A Novel PDE Based Image Denoising Model with Applications in Nanoindustry

Angela Slavova, Victoria Rashkova *

Abstract

Recently many mathematical models for image processing have been widely applied in computer visualization for nanoindustry. The nonlinear diffusion PDE have been broadly applied in image processing.
In this paper we propose a convection - diffusion filter by adding a convection term in the modified diffusion equation as a physical interpretation for removing the noise. We study the dynamics of such equations by the discretization of this convection - diffusion model. Numerical experiments show that our method is reasonably better in removing noise.
AMS Subject Classification: 92B20, 37M05, 37N30, 35Q68, 35K67, 35K57
Key words: convection-diffusion equations, Perona-Malik equation, Cellular Neural Networks (CNN), algorithms, nanoindustry, filters, image denoising

1 Introduction

Recently many mathematical models for image processing have been widely applied in computer visualization. The nonlinear diffusion partial differential equation have been broadly applied in image processing since the first model was introduced in 1987 [4]. Through the time evolution, the diffusion can effectively remove the noise as well as having edge enhancement simultaneously. Since then various nonlinear diffusion filters have been widely proposed in implementing the image denoising / enhancement, edge detection and flow filed visualization . The common feature for nonlinear diffusion model is that the diffusion coefficient is small as the gradient of image is large. However the diffusion coefficient is a function of the convolution of the Gaussian kernel and solution such that this requires an extra cost in computing the nonlinear diffusion coefficient. In the

*This paper is supported by the project DID 02/15

numerical experiments we find that when using the nonlinear diffusion model in denoising the noise is not quite good [6]. Hence we propose a convection-diffusion filter by adding a convection term in the modified diffusion equation as a physical interpretation for removing the noise. The aim of this paper is to focus on the noise removal algorithm for extracting the target information (image) precisely. The main idea of our model algorithm is to diffuse the noise by following the convection direction during time evolution. To prevent the numerical layer in the discontinuities on the relative coarse grids we use the Cellular Neural Network (CNN) approach [1,2,7].

The principle that physics is doing the computation leads us to uncover other physical and/or biological effects to implement our topographic sensory computer architecture. A quite natural choice is to turn to optics, keeping in mind that a spatial correlation can be made "instantly", with the speed of light. Taking a so called 4f focal plane system using two lenses and a programmable light valve, containing a plain with programmable light transmissivity in each pixel, a spatial correlation can be achieved. From this simple effect, a stored Programmable Opto-electronic Analogic CNN Computer (POAC) has been successfully constructed recently [5], using two different lasers (red and green) and a bacterio-rhodopsine film as the programmable light valve or Local Analog Memory array. The size of the templates might be as big as 31x31, presently using an acusto-optical deflector (later replaced by a semiconductor laser array). Needless to say, this cellular architecture will be a must in many nano-systems, already emerging.

In Section 2 we present nonlinear diffusion Perona-Malik equation for modeling the image de-noise and its properties studied via polynomial CNN (PCNN). In section 3 we introduce our models: the convection-diffusion filter with manipulation in both diffusion and convection terms for controlling the smoothing process. Section 4 deals with CNN model of the convection-diffusion equation and its dynamics. Two examples are given in order to demonstrate the simulation results.

2 Perona- Malik type nonlinear diffusion equation and its CNN model

Dynamic properties of Perona-Malik based filters are summarized in [4]. It is well known that it converges toward a constant steady state solution, representing the average value of the initial image. In order to obtain a non trivial output image, the system evolution has to be stopped after a finite time (usually called scale). In the most general case, the scale depends on the object and on the characteristics of the input image, and hence is no *a priori* known time for stopping the image processing.

