

A NEW SIMULATION METHOD FOR TIME-DERIVATIVE CELLULAR NEURAL NETWORKS

S. Nergis Tural Polat and Vedat Tavsanoglu

Department of Electronics and Communications Engineering, Yildiz Technical University
Yildiz Teknik Universitesi, Barbaros Bulvari, Besiktas, 34349, Istanbul, Turkey
email: {nergis, tavsnav}@yildiz.edu.tr

ABSTRACT

Time-derivative Cellular Neural Networks (TDCNN) is a general class of CNN having derivative connections between cells. It has been reported that several mixed-domain 3D spatiotemporal transfer functions for linear filtering can be implemented via TDCNN such as bandpass filters and visual receptive field models. In this paper a new computer simulation method for TDCNN is presented. The new simulation method reduces simulation time dramatically and simplifies the simulation steps.

1. INTRODUCTION

Biological organisms excel at processing large amount of data in real time whereas computers usually are not good at such tasks because of their point processor architecture. Thus a great deal of effort has been spent on the development of machines that mimic biological systems. Cellular Neural Networks consist of regular arrays of simple processing units that interact with only their nearest neighbors [1]. The CNN architecture bears striking resemblance to aforementioned biological organisms and they are tailor-made for analog VLSI implementations because of their nearest neighbor connections and usually space invariant connection weights. In recent years, an addition to the general CNN architecture which employs temporal derivative connections between cells is proposed [2,3,4,5]. It is shown that with these derivative connections a broader range of continuous-time discrete-space mixed-domain 3D spatiotemporal transfer functions can be realized. Some filtering examples that have not been modeled by linear CNNs such as visual cortical receptive field modeling and spatiotemporal cone filter design are given in [3]. Spatiotemporal bandpass filter examples can be found in [4,5].

The paper is organized as follows: Section II gives a brief description of the time-derivative CNN and the templates for a real spatiotemporal TDCNN bandpass filter. In section III we examine the computer simulation of the first derivative TDCNN and propose a new simulation method based on Euler's approximations. Simulation results and the elapsed times for the two methods are also compared in this section. Finally we conclude the paper in section IV.

2. TIME-DERIVATIVE CELLULAR NEURAL NETWORKS (TDCNN)

Time-derivative CNN (TDCNN) [2] extends the original CNN description in [1] by adding extra derivative connections between cells. These derivative connections enable us to implement a broader class of mixed domain spatiotemporal linear filters. A time-derivative linear CNN is described by

$$\begin{aligned} \frac{dx(i, j, t)}{dt} &= \sum_{m,n=-r}^r A(m, n)x(i+m, j+n, t) \\ &+ \sum_{m,n=-r}^r B(m, n)u(i+m, j+n, t) \\ &+ \sum_{q \in \{1..D\}} \left[\sum_{m,n=-r}^r A_q(m, n) \frac{dx^q(i+m, j+n, t)}{dt^q} \right. \\ &\quad \left. + \sum_{m,n=-r}^r B_q(m, n) \frac{du^q(i+m, j+n, t)}{dt^q} \right] \end{aligned} \quad (1)$$

The first two terms on the right hand side of (1) are the same as in the case of original CNN equation, A and B are feedback and feed-forward cloning templates, u is input and x denotes the state and output of the linear network. A_q and B_q are defined as qth derivative feedback and feedforward templates, respectively, and r denotes the neighborhood of the CNN. It has been shown that by adding first order derivatives of the outputs of the neighboring cells to the original CNN equation, bandpass spatiotemporal filters can be realized [2,4,5]. In [2] a real coefficient 5x5 neighborhood first derivative TDCNN bandpass filter and a complex coefficient 3x3 neighborhood first derivative TDCNN bandpass filter are designed. For these first derivative TDCNNs (1) becomes

$$\begin{aligned} \frac{dx(i, j, t)}{dt} &= \sum_{m,n=-r}^r A(m, n)x(i+m, j+n, t) \\ &+ \sum_{m,n=-r}^r B(m, n)u(i+m, j+n, t) \\ &+ \sum_{m,n=-r}^r A_1(m, n) \frac{dx(i+m, j+n, t)}{dt} \end{aligned} \quad (2)$$

