MULTILAYER RASTER CNN SIMULATION BY ARITHMETIC AND CENTROIDAL MEAN RKACeM(4,4)

R.Ponalagusamy and S.Senthilkumar

Department of Mathematics, National Institute of Technology,
Tiruchirappalli-620 015, Tamilnadu, INDIA
E-mail: rpalagu@nitt.edu,ssenthilkumar@yahoo.co.in

Abstract: *In this article a versatile algorithm for simulating CNN arrays is implemented using various means. The function of the simulator is that it is capable of performing Raster Simulation for any kind as well as any size of input image. It is a powerful tool for researchers to examine the potential applications of CNN. This article proposes an efficient pseudo code for exploiting the latency properties of Cellular Neural Networks along with well known RK-Fourth Order Embedded numerical integration algorithms. Simulation results and comparison have also been presented to show the efficiency of the Numerical integration Algorithms. It is found that the RK-Embedded Centroidal Mean outperforms well in comparison with the RK-Embedded Harmonic Mean and RK-Embedded Contra-Harmonic Mean.*

Keywords: *Cellular Neural Networks, Various Means, Edge Detection, Raster CNN Simulation.*

1. INTRODUCTION

The distinctiveness of Cellular Neural Networks (CNNs) are analog time-continuous, non-linear dynamical systems and formally belong to the class of recurrent neural networks. CNNs have been proposed by Chua and Yang [l,2], and they have found that CNN has many significant applications in signal and real-time image processing. Roska et al. [3] have presented the first extensively used simulation system which allows the simulation of a large class of CNN and is especially suited for image processing applications. It also includes signal processing, pattern recognition and solving ordinary and partial differential equations etc.

Evans et al. [4] introduced embedded Centroidal Mean, Yaacob and Sanugi [5] adapted embedded Harmonic mean and Yaakub and Evans [6] have presented the Contra-Harmonic Mean. In this article, the time-multiplexing CNN simulation problem is solved with different approach using the algorithms say Embedded Centroidal Mean, Embedded Harmonic Mean and Embedded Contra-Harmonic Mean.

2. FUNCTIONS OF CELLULAR NERURAL NETWORK

The general CNN architecture consists of M*N cells placed in a rectangular array. The basic circuit unit of CNN is called a cell. It has linear and nonlinear circuit elements. Any cell, C(i,j), is connected only to its neighbor cells (adjacent cells interact directly with each other). This intuitive concept is known as neighborhood and is denoted by N(i,j). Cells not in the immediate neighborhood have indirect effect because of the propagation effects of the dynamics of the network. Each cell has a state x, input u, and output y. For all time t > 0, the state of each cell is said to be bounded and after the transient has settled down, a cellular neural network always approaches one of its stable equilibrium points. It implies that the circuit will not oscillate. The dynamics of a CNN has both output feedback (A) and input control (B) mechanisms. The dynamics of a CNN network cell is governed by the first order nonlinear differential equation given below:

$$c\frac{dx_{ij}(t)}{dt} = \frac{-1}{R}x_{ij}(t) + \sum_{c(k,l)\in N(i,j)} A(i,j;k,l)y_{kj}(t) + \sum_{c(k,l)\in N(i,j)} B(i,j;k,l)u_{kl}(t) + I, \quad 1 \le i \le M; 1 \le j \le N. \quad (1)$$

and the output equation is given by,

$$y_{ij}(t) = \frac{1}{2}\left[\left|x_{ij}(t)+1\right| - \left|x_{ij}(t)-1\right|\right], \quad 1 \le i \le M;\ 1 \le j \le N.$$

where C is a linear capacitor, x_{ij} denotes the state of cell C(i,j), $x_{ij}(0)$ is the initial condition of the cell, R is a linear resistor, I is an independent current source, $A(i,j;k,l)y_{kl}$ and $B(i,j;k,l)u_{kl}$ are voltage controlled current sources for all cells C(k,l) in the neighborhood N(i,j) of cell C(i,j), and y_{ij} represents the output equation.

For simulation purposes, a discretized form of equ. (1) is solved within each cell to simulate its state dynamics. One common way of processing a large complex image is using a raster approach [1]. This approach implies that each pixel of the image is mapped onto a CNN processor. That is, it has an image processing function in the spatial domain that is expressed as:

$$g(x,y) = T(f(x,y)) \quad (2)$$

where g(.) the processed image, f(.) is the input image, and T is an operator on f(.) defined over the neighborhood of (x,y). It is an exhaustive process from the view of hardware implementation. For practical applications, in the order of 250,000 pixels, the hardware would require a large amount of processors which would make its implementation unfeasible. An different option to this scenario is multiplex the image processing operator.