Let the noisy image be a given scaled intensity map $u_0(x) : \Omega \to [0, 255]$ for the image domain $\Omega \in \Re^2$. The nonlinear diffusion equation was first proposed by Perona and Malik in filtering the noise. They built a sequence of continuous images $u(x, t)$ on the abstract scale t and through the nonlinear diffusion equation to remove the noise during the scaling (time) revolution. Many of such nonlinear diffusion filtering models have been implemented. We briefly review this model as follows. The Perona- Malik type nonlinear isotropic diffusion equation is:

$$(1) \qquad \begin{cases} u_t(x,t) - div(g(|\nabla G_\sigma * u|)\nabla u(x,t)) = 0 & \text{in} \quad \Omega \times I \\ \frac{\partial u}{\partial n}(x,t) = 0 & \text{on} \quad \partial\Omega \times I \\ u(x,0) = u_0(x) & \text{on} \quad \Omega \end{cases}$$

where the initial value $u_0(x)$ is the given noisy image in the gray level, $I = [0, T]$ is the scaling (time) interval for some $T > 0, \Omega$ is a simply bounded rectangular domain with boundary $\partial\Omega$ and n is the outward unit normal vector to $\partial\Omega; g$ is a given non- increasing function. There are several choices for $g(s)$. Select a monotonic decreasing function

$$g(s) = \frac{1}{1 + s^2}$$

and they introduced the Gaussian kernel,$G_\sigma * u$, for the existence and uniqueness of (1). Thus, the diffusion coefficient $g(|\nabla G_\sigma * u|)$ is inhibited as the gradient of image intensity is big i.e. the diffusion coefficient is small around the image edge. Hence ND preserves the edges of image and protects the brightness of the image simultaneously. Perona and Malik considered the Galerkin finite element method for the discretization of (1). We shall apply polynomial CNN in order to study its dynamics.
The following diffusion functions were proposed by Perona and Malik:

$$(2) \qquad g(\|\nabla u\|^2) = e^{-(\frac{\|\nabla u\|}{k})^2}, g(\|\nabla u\|^2) = [1 + (\frac{(\|\nabla u\|}{k})^2]^{-1}$$

Let us suppose that the continuous space domain is composed by $M \times M$ points arranged on a regular grid, u_{ij} represents the pixel value
In order to implement a general polynomial CNN architecture, let us consider

the following basis function:

$$f(z) = 1 - |\frac{1}{2}(|\frac{z}{m} + 1| - |\frac{z}{m} - 1|)|$$

and let approximate the $g(.)$ functions (2) with the following expression:

$$\gamma(z) = \sum_{p=1}^{Q} C_p f^p(z)$$

We obtain a general nonlinear PDE based polynomial CNN model:

$$\frac{du_{ij}(t)}{dt} = \sum_{(kl)\in N_{ij}} \Gamma_{kl}(u_{kl} - u_{ij}) \tag{3}$$

$$\Gamma_{kl} = \sum_{p=1}^{Q} \frac{C_p}{2h^2}[f^p(\|\widetilde{\nabla} u_{k,l}\|^2) + f^p(\|\widetilde{\nabla} u_{i,j}\|^2)]$$

The polynomial CNN model (PCNN) (3) approximate the functions $g(.)$ with the expression $\gamma(.)$ that turns out to be different from zero only for $|z| = \|\widetilde{\nabla} u_{i,j}\|^2 < m$ This practical approximation presents the advantage of stopping the evolution of the image when the approximated gradient magnitude $\|\widetilde{\nabla} u_{i,j}\|^2$ is greater than the threshold m. Hence, the output image exhibits a segmented structure. The above behavior is possible because the PCNN system (3) presents ore than one equilibrium point and for each initial image, the output corresponds to one of these equilibria.

Conclusion 1 *The PCNN model (3) exhibit the coexistence of the constant average value equilibrium point (that is only admissible for the Perona-Malik discretized models) and of an infinite set of equilibrium points. As a consequence, the correct output is obtained without stopping the evolution of the system. This represents a significant advantage from the algorithmic point of view.*

