Investigation on Multilayer Raster Cellular Neural Network by Arithmetic and Heronian Mean RKAHeM(4,4)

R. Ponalagusamy and S. Senthilkumar

Abstract- **We introduce a new technique for solving Initial Value Problems (IVPs) by formulating an embedded method involving RK methods based on Arithmetic Mean(AM) and Heronian Mean (HeM). 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. By using the newly proposed embedded method, a versatile algorithm for simulating multilayer CNN arrays is implemented. This article proposes an efficient pseudo code for exploiting the latency properties of CNN 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 Heronian Mean outperforms well in comparison with the RK-Embedded Centroidal Mean, Harmonic Mean and Contra-Harmonic Mean.**

Index Terms- **Cellular Neural Networks, Various Embedded Means, Edge Detection, Multilayer Raster.**

I. INTRODUCTION

The uniqueness 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 important applications in signal and real-time image processing. Roska et al. [3] have presented the first widely 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. It is of interest to state that embedded methods are actually two methods built into one. The first method is of order p and the second has order p + 1. The difference between these methods provides an error estimate for the first method with order p.

Manuscript received on February 4, 2007. This work was fully supported as a part of Technical Quality Improvement Programme, Under the Ministry of Human Resource Development, sponsored by Govt. of India, National Institute of Technology, Trichirappalli-620 015, Tamilnadu, India.

R. Ponalagusamy and S. Senthilkumar are with the Department of Mathematics, National Institute of Technology (formerly Regional Engineering College), Tiruchirappalli-620 015, Tamilnadu, India. (Telephone No. +91-0431-2501801/10; Fax: +91-0431-2500133; URL :http://www.nitt.edu; E-mail:rpalagu@nitt.edu; ssenthilkumar1974@yahoo. co.in).

Error estimates by these methods have been derived by Merson [4], Fehlberg [5]. Evans and Yaakub [6,7] introduced a new embedded Runge-Kutta RK(4,4) method which is actually two different RK methods but of the same order p = 4. This embedded method has been developed using Runge-Kutta methods based on arithmetic mean (RKAM) and Contraharmonic Mean (RKCoM). Evans et. al [8] introduced embedded centroidal mean and Yaacob and Sanugi [9] adapted embedded harmonic mean.

Ponalagusamy and Senthilkumar [10] has discussed Sixth order RK-algorithm for raster CNN simulation. Chi-Chien Lee and Jose Pineda de Gyvez [11] introduced Euler, Improved Euler, Predictor-Corrector and Fourth-Order (quartic) Runge-Kutta algorithms in time-multiplexing CNN simulation. Ponalagusamy and Senthilkumar proposed a new fourth order Embedded RKAHeM (4,4) method in detail based on Runge-Kutta Arithmetic and Heronian Means [12]. In this article, the Raster CNN simulation problem is solved with different approach using presently developed new RK-Embedded Heronian Mean, Embedded Centroidal Mean, Embedded Harmonic Mean and Embedded Contra-Harmonic Mean.

II. THE EMBEDDED RKAHEM (4,4) METHOD

The fourth order Runge-Kutta RKHeM [13] is,

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau) + \frac{1}{9}[k_{1ij} + 2(k_{2ij} + k_{3ij}) + k_{4ij} + \sqrt{k_{1ij} + k_{2ij}} + \sqrt{k_{2ij} + k_{3ij}} + \sqrt{k_{3ij} + k_{4ij}}]$$

(or)

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau) + \frac{1}{9}[k_{1ij} + 2(k_{2ij} + k_{3ij}) + k_{4ij} + \sqrt{|k_{1ij} + k_{2ij}|} + \sqrt{|k_{2ij} + k_{3ij}|} + \sqrt{|k_{3ij} + k_{4ij}|}] \quad (1)$$

where

$$k_{1ij} = f'(x_{ij}(n\tau)),$$

$$k_{2ij} = f'(x_{ij}(n\tau) + \frac{1}{2}k_{1ij}),$$

$$k_{3ij} = f'(x_{ij}(n\tau) - \frac{1}{48}k_{1ij} + \frac{25}{48}k_{2ij}), \quad (2)$$

$$k_{4ij} = f'(x_{ij}(n\tau) - \frac{1}{24}k_{1ij} + \frac{47}{600}k_{2ij} + \frac{289}{300}k_{3ij})$$