3. PERFORMANCE OF RASTER CNN SIMULATIONS

Raster CNN simulation is an image scanning-processing technique for solving the system of difference equations of CNN. The equation (1) is space invariant, which means that A (i,j;k,l) = A(i-k,j-1) and B(i,j;k,l) = B(i-k,j-1) for all i,j,k,l. Therefore, the solution of the system of difference equations can be seen as a convolution process between the image and the CNN processors. The basic approach is to imagine a square subimage area centered at (x,y), with the subimage being the same size of the templates involved in the simulation. The center of this subimage is then moved from pixel to pixel starting, say, at the top left corner and applying the A and B templates at each location (x,y) to solve the differential equation. This procedure is repeated for each time step, for all the pixels in the image. An instance of this image scanning-processing is referred to as “iteration”.

The processing stops when it is found that the states of all CNN processors have converged to steady-state values, and the outputs of its neighbor cells are saturated, e.g. they have $a \pm 1$ value [1,2]. This whole simulating approach is referred to as raster simulation. A simplified pseudo code is presented below gives the exact notion of this approach.

3.1 PSEUDO CODE FOR RASTER CNN SIMULATION

Step 1: Initially get the input image, initial conditions and templates from end user.
/* M,N = Number of rows and columns of the 2D image */
while (converged-cells < total number of cells)
{
for (i=l; i<=M; i++)
for (j=l; j<=N; j++)
{
if (convergence-flag[i] [j])
continue; /* current cell already converged*/

Step 2: /* Calculate the next state */

$$x_{ij}(t_{n+1}) = x_{ij}(t_n) + \int_{t_n}^{t_{n+1}} f'(x(t_n))dt$$

Step 3: /* Check the convergence criteria */

If $\left(\frac{dx_{ij}(t_n)}{dt}\right) = 0 \ and \ y_{kl} = \pm 1, \forall c(k,l) \in N_r(i,j)$
{
convergence-flag[i][j] = 1;
converged-cells++ ;
}
} /* end for */

Step 4: /* Update the state values of the entire Image */
for (i=l; i<=M; i++)
for (j=l; j<=N; j++)
{
if (convergence-flag[i][j]) continue;
$x_{ij}(t_n) = x_{ij}(t_{n+1})$;
}
Number of iteration++;
} /* end while */

4. NUMERICAL INTEGRATION TECHNIQUES

The CNN is described by a system of nonlinear differential equations. Therefore, it is necessary to discretize the differential equation for performing behavioral simulation. For computational purposes, a normalized time differential equation describing CNN is used by Nossek et al. [7]

$$f'(x(n\tau)) = \frac{dx_{ij}(n\tau)}{dt} = -x_{ij}(n\tau) + \sum_{c(k,l)\in N_r(i,j)} A(i,j;k,l) y_{kj}(n\tau) + \sum_{c(k,l)\in N_r(i,j)} B(i,j;k,l) u_{k,l} + I, 1 \le i \le M; 1 \le j \le N;$$

$$y_{ij}(n\tau) = \frac{1}{2}\left[\left|x_{ij}(n\tau)+1\right| - \left|x_{ij}(n\tau)-1\right|\right],$$

$$1 \le i \le M;\ 1 \le j\ N; \tag{5}$$

where τ is the normalized time. For the purpose of solving the initial-value problem, well established Single Step methods of numerical integration techniques are used in [8]. These methods can be derived using the definition of the definite integral

$$x_{ij}((n+1)\tau)-x_{ij}(n\tau)) = \int_{\tau_n}^{\tau_{n+1}} f'(x(n\tau))d(n\tau) \qquad (6)$$

Explicit Euler's, the Improved Euler Predictor-Corrector and the Fourth-Order (quartic) Runge-Kutta are the mostly widely used single step algorithm in the CNN behavioral raster simulation. Three types of numerical integration algorithms are used in raster CNN simulations. They are RK-Embedded Centroidal Mean is discussed by Evans et al. [4], RK-Embedded Harmonic Mean is discussed by Yaacub and Sanugi[5] and RK-Embedded Contra-Harmonic Mean is discussed by Yaakub and Evans[6]

4.1 EXPLICIT EULER'S ALGORITHM

Euler's method is the simplest of all algorithms for solving ordinary differential equations. It is an explicit formula which uses the Taylor-series expansion to calculate the approximation.

