Trends in

Applied Sciences Research

ISSN 1819-3579

Academic
Journals Inc.

www.academicjournals.com

Trends in Applied Sciences Research 3 (3): 242-252, 2008
ISSN 1819-3579

A Comparison of RK-Fourth Orders of Variety of Means on Multilayer Raster CNN Simulation

R. Ponalagusamy and S. Senthilkumar
Department of Mathematics, National Institute of Technology,
Tiruchirappalli-620 015, Tamilnadu, India

Abstract: In this study, an adaptable algorithm for simulating CNN arrays is implemented using various means such as Arithmetic Mean (AM), Centroidal Mean (CM), Harmonic Mean (HM), Contra Harmonic Mean (CoM), Heronian Mean (HeM), Geometric Mean (GM) and Root Mean Square (RMS). The role 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 investigate the potential applications of CNN. This study proposes an efficient pseudo code exploiting the latency properties of Cellular Neural Networks along with well known Runge-Kutta (RK) fourth order numerical integration algorithms. Simulation results and comparison have also been presented to show the efficiency of the various means in numerical integration algorithms. It is observed that the RK-Arithmetic Mean (AM) outperforms well in comparison with other means.

Key words: Cellular neural networks, various RK-fourth order means, edge detection, raster CNN simulation

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, 1988a) and they have found that CNN has many important applications in signal and real-time image processing. Roska (1994) and Chua and Roska (2002) has 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. Explicit Euler's Algorithm and the RK-fourth order Algorithm were discussed by Bader (1987, 1988). They have discussed about the application problem of comparative study of new truncation error estimates and intrinsic accuracies of some higher order RK algorithms. Also they adapted new technique for the early detection of stiffness in coupled differential equations and application to standard RK algorithms. Oliveira (1999) have discussed RK-Gill algorithm regarding evaluation of effectiveness of immobilized enzyme and discussed to solve mathematical undetermination at particle center point. Lee and de Gyvez (1994) introduced Euler, Improved Euler, Predictor-Corrector and fourth-order (quartic) RK algorithms in time-multiplexing CNN simulation.

Bottou *et al*. (2001) have proposed an efficient algorithm for converting digital documents to multilayer raster formats. Hadad and Piroozmand (2007) have described the application of a multilayer cellular network to model and solve the nuclear reactor dynamic equations. Chen *et al*. (2006) have studied in detail about the eight image tasks: Connected Component Detection (CCD) with down,

Corresponding Author: R. Ponalagusamy, Department of Mathematics, National Institute of Technology, Tiruchirappalli-620 015, Tamilnadu, India

right, +45 and −45° directions, edge detection, shadow projection with left and right directions and point removal are analyzed. Aizenberg *et al*. (2001) presented a special kind of cellular neural networks based on multiple valued threshold logic in the complex plane will be presented and its efficacy for medical imaging will be documented.

Wazwaz (1993) has performed numerical tests for several problems of IVPs in the form of y' = f (x, y) using fourth order RK formulas based on a variety of means. He has concluded that the level of accuracy in each formula adapted depends upon the error terms which mainly depend on the nature of the function f (x, y). It is of interest to mention that no investigation has so far been performed and the level of accuracy of each fourth order RK formulas on the problem of multilayer raster cellular numerical network simulation. In view of this, a modest effort has been made in the present paper to investigate the efficiency of the various RK formulas based on variety means in the problem of multilayer raster cellular numerical network simulation.

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:

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

and the output Equation is given by,

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

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.

From the Eq. 1 it is observed that the summation operators of each cell is affected by its neighboring cells. A: represents on the output of neighboring cells and is called as feedback operator, B: in turn affects the input control and is known as the control operator. In particular, the entry values of matrices A and B are dependent on the application chosen by the user which are space invariant and are referred as cloning templates. A current bias I and cloning templates establishes the transient behavior of the cellular nonlinear network. A continuous-time cell implementation is shown in Fig. 1b as an equivalent block diagram. CNNs have as input a set of analog values and its programmability is done via cloning templates. Thus, programmability is one of the most attractive properties of CNNs.