The template coefficients of the real TDCNN are given as

$$A = A_R \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{G_{YC} + G_{YS}}{2} & 0 & 0 \\ 0 & \frac{G_{XC} - G_{XS}}{2} & 0 & \frac{G_{XC} + G_{XS}}{2} & 0 \\ 0 & 0 & \frac{G_{YC} - G_{YS}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + B_R A_1 \quad (3)$$

$$A_1 = \begin{bmatrix} 0 & 0 & \frac{G_{YC}^2 + G_{YS}^2}{4} & 0 & 0 \\ 0 & \begin{pmatrix} \frac{G_{XC}G_{YC}}{2} \\ -\frac{G_{XS}G_{YS}}{2} \end{pmatrix} & 2a_0 G_{YC} & \begin{pmatrix} \frac{G_{XC}G_{YC}}{2} \\ +\frac{G_{XS}G_{YS}}{2} \end{pmatrix} & 0 \\ \frac{G_{XC}^2 + G_{XS}^2}{4} & 2a_0 G_{XC} & \begin{pmatrix} a_0 + \frac{G_{XC}^2 - G_{XS}^2}{2} \\ +\frac{G_{YC}^2 - G_{YS}^2}{2} \end{pmatrix} & 2a_0 G_{XC} & \frac{G_{XC}^2 + G_{XS}^2}{4} \\ 0 & \begin{pmatrix} \frac{G_{XC}G_{YC}}{2} \\ +\frac{G_{XS}G_{YS}}{2} \end{pmatrix} & 2a_0 G_{YC} & \begin{pmatrix} \frac{G_{XC}G_{YC}}{2} \\ -\frac{G_{XS}G_{YS}}{2} \end{pmatrix} & 0 \\ 0 & 0 & \frac{G_{YC}^2 + G_{YS}^2}{4} & 0 & 0 \end{bmatrix} \quad (4)$$

$$B = b(0,0) = 1 \quad (5)$$

Where $A_R, B_R, G_{XC}, G_{XS}, G_{YC}, G_{YS}, a_0$ are real-valued coefficients. The spatiotemporal transfer function of this real TDCNN is obtained as

$$H_{TDR} = \frac{1}{(p_t - B_R) \begin{pmatrix} a^2_{z_x z_y} & -b^2_{z_x z_y} \end{pmatrix} - A_R \begin{pmatrix} a_{z_x z_y} + j b_{z_x z_y} \end{pmatrix}}$$

$$a_{z_x z_y} = a_0 + \frac{1}{2} \left[G_{XC} (z_x + z_x^{-1}) + G_{YC} (z_y + z_y^{-1}) \right] \quad (6)$$

$$b_{z_x z_y} = \frac{1}{2j} \left[G_{XS} (z_x - z_x^{-1}) + G_{YS} (z_y - z_y^{-1}) \right]$$

The spatiotemporal amplitude frequency response of this TDCNN is given in Fig. 1.

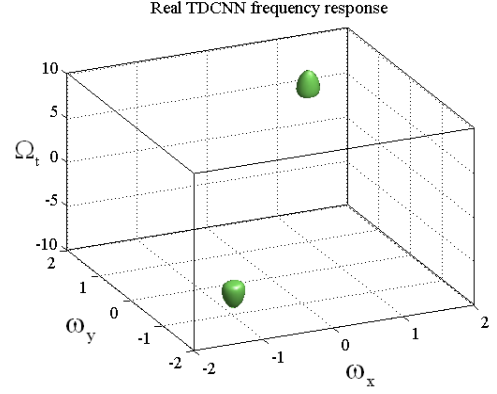


Figure 1 – Real TDCNN amplitude frequency response

3. DISCRETE TIME SIMULATION OF TDCNN

To simulate the TDCNN on a computer, it is necessary to use an approximation method for temporal derivatives. The simulation algorithms given so far for the original CNN structure are not suitable for the TDCNN and the only simulation available is given in [3] using MATLAB SIMULINK which, due to the slow nature of SIMULINK, involves a very cumbersome, extremely time consuming and error prone process. The time necessary to process a 30x30 3-D sinusoidal input image is about 30,000 seconds (8.5 hours) on a P4 C2Duo 3GHz, 4GB RAM PC.