$$y_{n+1} = y_n + h \qquad (8)$$

4.2 RK-GILL ALGORITHM

The RK-Gill algorithm discussed by Oliveria [9] is an explicit method which requires the computation of four derivatives per time step. The increase of the state variable x^{ij} is stored in the constant k_1. This result is used in the next iteration for evaluating k_2 and repeat the same process to obtain the values of k_3 and k_4.

$$k_1 = f(y_n), \quad k_2 = f(y_n + \frac{k_1}{2})$$
$$k_3 = f(y_n + (\frac{1}{\sqrt{2}} - \frac{1}{2})k_1) + (1 - \frac{1}{\sqrt{2}})k_2) \qquad (9)$$
$$k_4 = f(y_n - \frac{1}{\sqrt{2}}k_2 + (1 + \frac{1}{\sqrt{2}})k_3)$$

Therefore, the final integration is a weighted sum of the four calculated derivates is given below.

$$y_{n+1} = -y_n + \frac{1}{6}[k_1 +$$
$$(2-\sqrt{2})k_2 + (2+\sqrt{2})k_3 + k_4]$$
(10)

5. FOURTH ORDER RK METHOD BASED ON EMBEDDED MEANS

5.1 RK-EMBEDDED CENTROIDAL MEAN

The Fourth Order RK-Embedded Centroidal Mean is given by,

$$k_1 = f(y_n), \quad k_2 = f(y_n + \frac{hk_1}{2}),$$

$$k_3 = f(y_n + \frac{hk_1}{2} + \frac{11hk_2}{24}),$$

$$k_4 = f(y_n + \frac{hk_1}{12} - \frac{25hk_2}{132} + \frac{73hk_3}{66}) \qquad (7)$$

Therefore, the final integration is a weighted sum of the four calculated derivates is given below.

$$y_{n+1} = y_n + \frac{h}{3}[\sum_{i=1}^{3} \frac{2}{3} \frac{k_i^2 + k_{i+1}^2 + k_i k_{i+1}}{k_i + k_{i+1}}] \qquad (8)$$

5.2 RK-EMBEDDED HARMONIC MEAN

The Fourth Order RK-Embedded Harmonic Mean is given by,

$$k_1 = f(y_n), \quad k_2 = f(y_n + \frac{hk_1}{2}),$$

$$k_3 = f(y_n - \frac{hk_1}{8} + \frac{5hk_2}{8}),$$

$$k_4 = f(y_n - \frac{hk_1}{4} + \frac{7hk_2}{20} + \frac{9hk_3}{10}) \qquad (9)$$

Therefore, the final integration is a weighted sum of the four calculated derivates is given below.

$$y_{n+1} = y_n + h[\frac{k_2}{6} + \frac{k_3}{6} + \frac{2}{3}(\frac{k_1k_2}{k_1+k_2}) + \frac{2}{3}(\frac{k_3k_4}{k_3+k_4})]$$
(10)

5.3 RK-EMBEDDED CONTRA- HARMONIC MEAN

The Fourth Order RK-Embedded Contra-Harmonic Mean is given by,

$$k_1 = f(y_n)\ ,\ k_2 = f(y_n + \frac{hk_1}{2})\ ,$$

$$k_3 = f(y_n + \frac{hk_1}{8} + \frac{3hk_2}{8})\ ,$$

$$k_4 = f(y_n + \frac{hk_1}{4} - \frac{3hk_2}{4} + \frac{3hk_3}{2}) \quad (11)$$

Therefore, the final integration is a weighted sum of the four calculated derivates is given below.

$$y_{n+1} = y_n + \frac{h}{3}[\frac{k_1^2 k_2^2}{k_1 + k_2} + \frac{k_2^2 k_3^2}{k_2 + k_3} + \frac{k_3^2 k_4^2}{k_3 + k_4}] \quad (12)$$

6. SIMULATION RESULTS AND COMPARISONS

All the simulated outputs presented below here are performed using a high power workstation, and the simulation time used for comparisons is the actual CPU time used. The input image format is the X windows bitmap format (xbm), which is commonly available and easily convertible from popular image formats like GIF or JPEG. Figs. 1(b), 2(b) and 3(b) show the results of the raster simulator obtained from a complex image of 1,25,600 pixels.