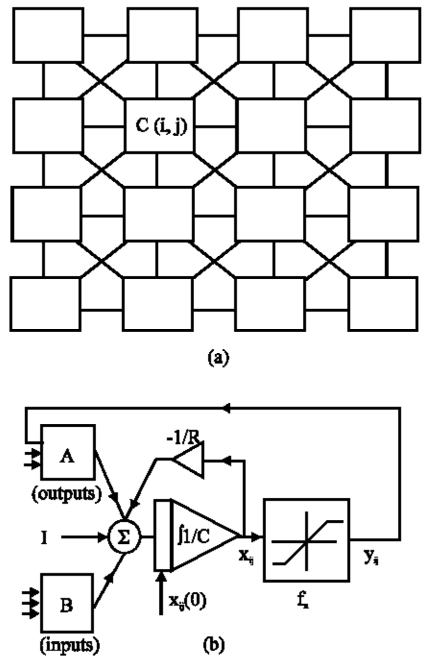


Fig. 1: Cellular neural networks: (a) Array structure and (b) Block diagram

PERFORMANCE OF RASTER CNN SIMULATIONS

Raster CNN simulation is an image scanning-processing technique for solving the system of difference equations of CNN. The Eq. 1 is space invariant, which means that A (i, j; k, l) = A (i-k, j-l) 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 fundamental 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 $\alpha\pm1$ value Chua and Yang (1988a). This whole simulating approach is referred to as raster simulation. A simplified pseudo code is presented below gives the exact notion of this approach.

Pseudo Code for Raster CNN Simulation

Step 1: Initially get the input image, initial conditions and templates from end user./* M, N = No. 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 $$\left(\frac{dx_{ij}(t_n)}{dt}\right) = 0 \text{ and } y_{kl} = \pm 1, \forall c(k,l) \in N_r(i,j)$$

```
{
convergence-flag[i][j] = 1;
converged-cells++ ;
}
} /* end for */
```

Step 4: /* Update the state values of the entire image */

```
for (i = 1; i< = M; i++)
for (j = 1; j< = N; j++)
{
if (convergence-flag[i][j]) continue;
x_ij(t_n) = x_ij(t_n+1);
}
Number of iteration++;
}
 /* end while */
```

For simulation purposes, a discretized form of Eq. 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 Chua and Yang (1988b) and Chua and Roska (1992). 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:

$$g\,(x,y) = T(f(x,y)) \tag{2}$$

Where:
g = The processed image.
f = The input image.
T = 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 alternative option to this scenario is multiplex the image processing operator.

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*. (1992).

$$f'(x(n\tau))=\frac{dx_{ij}(n\tau)}{dt}=-x_{ij}(n\tau)+\sum_{c(k,l)\in N_r(i,j)} A(i,j;k,l)y_{kl}(n\tau)+$$
$$\sum_{c(k,l)\in N_r(i,j)} B(i,j;k,l)u_{kl}(n\tau)+I, 1\leq i\leq M;\ 1\leq j\leq N;$$
$$y_{ij}(n\tau)=\frac{1}{2}\left[\left|x_{ij}(n\tau)+1\right|-\left|x_{ij}(n\tau)-1\right|\right],\ 1\leq i\leq M;\ 1\leq j\leq N; \quad (5)$$

Where:
τ = The normalized time.

For the purpose of solving the initial-value problem, well established Single Step methods of numerical integration techniques are used. (Wazwaz, 1993; Yaakub and Evans, 1997). These methods can be derived using the definition of the definite integral.

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

Seven types of numerical integration algorithms are used in time-multiplexing simulations. They are Arithmetic Mean (AM), Centroidal Mean (CM), Harmonic Mean (HM), Contra Harmonic Mean (CoM), Heronian Mean (HeM), Geometric Mean (GM), Root Mean Square (RMS) discussed in (Wazwaz, 1993; Yaakub and Evans, 1997).

Fourth Order RK Method Based on Different Means

In Table 1 the term h represents step size and k_1, k_2, k_3 and k_4 can be expressed as:
$k_1 = f(x_n,y_n)$.
$k_2 = f(x_n+a_1h,y_n+ha_1k_1)$.
$k_3 = f(x_n+(a_2+a_3)h, y_n+ha_2k_1+ha_3k_2)$.
$k_4 = f(x_n+(a_4+a_5+a_6)h,y_n+ha_4k_1+ha_5k_2+ha_6k_3)$.
Where, the given initial value problem is

$$\dot{y}=\frac{dy}{dx}=f(x,y)$$

and a_1, a_2, a_3, a_4, a_5 and a_6 are known constants on the type of fourth order RK-methods.