3 Convection- diffusion equation

Here we propose a popular nonlinear convection- diffusion model for image demising as well as image compression. Starting from the nonlinear isotropic diffusion described in previous section we modify (1) and construct both convection and diffusion terms based on the gradient of image intensity; thus the direction of image smoothing is normalized to the gradient of image intensity. Consider our filter: the convection- diffusion problem (CD)

$$(4)\qquad \begin{cases} \frac{\partial u}{\partial t} - \varepsilon(|\nabla u|)\Delta u + \beta(|\nabla u|).\nabla u = 0 & \text{in} \quad \Omega \times I \\ \frac{\partial u}{\partial n} = 0 & \text{on} \quad \partial\Omega \times I \\ u(x,0) = u_0(x) & \text{on} \quad \Omega \end{cases}$$

where the diffusion coefficient is denoted by $\varepsilon(|\nabla u|) \equiv \frac{1}{1+|\nabla u|^2}$ and the convection vector $\beta(|\nabla u|) \equiv \gamma\frac{\nabla u^{\perp}}{|\nabla u|_\epsilon})$ for positive constant γ (the size of convection vector) with the Evans- Spruck regularization [] $|\nabla u|_\epsilon \equiv \sqrt{u_x^2 + u_y^2 + \epsilon}$ for $0 < \varepsilon \ll 1$ to avoid the singularity. Since the diffusion coefficient ε is inhibited as the gradient of image intensity is big, CD preserves the edges of image and protects the brightness of the image simultaneously.

However it is well known that if the diffusion coefficient ε is sufficiently small in comparison with the quantity $|\beta|h$ where the solution is discontinuous then the Galerkin finite element scheme leads to severe oscillation. The streamline diffusion finite element method resolves the oscillation problem; however it causes some artifacts (overshooting/ down shooting) around the edge of discontinuities. To avoid above phenomena we have proposed a CNN approach.

Consider now the following convection-diffusion problem:

$$(5)\qquad \begin{cases} \frac{\partial u}{\partial t} = b\nabla u + cu, & \text{in} \quad D \equiv (0,1)^2 \\ u = 0 & \text{on} \quad \partial D, \end{cases}$$

where $b(x,t) \geq \beta$, $c(x,t) \geq 0$ and $c_0^2(x,t) \equiv (c - b_x/2) \geq \gamma$, $x, t \in \bar{D}$, β and γ are some positive constants. It is known [] that for this assumptions there exists an unique solution of the CD problem (5).

In the next section we shall construct the CNN algorithm for studying the dynamics of the CD equation (5).

4 CNN approach for studying the dynamics of the convection-diffusion model

CNN is simply an analogue dynamic processor array, made of cells, which contain linear capacitors, linear resistors, linear and nonlinear controlled sources. Let us consider a two-dimensional grid with 3×3 neighborhood system as it is shown on Fig.1.

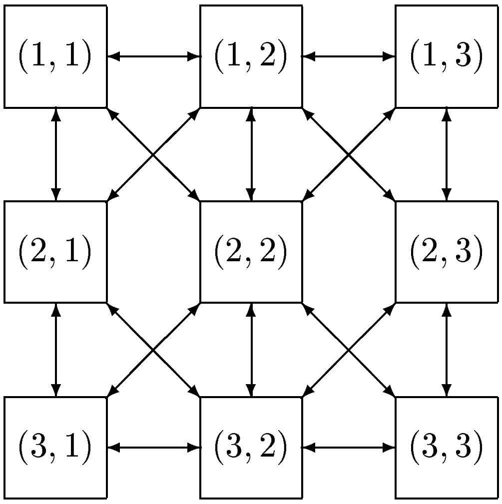

Fig.1. 3×3 neighborhood CNN.

The squares are the circuit units - cells, and the links between the cells indicate that there are interactions between linked cells. One of the key features of a CNN is that the individual cells are nonlinear dynamical systems, but that the coupling between them is linear. Roughly speaking, one could say that these arrays are nonlinear but have a linear spatial structure, which makes the use of techniques for their investigation common in engineering or physics attractive.

We will give the general definition of a CNN which follows the original one:

Definition 1 *The CNN is a*

a). 2-, 3-, or n- dimensional array of

b). mainly identical dynamical systems, called cells, which satisfies two properties:

c). most interactions are local within a finite radius r, and

d). all state variables are continuous valued signals.