The SIMULINK simulation steps involve:

- building one cell of the TDCNN network from the signal- flow graph representation of the (2),
- constructing the TDCNN network by placing the cells and connecting $3 \times (2r+1) \times (2r+1)$ neighboring cells (total number of elements of the A, B and A_1 templates) for each cell (In the general case this number is 75 for the templates of size 5x5. However, this number reduces to 25 for the given templates),
- rearranging the 3-D spatiotemporal input image as a 1-D temporal function of the spatial location of the (i,j)th cell,
- running the SIMULINK model file,
- rearranging the output as a 3-D spatiotemporal output image from each individual cell's 1-D temporal output.

In the following, we introduce a new method which uses Euler approximations for the time derivatives. Equation (2) has temporal derivatives on both sides. Decomposing A_1 template to its center and surround templates and rearranging (2) to have only the first derivative of the cell (i,j) on the left hand side yields

$$\begin{aligned} [1-A_1(0,0)] \frac{dx(i,j,t)}{dt} &= \sum_{m,n=-r}^r A(m,n)x(i+m,j+n,t) \\ &+ \sum_{m,n=-r}^r B(m,n)u(i+m,j+n,t) \end{aligned} \quad (7)$$

$$\begin{aligned} &+ \sum_{m,n=-r}^r \tilde{A}_1(m,n) \frac{dx(i+m,j+n,t)}{dt} \\ \tilde{A}_1(m,n) &= \begin{cases} A_1(m,n), & m,n \neq 0,0 \\ 0, & m,n = 0,0 \end{cases} \end{aligned} \quad (8)$$

Applying forward Euler approximation for the derivatives in (7) gives

$$\begin{aligned} [1-A_1(0,0)] \frac{x(i,j,k+1)-x(i,j,k)}{T_s} &= \sum_{m,n=-r}^r A(m,n)x(i+m,j+n,k) \\ &+ \sum_{m,n=-r}^r B(m,n)u(i+m,j+n,k) \\ &+ \sum_{m,n=-r}^r \tilde{A}_1(m,n) \frac{x(i+m,j+n,k+1)-x(i+m,j+n,k)}{T_s} \end{aligned} \quad (9)$$

where T_s denotes the integration time step and k is the discrete time variable. Equation (9) implies that in order to evaluate the output of a cell at $(k+1)$ th time step it is necessary to know the outputs of the neighboring cells at the same time step. Thus the computation of the output at $(k+1)$ th time step becomes impossible. To overcome this obstacle, while using forward Euler approximation for the derivative on the left hand side, which is the derivative of the cell (i,j) in (7), we use backward Euler approximation for the derivatives on the right hand side, which are the derivatives from the neighboring cells. By doing this, the output of a cell at $(k+1)$ th time step can be evaluated from the outputs and inputs of the neighboring cells at k th and $(k-1)$ th time steps. Equation (7) then becomes

$$\begin{aligned} [1-A_1(0,0)] \frac{x(i,j,k+1)-x(i,j,k)}{T_s} &= \sum_{m,n=-r}^r A(m,n)x(i+m,j+n,k) \\ &+ \sum_{m,n=-r}^r B(m,n)u(i+m,j+n,k) \\ &+ \sum_{m,n=-r}^r \tilde{A}_1(m,n) \frac{x(i,j,k)-x(i,j,k-1)}{T_s} \end{aligned} \quad (10)$$