Combination of RKAM and RKHeM is referred as RKAHeM (4,4), and can be formulated as [12],

$$k_{1ij} = f'(x_{ij}(n\tau)) = k^*_{1ij},$$
$$k_{2ij} = f'(x_{ij}(n\tau) + \frac{1}{2}k_{1ij}) = k^*_{2ij},$$
$$k_{3ij} = f'(x_{ij}(n\tau) + \frac{1}{2}k_{2ij}),\ k_{4ij} = f'(x_{ij}(n\tau) + k_{3ij}),$$
$$k_{3ij} = f'(x_{ij}(n\tau) - \frac{1}{48}k_{1ij} + \frac{25}{48}k_{2ij}) = k^*_{3ij},$$
$$k_{4ij} = f'(x_{ij}(n\tau) - \frac{1}{24}k_{1ij} + \frac{47}{600}k_{2ij} + \frac{289}{300}k_{3ij}) = k^*_{4ij}$$

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau) + \frac{1}{3}[\frac{k_{1ij}+k_{2ij}}{2} + \frac{k_{2ij}+k_{3ij}}{2} + \frac{k_{3ij}+k_{4ij}}{2}] \quad (3)$$

$$x^*_{ij}((n+1)\tau) = x_{ij}(n\tau) + \frac{1}{9}[k^*_{1ij} + 2(k^*_{2ij} + k^*_{3ij}) + k^*_{4ij} + \sqrt{|k^*_{1ij} + k^*_{2ij}|} + \sqrt{|k^*_{2ij} + k^*_{3ij}|} + \sqrt{|k^*_{3ij} + k^*_{4ij}|}]$$

III. STRUCTURE AND FUNCTIONS OF CELLULAR NEURAL 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:

$$c\frac{dx_{ij}(t)}{dt} = \frac{-1}{R}x_{ij}(t) + \sum_{c(k,l)\in N(i,j)} A(i,j;k,l)y_{kj}(t) + \sum_{c(k,l)\in N(i,j)} B(i,j;k,l)u_{kl}(t) + I, 1 \le \mathrm{i} \le \mathrm{M}; 1 \le \mathrm{j} \le \mathrm{N}. \quad (4)$$

and the output equation is given by,

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

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.

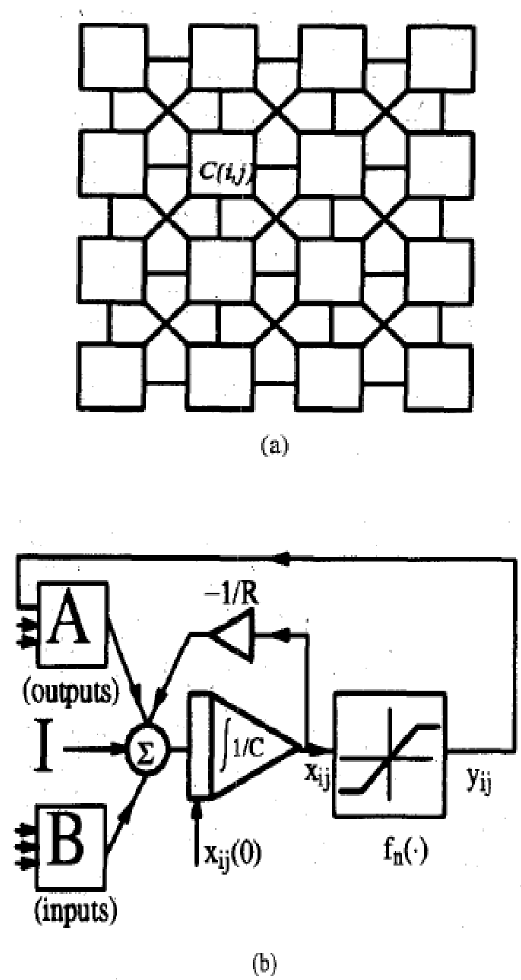


Fig. 1. Cellular Neural Networks: (a) Array Structure and (b) Block Diagram

For simulation purposes, a discretized form of equ. (4) is solved within each cell to simulate its state dynamics. One common way of processing a large complex image is using a raster approach [1,2]. 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:

$$g(x,y) = T(f(x,y)) \quad (5)$$

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.

IV. PERFORMANCE OF RASTER CNN SIMULATIONS

Raster CNN simulation is an image scanning-processing technique for solving the system of difference equations of CNN. The equ. (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],[14,15]. This whole simulating approach is referred to as raster simulation. A simplified pseudo code is presented below gives the exact notion of this approach.