Using RK-Embedded Harmonic Mean, Embedded Contra-Harmonic Mean and RK-Embedded Centroidal Mean the results of the raster simulator obtained from a complex image of 1, 25,600 pixels are depicted respectively in Figs.1, 2, and 3. For the present example an averaging template followed by an Edge Detection template were applied to the original image to yield the images displayed in Fig. 1(b). The same procedure has been adapted for getting the results shown in Figs. 2(b) and 3(b). It is observed from Figs. 1(b), 2(b) and 3(b) that the edges obtained by the Embedded Centroidal Mean is better than that obtained by the Embedded Harmonic Mean, Embedded Contra-Harmonic Mean.

As speed is one of the major concerns in the simulation, determining the maximum step size that still yields convergence for a template can be helpful in speeding up the system. The speed-up can be achieved by selecting an appropriate (Δt) for that particular template. Even though the maximum step size may slightly vary from one image to another, the values in Fig. 4 show a comparison between three different templates. These results were obtained by trial and error over more than 100 simulations on a Coins figure.

It is observed from Fig. 4 that RK-Embedded Centroidal mean allows us to select a maximum step-size (Δt) as compared to other two methods irrespective of the selection of templates. Fig. 5 shows that the importance of selecting an appropriate time step size (Δt). If the step size is chosen is too small, it might take many iterations, hence longer time, to achieve convergence. But, on the other hand, if the step size taken is too large, it might not converge at all or it would be converges to erroneous steady state values. The results of Fig. 5 were obtained by simulating a small image of size 256×256 pixels using Averaging template on a Coins figure.

7. CONCLUSION

The attention of the present article is focussed on different numerical integration algorithms involved in the raster CNN simulation. The significance of the simulator is capable of performing raster simulation for any kind as well as any size of input image. It is a powerful tool for researchers to investigate the potential applications of CNN. It is pertinent to pin-point out here that the RK-Embedded Centroidal Mean guarantees the accuracy of the detected edges and greatly reduces the impact of random noise on the detection results in comparison with the RK-Embedded Harmonic Mean and Embedded Contra-Harmonic Mean. It is of interest to mention that using RK-Embedded Centroidal Mean; the edges of the output images are proved to be feasible and effective by theoretic analysis and simulation.

Fig. 1. (a) Original Coins Image;(b) After Averaging and Edge Detection Templates by employing RK-Embedded Centroidal Mean.

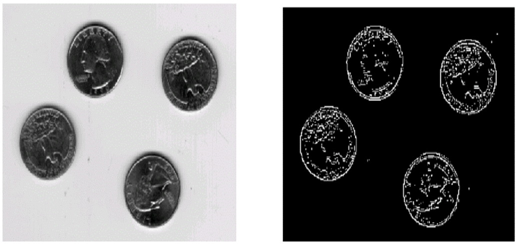

Fig. 2. (a) Original Coins Image;(b) After Averaging and Edge Detection Templates by adapting RK-Embedded Harmonic Mean.

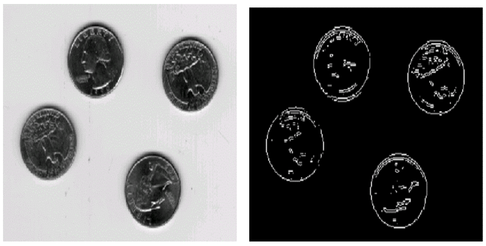

Fig. 3. (a) Original Coins Image;(b) After Averaging and Edge detection Templates by adapting RK-Embedded Contra-Harmonic Mean.

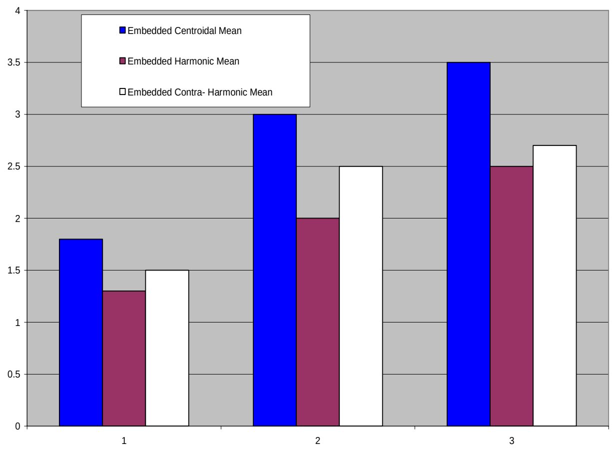

Fig. 4. Maximum Step Size (Δt) yields the convergence for three different templates.