Rearranging (10) yields

$$\begin{aligned} x(i,j,k+1) &= x(i,j,k) \\ &+ \frac{T_s}{1-A_1(0,0)} \left[\sum_{m,n=-r}^r A(m,n)x(i+m,j+n,k) \right. \\ &\quad \left. + \sum_{m,n=-r}^r B(m,n)u(i+m,j+n,k) \right] \\ &+ \frac{1}{1-A_1(0,0)} \left[\sum_{m,n=-r}^r \tilde{A}_1(m,n)x(i+m,j+n,k) \right. \\ &\quad \left. - \sum_{m,n=-r}^r \tilde{A}_1(m,n)x(i+m,j+n,k-1) \right] \end{aligned} \quad (11)$$

By using both forward and backward Euler approximations, a first-order continuous-time system has been represented with a second-order discrete-time system, which holds as long as the integration time-step T_s is small.

There are further points of importance in simulating (11). The spatiotemporal image processing is somewhat different from spatial image processing with CNN. Firstly, for spatial image processing, the output is obtained when the time derivatives become zero. The time constant $\tau = RC$ of the CNN circuit is chosen very small so that the circuit converges in a very short time and the derivatives of the cell outputs become zero. Secondly, the input image frame is constant and applied to the cell inputs during the whole simulation. For spatiotemporal image processing, we are interested in the time changing behavior of the cell outputs, in other words we must apply the inputs when the time derivatives of the outputs are nonzero. Thus the time constant of the CNN should be chosen larger. In addition, the input image consists of several image frames changing with time. Let T_f denote the period between the two input frames. The simulation results have shown that $\tau \gg T_f \geq 3T_s$ must be satisfied.

To compare the two methods, three different size TDCNNs are simulated for each of the sinusoidal and impulse inputs. The proposed method works a lot faster than the SIMULINK simulation regardless of the size of the circuit. The elapsed times for these simulations are measured on a P4 C2Duo 3GHz, 4GB RAM PC and summarized in Table 1.

The impulse response simulation outputs of SIMULINK and the proposed method of the real TDCNN are presented in Fig. 2 and Fig. 3 respectively. The sinusoidal simulation results for the two methods are given in Fig. 4 and Fig. 5. The simulation results obtained using the proposed method for an impulsive input differs by 5% and that obtained for a sinusoidal input differs by 10% compared to the SIMULINK outputs.

Table 1 – Simulation Elapsed Times

Simulation Elapsed Times in seconds		
Circuit size and input type	SIMULINK	Proposed Method
10x10 impulse input	31.964	0.014
10x10 sinusoidal input	110.11	0.0374
20x20 impulse input	1561.63	0.0249
20x20 sinusoidal input	3386.43	0.0819
30x30 impulse input	16500	0.0465
30x30 sinusoidal input	30000	0.159

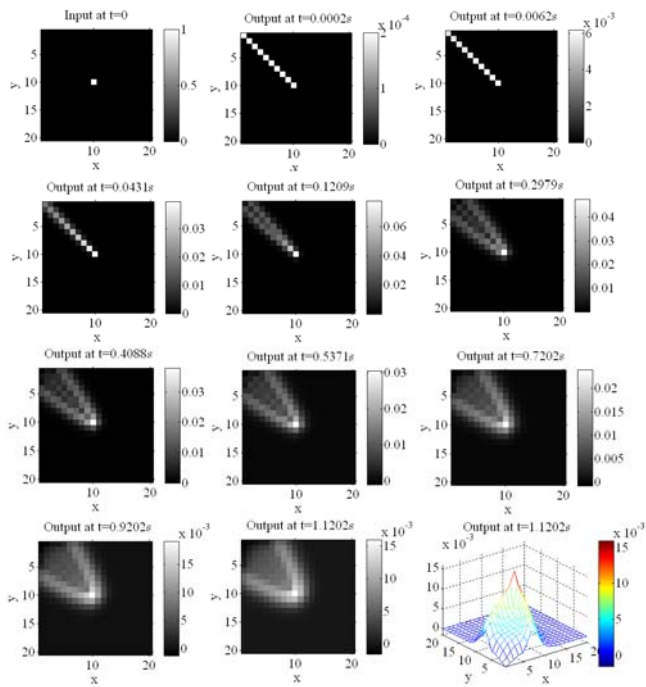


Figure 2 – SIMULINK impulse response

$$\text{for } A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, B = [1], A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