Definition 2 *An $M \times M$ cellular neural network is defined mathematically by four specifications:*

1). CNN cell dynamics;

2). CNN synaptic law which represents the interactions (spatial coupling) within the neighbor cells;

3). Boundary conditions;

4). Initial conditions.

Suppose for simplicity that the processing elements of a CNN are arranged on a 2-dimensional (2-D) grid (Fig.1). Then the dynamics of a CNN, in general, can be described by:

$$
\begin{aligned}
\dot{x}_{ij}(t) &= -x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} \tilde{A}_{ij,kl}(y_{kl}(t), y_{ij}(t)) + \\
&\quad + \sum_{C(k,l)\in N_r(i,j)} \tilde{B}_{ij,kl}(u_{kl}, u_{ij}) + I_{ij},
\end{aligned} \tag{6}
$$

$$
y_{ij}(t) = f(x_{ij}), \tag{7}
$$

$$
1 \le i \le M, 1 \le j \le M,
$$

x_{ij}, y_{ij}, u_{ij} refer to the state, output and input voltage of a cell $C(i,j)$; $C(i,j)$ refers to a grid point associated with a cell on the 2-D grid, $C(k,l) \in N_r(i,j)$ is a grid point (cell) in the neighborhood within a radius r of the cell $C(i,j)$, I_{ij} is an independent current source. $\tilde{A}$ and $\tilde{B}$ are nonlinear cloning templates, which specify the interactions between each cell and all its neighbor cells in terms of their input, state, and output variables. Moreover, as we mentioned above the cloning template has geometrical meanings which can be exploited to provide us with geometric insights and simpler design methods.

Now in terms of definition 2 we can present the dynamical systems describing CNNs. For a general CNN whose cells are made of time-invariant circuit elements, each cell $C(ij)$ is characterized by its CNN cell dynamics :

$$
\dot{x}_{ij} = -g(x_{ij}, u_{ij}, I^s_{ij}), \tag{8}
$$

where $x_{ij} \in \mathbf{R}^m$, u_{ij} is usually a scalar. In most cases, the interactions (spatial coupling) with the neighbor cell $C(i+k, j+l)$ are specified by a CNN synaptic law:

$$
\begin{aligned}
I^s_{ij} &= A_{ij,kl}x_{i+k,j+l} + \\
&+ \tilde{A}_{ij,kl} * f_{kl}(x_{ij}, x_{i+k,j+l}) + \\
&+ \tilde{B}_{ij,kl} * u_{i+k,j+l}(t).
\end{aligned} \tag{9}
$$

The first term $A_{ij,kl}x_{i+k,j+l}$ of (9) is simply a linear feedback of the states of the neighborhood nodes. The second term provides an arbitrary nonlinear coupling, and the third term accounts for the contributions from the external inputs of each neighbor cell that is located in the N_r neighborhood.

It is known that some autonomous CNNs represent an excellent approximation to nonlinear partial differential equations (PDEs). The intrinsic space distributed topology makes the CNN able to produce real-time solutions of nonlinear PDEs. Consider the following well-known PDE, generally referred to us in the literature as a reaction-diffusion equation:

$$\frac{\partial u}{\partial t} = f(u) + D\nabla^2 u,$$

where $u \in \mathbf{R}^N$, $f \in \mathbf{R}^N$, D is a matrix with the diffusion coefficients, and $\nabla^2 u$ is the Laplacian operator in $\mathbf{R}^2$. There are several ways to approximate the Laplacian operator in discrete space by a CNN synaptic law with an appropriate A-template.

In our case the CNN model of CD equation (5) is:

$$\dot{u}_j(t) = b * A_1 * u_j + c * u_j, 1 \leq j \leq N, \tag{10}$$

where $A_1 = (1, -2, 1)$ is one dimensional discretize Laplacian CNN template.