Table 1: Rank (Less Error in the order of Ascending): Fourth order RK with various means

Different means	y_{n+1} =
AM	$y_n+\frac{h}{6}[k_1+2(k_2+k_3)+k_4]$
CM	$y_n+\frac{2h}{9}[\frac{k_1^2+k_1k_2+k_2^2}{k_1+k_2}+\frac{k_2^2+k_2k_3+k_3^2}{k_2+k_3}+\frac{k_3^2+k_3k_4+k_4^2}{k_3+k_4}]$
HM	$y_n+\frac{2h}{3}[\frac{k_1k_2}{k_1+k_2}+\frac{k_2k_3}{k_2+k_3}+\frac{k_3k_4}{k_3+k_4}]$
CoM	$y_n+\frac{h}{3}[\frac{k_1^2k_2^2}{k_1+k_2}+\frac{k_2^2k_3^2}{k_2+k_3}+\frac{k_3^2k_4^2}{k_3+k_4}]$
HeM	$y_n+\frac{h}{9}[k_1+2(k_2+k_3)+k_4+\sqrt{\lvert k_1k_2\rvert}+\sqrt{\lvert k_2k_3\rvert}+\sqrt{\lvert k_3k_4\rvert}]$
GM	$y_n+\frac{h}{3}[\sqrt{\lvert k_1k_2\rvert}+\sqrt{\lvert k_2k_3\rvert}+\sqrt{\lvert k_3k_4\rvert}]$
RMS	$y_n+\frac{h}{3}[\sqrt{\frac{k_1^2+k_2^2}{2}}+\sqrt{\frac{k_2^2+k_3^2}{2}}+\sqrt{\frac{k_3^2+k_4^2}{2}}]$

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. Figure 2b, 3b, 4b, 5b, 6b 7b and 8b show the results of the raster simulator obtained from a complex image of 1, 25,600 pixels.

Arithmetic Mean (AM), Centroidal Mean (CM), Harmonic Mean (HM), Contra Harmonic Mean (CoM), Heronian Mean (HeM),Geometric Mean (GM) and Root Mean Square (RMS) the results of the raster simulator obtained from a complex image of 1, 25,600 pixels are depicted, respectively

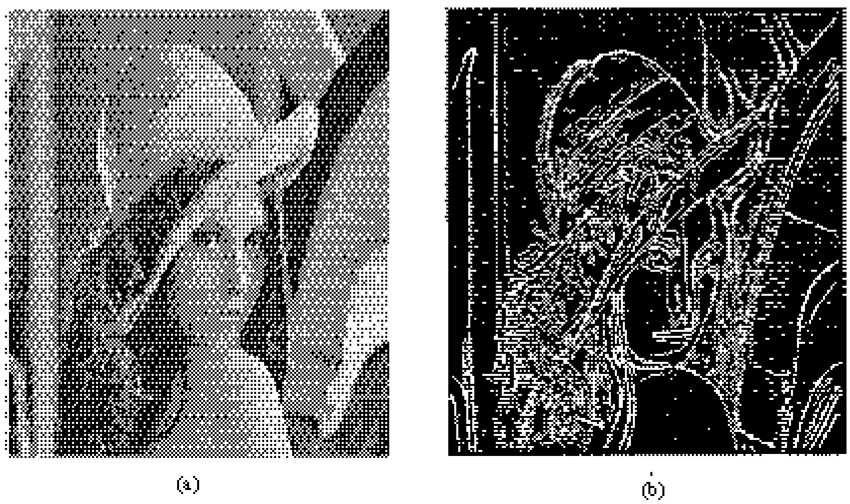


Fig. 2: (a) Original Lena Image and (b) After averaging and edge detection templates by employing RK-Arithmetic mean algorithm