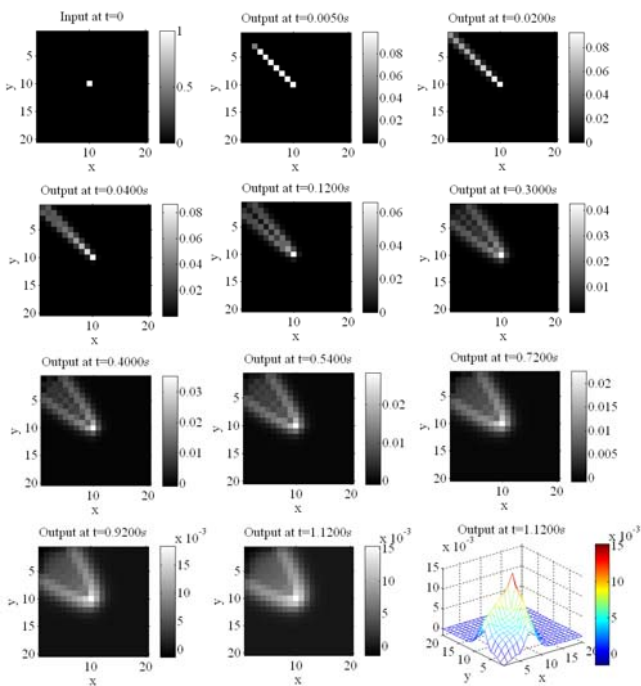


Figure 3 – Proposed algorithm's impulse response for the same templates as Fig.2

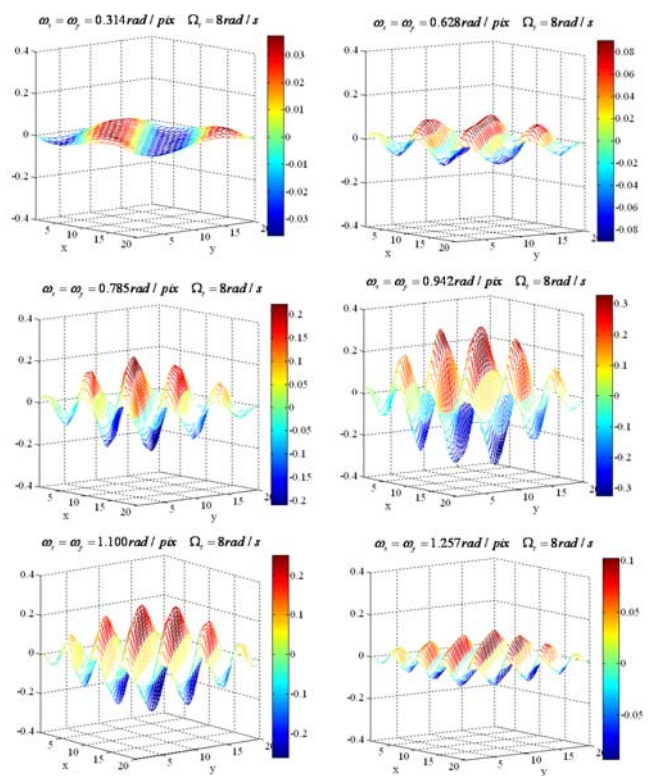


Figure 4 – SIMULINK sinusoidal input simulation results for several spatial frequencies. The passband of the filter is around