A. *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 */
```

V. 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. [16].

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

$$y_{ij}(n\tau) = \frac{1}{2}\left[\left|x_{ij}(n\tau)+1\right| - \left|x_{ij}(n\tau)-1\right|\right], \quad 1 \le \mathrm{i} \le \mathrm{M};\ 1 \le \mathrm{j}\ \mathrm{N}; \tag{6}$$

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

$$x_{ij}((n+1)\tau) - x_{ij}(n\tau)) = \int_{\tau_n}^{\tau_{n+1}} f'(x(n\tau))d(n\tau) \tag{7}$$

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. These methods vary in the way they evaluate the integral presented in (4). In addition four types of numerical integration algorithms are also used in raster simulations. They are RK-Embedded Heronian Mean, Embedded Centroidal Mean, Embedded Harmonic Mean and Embedded Contra-Harmonic Mean.

A. 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.

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau) + \tau\ f'(x(n\tau)) \tag{8}$$

B. RK-Gill Algorithm

The RK-Gill algorithm discussed by Oliveria [17] 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_{1ij}. This result is used in the next iteration for evaluating k_{2ij} and repeat the same process to obtain the values of k_{3ij} and k_{4ij}.

$$\begin{aligned} k_{1ij} &= f'(x_{ij}(n\tau)),\ k_{2ij} = f'(x_{ij}(n\tau)+\frac{1}{2}k_{1ij}) \\ k_{3ij} &= f'(x_{ij}(n\tau)+(\frac{1}{\sqrt{2}}-\frac{1}{2})k_{1ij}+(1-\frac{1}{\sqrt{2}})k_{2ij}) \\ k_{4ij} &= f'(x_{ij}(n\tau)-\frac{1}{\sqrt{2}}k_{2ij}+(1+\frac{1}{\sqrt{2}})k_{3ij}) \end{aligned} \tag{9}$$

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

$$\begin{aligned} x_{ij}((n+1)\tau) &= x_{ij}(n\tau)+\frac{1}{6}[k_{1ij}+ \\ &(2-\sqrt{2})k_{2ij}+(2+\sqrt{2})k_{3ij}+k_{4ij}] \end{aligned} \tag{10}$$

VI. FOURTH ORDER RK METHOD BASED ON EMBEDDED MEANS

A. RK-Embedded Centroidal Mean

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

$$\begin{aligned} k_{1ij} &= f'(x_{ij}(n\tau))\ ,\ k_{2ij} = f'(x_{ij}(n\tau)+\frac{1}{2}k_{1ij}), \\ k_{3ij} &= f'(x_{ij}(n\tau)+\frac{1}{24}k_{1ij}+\frac{11}{24}k_{2ij}), \\ k_{4ij} &= f'(x_{ij}(n\tau)+\frac{1}{12}k_{1ij}-\frac{25}{132}k_{2ij}+\frac{73}{66}k_{3ij}) \end{aligned} \tag{11}$$

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

$$\begin{aligned} x_{ij}((n+1)\tau) &= x_{ij}(n\tau)+ \\ &\frac{1}{3}[\sum_{\rho=1}^{3}\frac{2}{3}\frac{k_{\rho ij}^2+k_{\rho+1ij}^2+k_{\rho ij}k_{\rho+1ij}}{k_{\rho ij}+k_{\rho+1ij}}] \end{aligned} \tag{12}$$

B. RK-Embedded Harmonic Mean

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

$$\begin{aligned} k_{1ij} &= f'(x_{ij}(n\tau))\ , \\ k_{2ij} &= f'(x_{ij}(n\tau)+\frac{1}{2}k_{1ij}), \\ k_{3ij} &= f'(x_{ij}(n\tau)-\frac{1}{8}k_{1ij}+\frac{5}{8}k_{2ij}), \\ k_{4ij} &= f'(x_{ij}(n\tau)-\frac{1}{4}k_{1ij}+\frac{7}{20}k_{2ij}+\frac{9}{10}k_{3ij} \end{aligned} \tag{13}$$

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

$$\begin{aligned} x_{ij}((n+1)\tau) &= x_{ij}(n\tau)+[\frac{k_{2ij}}{6}+\frac{k_{3ij}}{6}+ \\ &\frac{2}{3}(\frac{k_{1ij}k_{2ij}}{k_{1ij}+k_{2ij}})+\frac{2}{3}(\frac{k_{3ij}k_{4ij}}{k_{3ij}+k_{4ij}})] \end{aligned} \tag{14}$$

