This results in the desired limited set of colors that can be displayed.

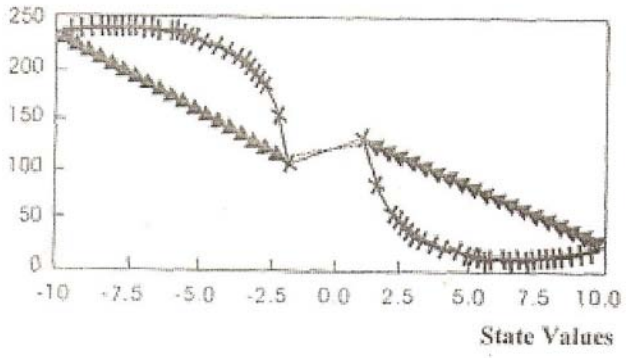

Figure 2.Color mapping scheme

B. Designing Templates:

The various cloning templates are designed. They are Shadow Detector, Connected Component Detector and Hole – Filter.

a) CNN Cloning Template: Shadow Detector: A simple cloning template for CNN capable of detecting the "Shadow" cast by a two – dimensional image is described. For example consider a 64 x 64 bipolar image as input. Applying this input to an extremely simple CNN to be described below, we can obtain an image which is "shadow" of the input image when the object is illuminated by a "light source" coming from the right. The template in improving the recognition rate of handwritten character is given below:

Consider the CNN [1] defined by

$$C^{dVx}{}_{ij} / dt = -1/RxVx_{ij} + A*Vy_{ij} + B*Vu_{ij} + I$$

$$Vy_{ij} – ½(|vx_{ij} + 1| - |Vx_{ij} – 1|) = f(Vx_{ij}),\ 1<=i<=M,\ 1<=j<=N$$

where * denotes the two dimensional "convolution operator".

It is important to note that although each pixel receives the information only from its immediate neighborhood on the same row, the dynamics of the entire array is global nature. In other words, each pixel cannot determine output should be +1 or -1 by looking at only the input value of its immediate neighborhood.

b) CNN Cloning Template: Connected Component Detector: The simple cloning template for CNN that is capable of detecting the number of connected components of a vector in $(+1, -1)^N$ is described. By exploiting these unique capabilities, architecture for a handwritten character reorganization system is proposed [8]. A character recognition system is given below (Figure 3).

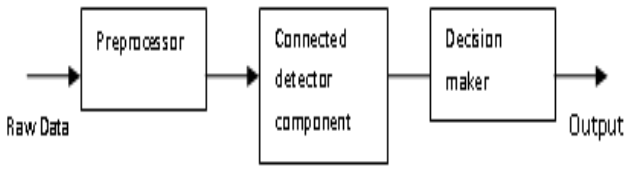

Figure 3. A character recognition system

The raw data for each handwritten numerals is supplied by a standard database. Since the raw data are ragged, a preprocessor is used to remove noise. The preprocessor is fed parallel into four CNN's. The compressed data was further reduced into an 83 dimensional vector and then fed into a decision making network.

c) CNN Cloning Template: Hole – Filter: The interest in the hole – filter comes from an effort to improve the performance of handwritten character reorganization. The Hole – Filter can probably be used for 2-D image reorganization problems. It is important to note that each pixel receives the information only from its immediate neighbors.

d) Image Thinning with a CNN: In Image processing, thinning refers to any system which transforms image into one-pixel-thick binary (0,1) or bipolar (±1) pattern while preserving the connectivity of the images.

The thinning problem is much more difficult than it looks. Basically two tasks must be implemented:

i. Peeling the thick pixels off.
ii. Stopping the peeling process when the pixel size reduces to exactly one.

The first part can be achieved with relative ease. The main difficulty lies in the second task, because the stopping decision must be done automatically. If the first task keeps on going, the patterns would disappear all together. In fact, there is a vast variety of works on the subject of thinning. All of them are digital and basically sequential. Our CNN thinning processor takes entirely different view point; it is analog and parallel.

C. Color processing:

The color processing module has the procedures to perform the following four kinds of color effects.

a. Color contrast effect.
b. Edge detection effect
c. CNN post processor effect
d. Monochrome to color effect.

RGB color model [9] is selected to perform color image processing using CNN multilayer structure. The RGB (Red, Green, Blue) coordinate system is commonly used for representing digital color images.

Using the RGB model, each primary color can be represented by a CNN layer, e.g red, green and blue layers Lr, Lg, Lb. Thus a simulation approach is to have a triplet <RGB> processed by a three layer CNN, with each layer processing a primary color. Following this idea, it is then possible to apply distinct templates to each layer. Therefore with the ability to process RGB separately plus the interlayer template effects, more complex image processing applications can be done. It is then possible to do edge detection in Lr and averaging in Lg, simultaneously.