$$\omega_x = \omega_y = 1 \text{ rad / pix}, \Omega_t = 8 \text{ rad / s}$$

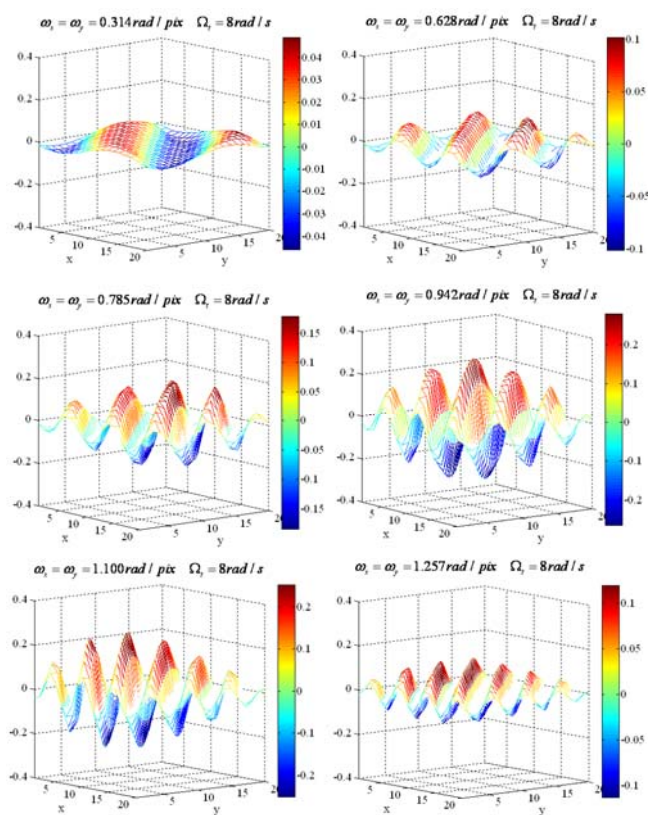


Figure 5 – Proposed algorithm's sinusoidal input simulation results for several spatial frequencies

4. CONCLUSION

General 3D continuous-time discrete-space mixed-domain spatiotemporal filters can be realized by TDCNNs. In this paper a new computer simulation method for the TDCNN is presented. The presented method obtains the desired result in milliseconds regardless of the circuit size. The proposed simulation is carried out in a single step and can be claimed to be fastest method given to date. The only other existing method uses SIMULINK which requires steps such as generating the SIMULINK model file, transferring input pixel values to cell inputs and obtaining the outputs from the cell outputs. It is believed that the introduction of the proposed simulation method will pave the way for the design of new processor architectures implementing the TDCNN on FPGA.

ACKNOWLEDGMENT

The authors would like to acknowledge Drs. Emmanuel M. Drakakis and Henry Man Den Ip for helpful discussions and providing information on SIMULINK simulations.

REFERENCES

- [1] L. O. Chua and L. Yang, "Cellular neural networks: Theory and applications," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 35, pp. 1257–1290, Oct. 1988
- [2] H. M. D. Ip, E. M. Drakakis, and A. A. Bharath, "Towards analog vlsi arrays for nonseparable 3-D spatiotemporal filtering," in *Proc. IEEE Int. Workshop Cellular Neural Networks Its Applications (CNNA'06)*, Istanbul, Turkey, 2006, pp. 1–6.
- [3] H. M. D. Ip, E. M. Drakakis, and A. A. Bharath, "On analog networks and mixed-domain spatiotemporal frequency response," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 272–285, Feb. 2008.
- [4] H. M. D. Ip, E. M. Drakakis, and A. A. Bharath, "Synthesis of nonseparable 3-D spatiotemporal bandpass filters on analog networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 286–298, Feb. 2008
- [5] H. M. D. Ip, E. M. Drakakis, and A. A. Bharath, "A non-separable 3-D spatiotemporal bandpass filters with analog networks," *ISCAS 2008*, 18-21 May 2008, Seattle, WA.