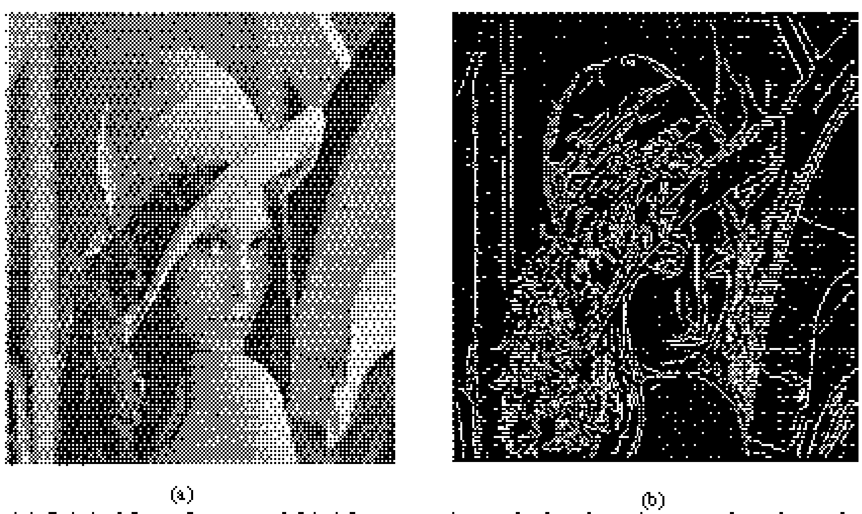
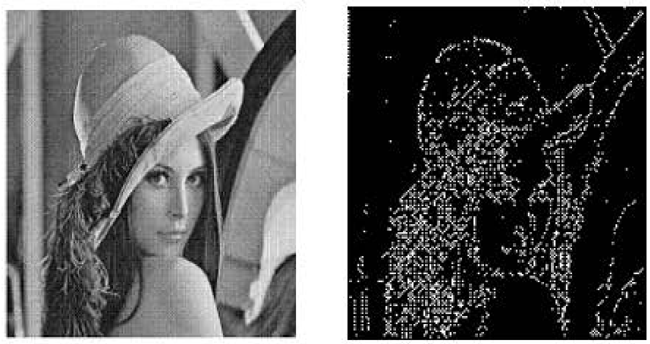


Fig. 3: (a) Original Lena Image and (b) After averaging and edge detection templates by mploying RK-centroidal mean algorithm

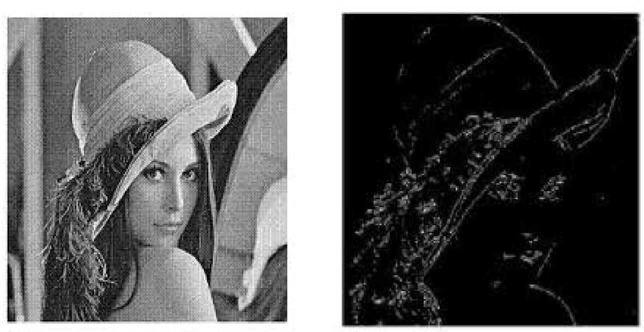


(a) (b)

Fig. 4: (a) Original Lena Image and (b) After averaging and edge detection templates by employing RK-harmonic mean algorithm

(a) (b)

Fig. 5: (a) Original Lena Image and (b) After averaging and edge detection templates by employing K-contra-harmonic mean algorithm



(a) (b)

Fig. 6: (a) Original Lena Image and (b) After averaging and edge detection templates by employing RK-heronian mean algorithm

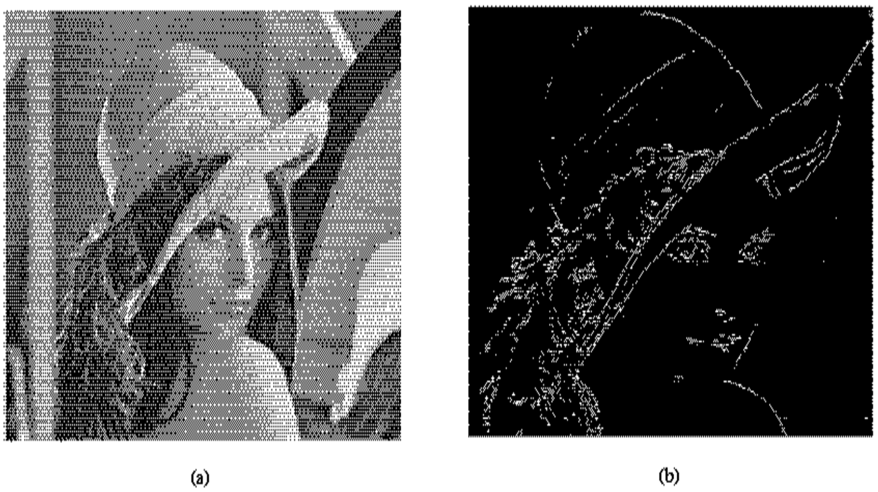


Fig. 7: (a) Original Lena Image and (b) After averaging and edge detection templates by employing RK-geometric mean algorithm