C. RK-Embedded Contra- Harmonic Mean

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

$$\begin{aligned} k_{1ij} &= f'(x_{ij}(n\tau))\ ,\ k_{2ij} = f'(x_{ij}(n\tau)+\frac{1}{2}k_{1ij})\ , \\ k_{3ij} &= f'(x_{ij}(n\tau)+\frac{1}{8}k_{1ij}+\frac{3}{8}k_{2ij})\ , \\ k_{4ij} &= f'(x_{ij}(n\tau)+\frac{1}{4}k_{1ij}-\frac{3}{4}k_{2ij}+\frac{3}{2}k_{3ij}) \end{aligned} \tag{15}$$

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

$$\begin{aligned} x_{ij}((n+1)\tau) &= x_{ij}(n\tau)+\frac{h}{3}[\frac{k_{1ij}^2k_{2ij}^2}{k_{1ij}+k_{2ij}} \\ &+\frac{k_{2ij}^2k_{3ij}^2}{k_{2ij}+k_{3ij}}+\frac{k_{3ij}^2k_{4ij}^2}{k_{3ij}+k_{4ij}}] \end{aligned} \tag{16}$$

d. RK-Embedded Heronian Mean

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

$$\begin{aligned} k_{1ij} &= f'(x_{ij}(n\tau)) = k_{1ij}^{*}, \\ k_{2ij} &= f'(x_{ij}(n\tau)+\frac{1}{2}k_{1ij}) = k_{2ij}^{*}, \\ k_{3ij} &= f'(x_{ij}(n\tau))+\frac{1}{2}k_{2ij})\ ,\ k_{4ij} = f'(x_{ij}(n\tau))+k_{3ij})\ , \\ k_{3ij} &= f'(x_{ij}(n\tau)-\frac{1}{48}k_{1ij}+\frac{25}{48}k_{2ij}) = k_{3ij}^{*}\ , \\ k_{4ij} &= f'(x_{ij}(n\tau)-\frac{1}{24}k_{1ij}+ \\ &\frac{47}{600}k_{2ij}+\frac{289}{300}k_{3ij}) = k_{4ij}^{*} \end{aligned} \tag{17}$$

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

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau) + \frac{1}{3}\left[\frac{k_{1ij}+k_{2ij}}{2} + \frac{k_{2ij}+k_{3ij}}{2} + \frac{k_{3ij}+k_{4ij}}{2}\right] \tag{18}$$

$$x^{*}_{ij}((n+1)\tau) = x_{ij}(n\tau) + \frac{1}{9}\left[k_{1ij} + 2(k_{2ij}+k_{3ij}) + k_{4ij} + \sqrt{|k_{1ij}+k_{2ij}|} + \sqrt{|k_{2ij}+k_{3ij}|} + \sqrt{|k_{3ij}+k_{4ij}|}\right].$$

VII. 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. 2(b), 3(b), 4(b) and 5(b) show the results of the raster simulator obtained from a complex image of 1,25,600 pixels.

Using RK-Embedded Heronian Mean, RK-Embedded Centroidal Mean, RK-Embedded Harmonic Mean and RK-Embedded Contra-harmonic Mean, the results of the raster simulator obtained from a complex image of 1, 25,600 pixels are depicted respectively in Figs. 2, 3, and 4. 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. 2(b). The same procedure has been adapted for getting the results shown in Figs. 3(b), 4(b) and 5(b). It is observed from Figs. 2(b), 3(b), 4(b) and 5(b) that the edges obtained by the RK-Embedded Heronian Mean is better than that obtained by the Embedded Centroidal Mean, Embedded Harmonic Mean and 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. 6 show a comparison between four different templates. These results were obtained by trial and error over more than 200 simulations on a Coins figure. compared to other two methods irrespective of the selection of templates. Fig. 7 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. 6 were obtained by simulating a small image of size 256×256 pixels using Averaging template on a Coins figure.

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

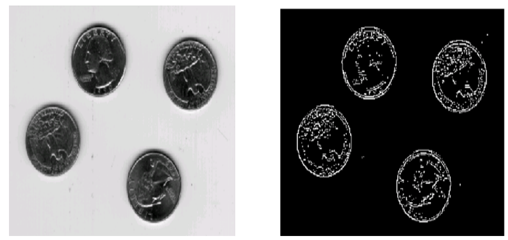


Fig. 3. (a) Original Coins Image; (b) After Averaging and Edge Detection Templates by adapting RK-Embedded Centroidal Mean**.**