In this section we will introduce an approximative method for studying the dynamics of CNN model (9), based on a special Fourier transform. The idea of using Fourier expansion for finding the solutions of PDEs is well known in physics. It is used to predict what spatial frequences or modes will dominate in nonlinear PDEs. In CNN literature this approach, has been developed for analyzing the dynamics of CNNs with symmetric templates [1].

We shall investigate the dynamic behavior of CNN model (10) by use of Harmonic Balance Method well known in control theory and in the study of electronic oscillators [3] as describing function method. The method is based on the fact that all cells in CNN are identical [1], and therefore by introducing a suitable double transform, the network can be reduced to a scalar Lur's scheme [3].

We shall present the algorithm briefly:

1. Apply the double Fourier transform:

$$F(s,z) = \sum_{k=-\infty}^{k=\infty} z^{-k} \int_{-\infty}^{\infty} f_k(t) exp(-st) dt,$$

to the CNN equation (10).

2. Find the transform function $H(s,z) =$, where $s = i\omega_0$, $z = exp(i\Omega_0$, $i = \sqrt{-1}$, ω_0 is a temporal frequency, Ω_0 is a spatial frequency.

3. Look for possible solutions of (10) in the form:

$$u_j = U_{m_0} sin(\omega_0 t + j\Omega_0)$$

4. The amplitude U_{m_0}, the temporal frequency ω_0 and the spatial frequency Ω_0 are unknowns to be determined.

5. Our CNN model (10) is a finite circular array of N cells we have finite set of spatial frequencies:

$$\Omega_0 = \frac{2\pi k}{N}, 0 \leq k \leq N-1. \tag{11}$$

Based on the above considerations the following proposition hold:

Proposition 1 *CNN model (10) of the convection-diffusion problem (5), consisting of circular array of N cells, has state solution $u_j(t)$ with a finite set of spatial frequencies $\Omega_0 = 2\pi k/N$, $0 \leq k \leq N-1$.*

We obtain the following simulation results for different values of the parameters β and γ:

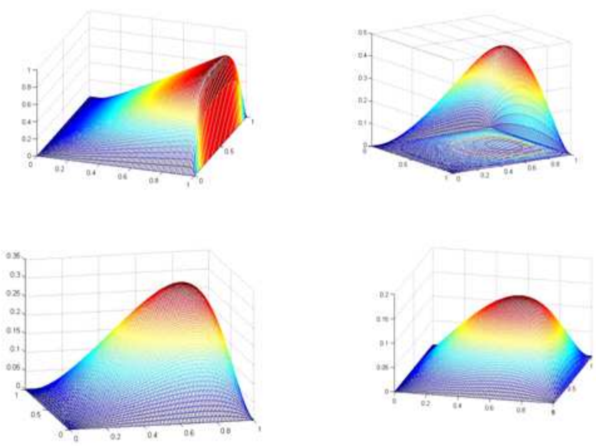

Fig.2. Simulations of the CNN algorithm for CD problem (5).

Example 1. Consider the following singularly perturbed boundary value problem

$$-\varepsilon \Delta u + b u_x + c u = 0, \quad \text{in} \quad \Omega \equiv (0,1)^2 \tag{12}$$
$$u = 0, \quad \text{on} \quad \partial\Omega.$$

In order to construct a robust numerical method for the considered problem, it is of key interest to have information on a behavior of the solution. The state equation of the CNN model of (12) is:

$$-\varepsilon A_1 * u_i + b * A_1 * u_i + c * u_i = 0 \tag{13}$$

Applying CNN algorithm we obtain the following simulation results:

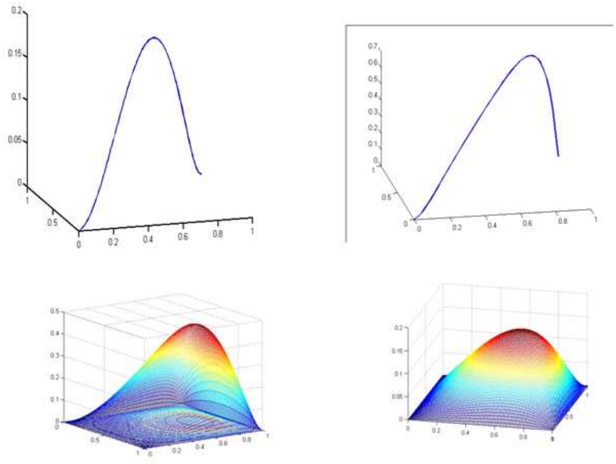

Fig.3. Simulation of the CNN algorithm for the problem (12).