To be able to work with multiple layers, the basic CNN equation (1) can be rapidly being expanded to a matrix equation of the following form:

$dX_{ij}(t)/dt=-X_{ij}(t) +\sum A(i,j;k,l)ykl(t)\ C(k,l)\ \varepsilon\ Nr\ (i,j) +\sum \underline{B}(i,j;k,l)ukl + 1$ ----------4

$Y_{ij}(t) = ½ (|X_{ij}(t) + 1|-|X_{ij}(t)-1|)$ ----------5

For simplicity the time iteration constant has been assumed to be unity.

In this late equation, instead of only one state variable per cell there are three state variables to be able to process color.

a. CNN Postprocessor:

Image processing with CNN may not always yield the desired visual results so postprocessing becomes necessary to enhance the vision of the image.

A CNN postprocessor [10] consists of a compiler capable of handling logical pixelwise operations among distinct color layers. This compiler follows the trends of having CNN as an analogic microprocessor. The added capability allows us to create new processed images with, for example, one layer processed by CNN and the remaining layers logically manipulated between CNN results and the original image. The syntax of the post processing language uses a Backus-Naur form notation.

D. Display:

All the files to be processed must be specified at the beginning of the program. When a file is read, the program splits the pixel information into its basic RGB components [11]. This strategy is used to create three unique layers that contain the color coded information of the image. The logical pixel wise operations include the conventional NOT, OR, AND, XOR, shift-left, shift – right functions. Operation can be performed on the layers of a file or a variable but most always be stored in a variable. The only three valid layers are assigned to a triplet <RGB> and are specified by means of keywords. Every variable's layer is initialized to "black" when first used. Finally, the new processed image is spooled out. It is required to specify the name of output and the variable containing the image to be printed.

V. RESULTS

The primary objective of the system is to perform color effects on the input images. The input images are stored in the backend tool MS – SQL Server. When entering into the system, the image file name which is to be processed is given as input. (Figure 4).

On the input image, the color mapping is done using any one of the mapping method. (Figure 5) Pixel's value transformation is done using single step nonlinear filter procedure. Then any one of the color process is selected. The color processed image shown in Figure 6 and in Figure 7. According to that process, the CNN cloning templates are designed (Figure 8). The modified image is then stored into a new file. The required output is displayed using the display module.

Figure 4 Input Image

Figure 5 After mapping

Figure 6 Color processed Image

Figure 7 Color processed Image

Figure 7

Figure 8 CNN Cloning templates

VI. CONCLUSION

When low – level hardware simulations of CNN are very costly for exploring new applications, the use of a behavioral simulator becomes indispensable. The system hereby presented allows exploring new color image processing applications in short turn around times. Processing with CNN is very attractive because the continuous transition from state to state shows the evolution of the image to its final appearance.

Although the systematic development of programming templates is still an attractive area of research, the procedure to process an image using only two templates is appealing for its simplicity. The work hereby introduced advances the state of the art of the cell's state rather than from its binary limited output. It is possible to obtain full gamut of color tones. Two color mapping schemes are introduced that effectively assign states to distinct color hues. The error produced by these schemes is minimum. Therefore they are seemed to be suitable for CNN color simulations. From the output presented, the robustness of the software and vast potential of CNN can be seen. This software uses only three layers of CNN's multilayer structure to assign the three primary colors in RGB color model. It may be enhanced to apply any no. of colors for any color model.

VII. REFERENCES

[1] C.C. Lee, J. Pineda de Gyvez, "Single layer CNN simulator " in Proc, IEEE. Symp. Circuits Syst., 1994.

[2] Chi – Chien Lee, Jose Pineda de Gyvez, "Color image processing in a Cellular Neural Network Environment", IEEE Transaction on Neural Networks, Vol.7, Issue 5, 1086 – 1098..

[3] V. Murugesh, N. Rengarajan, "An efficient numerical integration technique for Multi – Layer Raster CNN Simulator", Information Technology Journal, Vol. 6, Issue 2, pp 202 – 206, 2007.

[4] Droblas Greenberg, Seth Greenberg, "Digital Images: A Practional guide".

[5] J.A. Nossek, T.Roska, Guest Eds., "Special Issue on cellular neural networks", in IEEE Transactions on Circuit Systems.

[6] Jeffery J. Rodriguez, Christopher C. Yang, "Effects of Luminance quantization error on Color Image Processing", IEEE Transactions on Image Processing, Vol 3, No.6, November 1994.

[7] K.R. Crounse, T.Roska, L.O. Chuo, "Color Image Processing with Cellular neural networks", IEEE Transaction on Circuits Systems., Vol 40, pp. 267 – 283, April 1993.

[8] L.O. Chuo, L. Yand, "Cellular Neural Networks: Theory", IEEE Transaction on Circuits Systems.

[9] K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall of India, 1989.

[10] L.O. Chuo, T. Roska, "The CNN universal machine, Part I: The architecture", in the IEEE Proceedings International workshop on Cellular Neural Networks and its Applications, 1992, pp 1 – 10.

[11] M. Egmont – Petersen. D.de Ridder, H.Handels, "Image Processing with neural networks – a review", Pattern Recognition, Vol. 35, No.10, pp. 2279 – 2301, 2002.