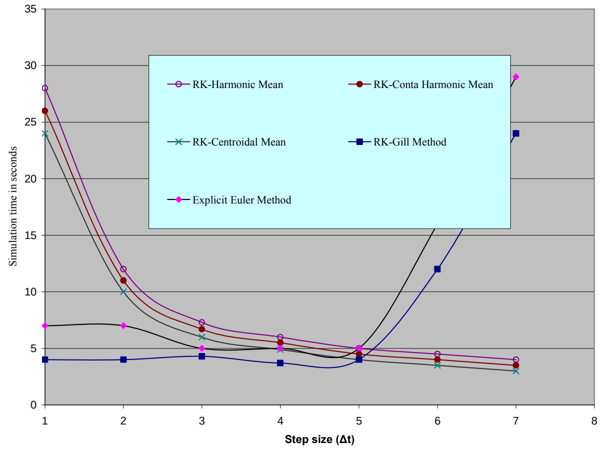

Fig. 5. Comparison of Five Numerical Integration Techniques using the Averaging Template.

REFERENCES

[1] Chua, L.O., Yang, L.: *Cellular Neural Networks: Theory*, IEEE Transactions on Circuits and Systems, 35, 1257–1272, 1988.

[2] Chua, L.O., Yang, L.: *Cellular Neural Networks: Applications*, IEEE Transactions on Circuits and Systems, 35, 1273–1290, 1998.

[3] Roska et al. *CNNM Users Guide*, Version 5.3x, Budapest (1994).

[4] K.Murugesan, D.Paul Dhayabaran,E.C.Henry Amirtharaj, and David J.Evans,"*A fourth order Embedded Runge-Kutta RKCeM(4,4) Method based on Arithmetic and Centrroidal Means with Error Control"* International Journal of Computer Mathematics,79(2),pp.247-269, 2002.

[5] Nazeerudin Yaacob and Bahrom Sanugi,"*A New Fourth-Order Embedded Method Based on the harmonic mean",Mathematika"*, Jilid,hml 1-6, 1998.

[6] A.R.Yaakub and D.J.Evans." *A Fourth order Runge-Kutta RK (4, 4) Method with Error Control"*,International Journal of Computer Mathematics, 71, pp.383-411, 1997.

[7] Nossek, J.A., Seiler, G., Roska, T., Chua, L.O.: *Cellular Neural Networks: Theory and Circuit Design*, Int. J. of Circuit Theory and Applications, 20,533-553, 1992.

[8] L.O. Chua and T. Roska, "*The CNN Universal Machine Part 1: The Architecture"*, in Int. Workshop on Cellular Neural Networks and their Applications (CNNA), 1-10, 1992.

[9] Oliveira, S.C.: Evaluation of effectiveness factor of immobilized enzymes using Runge-Kutta Gill method: how to solve mathematical undetermination at particle center point?, Bio Process Engineering, 20 (1999) 185-187.

Dr.R.PONALAGUSAMY was born in Annanji, Theni,Tamilnadu, India on 27th October,1958. He received his MSc in Applied Mathematics from Madurai Kamaraj University in 1982. He received his PhD in Computational Biomechanics from the Indian Institute of Technology, Bombay, in 1986. The Environmental Research Corporation Tokyo, Japan, awarded him a post-doctoral fellowship for two years at Chuo University, Tokyo. In 1989, he joined the National Institute of Technology, Tiruchirappalli, India, as a lecturer, and became Assistant Professor in 1996 in the Department of Mathematics. At present he is a Professor in the same department. His current research areas include computational experimentation on biofluid mechanics, plastic flow, finite element method, computer models on metal forming and powder metallurgical materials, wavelets, digital image processing, and development of parallel algorithms.

SUKUMAR SENTHILKUMAR was born in Neyveli Township, Cuddalore District, Tamilnadu, India on 18th July 1974. He received his B.Sc Degree in Mathematics from Madras University, M.Sc, Degree in Mathematics from Bharathidasan University in 1996, M.Phil Degree in Mathematics from Bharathidasan University in 1999 and M.Phil Computer Science & Engineering from Bharathiar University in 2000. Also, he received PGDCA and PGDCH in Computer Applications and Computer Hardware from Bharathidasan University in 1996 and 1997 respectively. He is Currently Pursuing Doctoral Degree in the Department of Mathematics at National Institute of Technology [REC], Tiruchirappalli, Tamilnadu, India. He served as a Lecturer, in the Department of Computer Science at Asan Memorial College of Arts and Science, Chennai, Tamilnadu, India. His Current Research Interests include Digital Image Processing and other related areas.