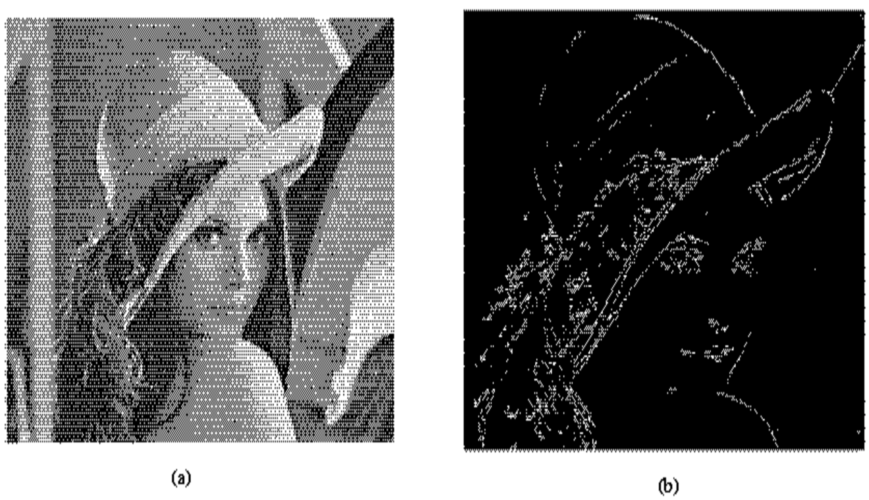


Fig. 8: (a) Original Lena Image and (b) After averaging and edge detection templates by employing RK-root mean square algorithm

in Fig. 2-8. For the present example an averaging template followed by an Edge Detection (Gonzalez *et al*., 2005) template were applied to the original image to yield the images displayed in Fig. 2b. The same procedure has been adapted for getting the results shown in Fig. 3b, 4b, 5b, 6b,7b and 8b. It is observed from Fig. 2b, 3b, 4b, 5b, 6b, 7b and 8b that the edges obtained by the Arithmetic Mean (AM) is better than that obtained by the Centroidal Mean (CM), Harmonic Mean (HM), Contra Harmonic Mean (CoM), Heronian Mean (HeM), Geometric Mean (GM) and Root Mean Square (RMS).

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

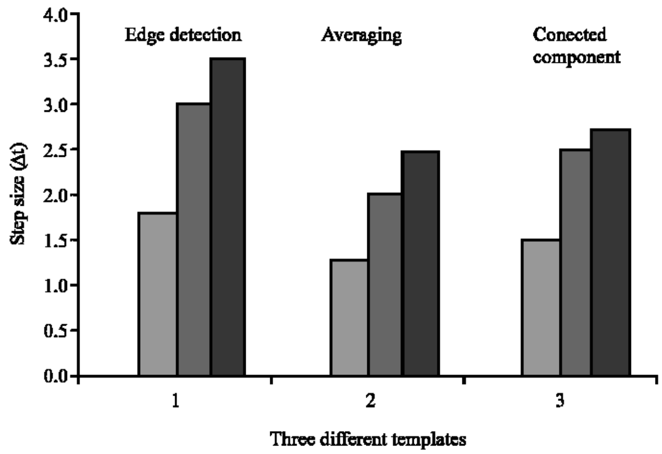


Fig. 9: Maximum step size (Δt) yields the convergence for three different templates

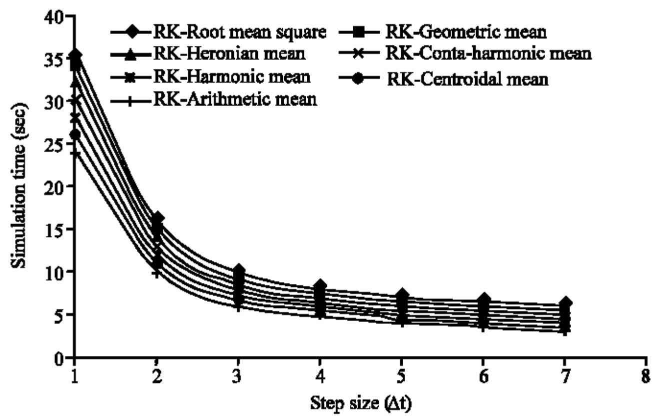


Fig. 10: Comparison of seven numerical integration techniques using the averaging template

size may slightly vary from one image to another, the values in Fig. 9 show a comparison between three different templates. It is observed from Fig. 9 that RK-Arithmetic Mean allows us to select a maximum step size as compared to other two methods irrespective of the selection of templates. These results were obtained by trial and error over more than 100 simulations on a Lena image.