Example 2. We consider the following example of (4) on the layer-adapted mesh:

$$-\varepsilon \Delta u + (1 + x^3)u_x + (1 + xy)u = 0, in\Omega \tag{14}$$

$$u = 0, on\partial\Omega$$

CNN model of the above system is:

$$-\varepsilon A_1 * u_i + (1 + x^3)A_1 * u_i + (1 + xy)u_i = 0 \tag{15}$$

$$1 \leq i \leq N$$

We obtain simulation results of (14) on Fig.4.

Remark 1. We consider a CNN programmable realization allowing the calculation of all necessary processing steps in real time. The network parameter values of CNN model, described by the dynamical system (11), are determined in a supervised optimization process.During the optimization process the mean square error is minimized using Powell method and Simulated Annealing [8]. The results are obtained by the CNN simulation system MATCNN applying 4th order Runge-Kutta integration.

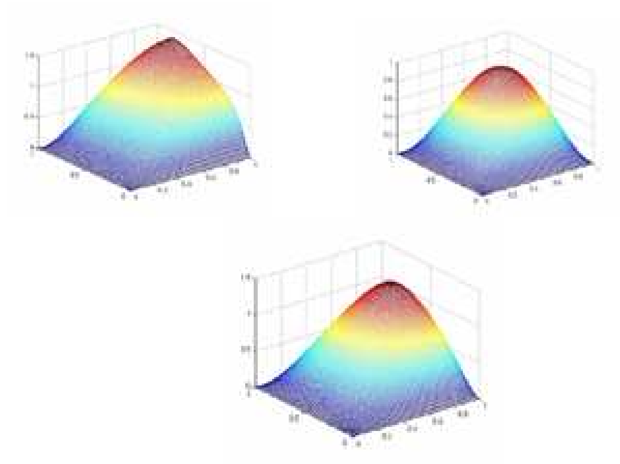

Fig.4 Simulation of the CNN algorithm for the problem (14).

References

[1] Chua L.O., Yang L., Cellular Neural Network: Theory and Applications, *IEEE Trans. CAS*, vol. 35, pp. 1257-1290, Oct. 1988.

[2] Chua L.O., Hasler M., Moschytz G.S., Neirynsk J., Autonomous cellular neural networks: a unified paradigm for pattern formation and active wave propagation, *IEEE Trans. CAS-I*, vol. 42, N 10, pp. 559-577, Oct. 1995.

[3] Mees A.I., *Dynamics of Feedback Systems*, London, England:Wiley, 1981.

[4] P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion. In: Proceedings of IEEE Computer Soc. Workshop and Computer Vision, 1987, pp. 16-22.

[5] B.Roska and F.S.Werblin, Vertical Interactions across Ten Parallel, Stacked Representations in the Mammalian Retina, Nature, vol. 410, pp. 583-587, 2001

[6] Y. Shih, H. C. Elman, Modified streamline diffusion schemes for convection - diffusion problems, Computational Methods in Applied Mechanics and Engineering, 1998.

[7] A.Slavova,Cellular Neural Networks: Dynamics and Modelling, Kluwer Academic Publishers, 2003.

[8] R. Tetzlaff, F. Gollas, Modeling complex systems by reaction-diffusion Cellular Nonlinear Networks with polynomial Weight-Functions,Proc. IEEE CNNA2005, 2005.

Institute of Mathematics
Bulgarian Academy of Sciences
Sofia 1113, Bulgaria
slavova@math.bas.bg

Department of Informatics
Rousse University
Rousse 7017, Bulgaria
viktoriqi$_7$8@*abv.bg*