Fig. 4. (a) Original Coins Image ;(b) After Averaging and Edge Detection Templates by adapting RK-Embedded Harmonic Mean.

Fig. 5. (a) Original Coins Image; (b) After Averaging and Edge Detection Templates by adapting RK-Embedded Contra-Harmonic Mean.

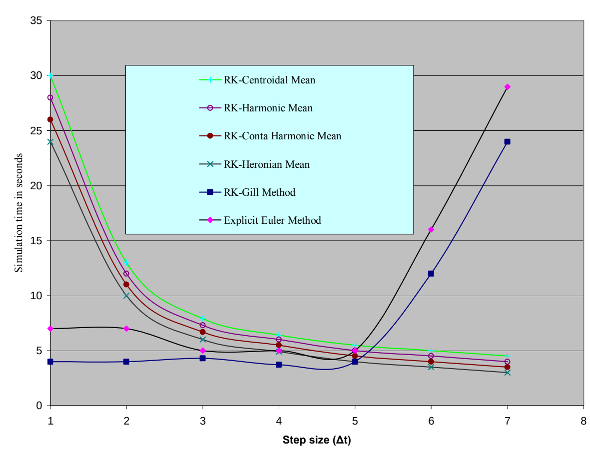
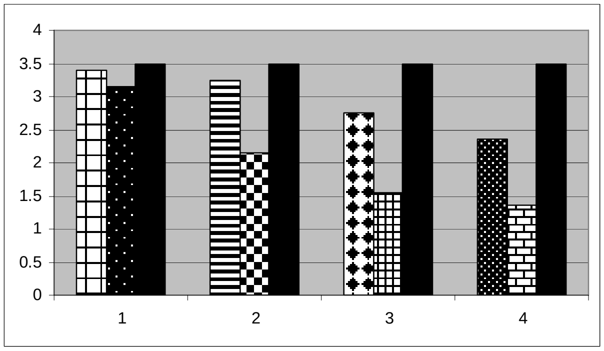


Fig. 6. Comparison of Six Numerical Integration Techniques using the Averaging Template

ISBN:978-988-98671-5-7



(Edge Detection Averaging Connected Component)

Fig. 7. Maximum Step Size (Δt) yields the convergence for different templates

VIII. CONCLUSION

The attention of the present article is focused 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 Heronian 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 Centroidal Mean, RK-Embedded Harmonic Mean and Embedded Contra-Harmonic Mean. It is of interest to mention that using RK-Embedded Heronian Mean; the edges of the output images are proved to be feasible and effective by theoretic analysis and simulation.

REFERENCES

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

[2] Chua, L.O., Yang, L.," Cellular Neural Networks: Applications, "*IEEE Transactions on Circuits and Systems,* 1988, 35, pp.1273–1290.

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

[4] Merson,R.H., An operational Method for the study of Integrating processes *Proc. Symposium. Data Processing,* Weapon Research Establishment, Salisbury, S.Australia, 1957.

[5] Fehlberg, E.,"Low order Classical Runge Kutta Formulas with step-size control and their application to some heat transfer problems," *NASA Technical Report,* 315, 1969, Extract Published in Computing, 1970, pp. 66-71.

[6] Evans, D. J. and Yaakub, A. R., "A new Runge Kutta RK(4,4) method," *Intern. J. Computer Math*., 1995, 58, pp.169-187.

[7] Yaakub, A. R. and Evans, D. J., "A fourth order Runge- Kutta RK(4,4) method with Error Control," Intern*. J. Computer Math*., 1999, 71, pp. 383-411.

[8] 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 Centroidal Means with Error Control," *International Journal of Computer Mathematic,2002,* 79(2), pp.247-269.

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

[10] R.Ponalagusamy and S.Senthilkumar, Multilayer raster CNN Simulation: An Efficient Numerical Integration Algorithm, Communicated *to Vision Image and Computing,* 2006.

[11] Chi-Chien Lee, Jose Pineda de Gyvez: "Single-Layer CNN Simulator, *International Symposium on Circuits and Systems*," 1994,6, pp.217–220.

[12] R.Ponalagusamy and S.Senthilkumar, A New Fourth Order Embedded RKAHeM(4,4) method with Error Control on Multilayer Raster Cellular Neural Network, Communicated to *Journal of Information Sciences, 2007.*

[13] D.J.Evans and N.Yaacob, "A Fourth order Runge-Kutta Method Based on the Heronian Mean Formula," Intern. J. Computer Math., 1995, 58, pp.103-115.

[14] 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)*, 1992, pp.1-10.

[15] K.K.Lai,and P.H.W.Leong,,"An area efficient Implementation of a Cellular Neural Network," IEEE, 1995, pp.51-54.

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

[17] 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, pp. 185-187.