It is observed from Fig. 9 that RK-Arithmetic Mean allows us to select a maximum step-size (Δt) as compared to other six methods irrespective of the selection of templates. Fig. 10 show that the significance of selecting an appropriate time step-size (Δt). If the step-size (Δt) 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 (Δt) taken is too large, it might not converge at all or it would be converges to erroneous steady state values. The results of Fig. 10 were obtained by simulating a small image of size 256×256 pixels using Averaging template on a Lena image.

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. In this paper an averaging template followed by edge detection template were applied to the original image to yield the required output image and it may be true for most of the edge detection approaches but these edge detection approaches vary in processing time due to their hardware limitations and the simulator chosen. It is pertinent to pin-point out here that the RK-Arithmetic Mean guarantees the accuracy of the detected edges and greatly reduces the impact of random noise on the detection results in comparison with other Means. It is of interest to mention that using RK-Arithmetic Mean; the edges of the output images are proved to be feasible and effective by theoretic analysis and simulation.

ACKNOWLEDGMENTS

This research was fully supported as a part of Technical Quality Improvement Programme (TEQIP), sponsored by Govt. of India, National Institute of Technology, Tiruchirappalli-620 015, Tamilnadu, India.

REFERENCES

Aizenberg, I., N. Aizenberg, J. Hiltner, C. Moraga and B.E. Meyer Zu, 2001. Erratum to cellular neural networks and computational intelligence in medical image processing. Image Vision Comput., 19: 177-183.

Bader, M., 1987. A comparative study of new truncation error estimates and intrinsic accuracies of some higher order Runge-Kutta algorithms. Comput. Chem., 11: 121-124.

Bader, M., 1988. A new technique for the early detection of stiffness in coupled differential equations and application to standard Runge-Kutta algorithms. Theor. Chem. Accounts, 99: 215-219.

Bottou, L., P. Haffner and Y. LeCun, 2001. Efficient convergence of digital documents to multilayer raster formats. IEEE, pp: 444-448.

Chen, H.C., Y.C. Hung, C.K. Chen, T.L. Liao and C.K. Chen, 2006. Image-processing algorithms realized by discrete-time cellular neural networks and their circuit implementations, 29: 1100-1108.

Chua, L.O. and L. Yang, 1988a. Cellular neural networks: Theory. IEEE Trans. Circuits Syst., 35: 1257-1272.

Chua, L.O. and L. Yang, 1988b. Cellular neural networks: Applications. IEEE Trans. Circuits Syst., 35: 1273-1290.

Chua, L.O. and T. Roska, 1992. The CNN universal machine. Part 1: The Architecture in International Workshop on Cellular Neural Networks and their Applications, pp: 1-10.

Chua, L.O. and T. Roska, 2002. Cellular Neural Networks and Visual Computing. Cambridge University Press, UK.

Gonzalez, R.C., E. Richard, S. Woods and L. Eddin, 2005. Digital Image Processing using MATLAB. Pearson Education Asia.

Hadad, K. and A. Piroozmand, 2007. Application of Cellular Neural Network (CNN) method to the nuclear reactor dynamic equations. Ann. Nucl. Energy, 34: 406-416.

Lee, C.C. and J.P. de Gyvez, 1994. Single-layer CNN simulator. Int. Symposium Circuits Syst., 6: 217-220.

Nossek, J.A., G. Seiler, T. Roska and L.O. Chua, 1992. Cellular neural networks: Theory and circuit design. Int. J. Circuit Theory Appl., 20: 533-553.

Oliveira, S.C., 1999. Evaluation of effectiveness factor of immobilized enzymes using Runge Kutta-Gill method: How to solve mathematical undetermination at particle center point? Bio. Proc. Eng., 20: 185-187.

Roska, 1994. CNNM Users Guide. Version 5.3x, Budapest.

Wazwaz, A.M., 1993. A comparison of modified runge-kutta formulas based on variety of means. Int. J. Comput. Math., 50: 105-112.

Yaakub, A.R. and D.J. Evans, 1997. New runge-kutta starters for multi-step methods. Int. J. Comput. Math., 71: 99-104.