2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Spatial-Temporal Level Set Algorithms on CNN-UM

Gábor J. Tornai†, György Cserey‡ and Ádám Rák†

†Faculty of Information Technology, Pázmány Péter Catholic University
Práter u. 50/A, H-1083 Budapest, Hungary
‡Hungarian Academy of Sciences - Pázmány Péter Catholic University - Semmelweis University
Infobionic and Neurobiological Plasticity Research Group
Práter u. 50/a. Budapest, H-1083, Hungary
Email: torgaja@digitus.itk.ppke.hu, cserey@itk.ppke.hu, rakad@digitus.itk.ppke.hu

Abstract—In this paper we propose to describe 2D and 3D spatial-temporal algorithms based on level sets using the advantages of the local connectivities in the cellular neural-nonlinear network (CNN). The primary goal of this paper is to contribute a development and parallel implementation of fast global level-set algorithms [1] using the idea of a local interaction based level-set algorithms. Our CNN algorithms can handle more initial curves which can fuse or keep distance according to the requirements. This could be a very good base to achieve fast image segmentation and object detection. Finally 2D and 3D simulation results are presented.

1. Introduction

The localisation of the object boundaries is a challenging and important task in many imaging problem such as segmentation and tracking. The level set method has been very popular in the recent years. The original idea was introduced by Osher and Sethian [2] in 1988 - how can we trace the motion of surfaces or curves in a velocity field. The velocity field can change during the time and can depend on space or the temporary geometry. This method can handle topological changes automatically meaning all object can split into two or more sepatare objects and of course the opposite also happens during the time evolution. Mainly to solve these problems we must face with partial differential equations (PDE). The von Neumann architecture can't dissolve the bottleneck of such computing cost.

The cellular neural-nonlinear network (CNN) [3] has solved many interesting problems in the past few years. The massively parallel model's computing power is far beyond the single instruction single data stream (SISD) architecture in these kind of operations and tasks.

Several researchers and projects succeeded to find solvation for 2D image processing problems using level set based algorithms implemented on CNN architecture [4, 5]. Nowadays many-processor architectures and the presence of modern 3D medical imaging systems inspires of the development such systems and algorithms, which can process 3D volume data flows in real time. Our aim is to achieve fast detection of objects in 3D volume flows using level set based algorithms.

This paper is organized as follows. First a general overview of the level set methods and the implicit curve evolutions are given. Then we go into the details of both 2D and 3D spatial-temporal level set algorithms and the implementation issues. Finally simulation results are presented.

2. Level sets - Motion of implicitly represented curves under velocity field - general view

In this section we give a general overview on the basics of the level set method. The original idea was introduced in 1988 by Osher and Sethian [2] to track the motion of an interface, or curve C in R^n mainly in R^2 or in R^3. For convenience, we use the 2D terminology "curve" to denote the surfaces in R^n, where $(n \geq 2)$. C bounds an open region Ω, which could be multiply connected. Our goal is to obtain the subsequent motion of C under a velocity field $\vec{F}$ [6]. The velocity field can depend on position, time and the geometry of the interface (it's normal or it's mean curvature).

Let's define a smooth (at least Lipschitz continuous) function $\phi(x,t)$ that represents the curve as the set where $\phi(x,t) = 0$ and $x = (x_1, x_2 ... x_n)$.

The level set function ϕ has the following properties:

$$\phi(x,t) > 0 \; for \; x \in \Omega \quad (1)$$

$$\phi(x,t) < 0 \; for \; x \notin \Omega \quad (2)$$

$$\phi(x,t) = 0 \; for \; x \in \partial\Omega = C(t) \quad (3)$$

Thus, the interface is to be captured for all later time, by merely locating the set $C(t)$ for which ϕ vanishes. This trivial statement of great significance because topological changes such as breaking and merging are well defined and performed automatically. The motion is analyzed by convecting the ϕ values (levels) with the velocity field $\vec{F}$. The elementary equation is:

$$\frac{\partial \phi}{\partial t} + \vec{F} \cdot \nabla \phi = 0 \quad (4)$$

Here $\vec{F}$ is the velocity on the curve, and is arbitrary elsewhere. Actually only the normal component of $\vec{F}$ is

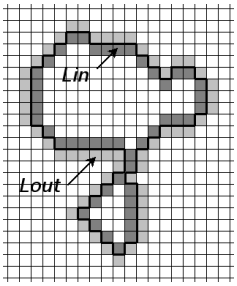

Figure 1: Curve C (solid black line) on regular grid represented implicitly by L_{in} and L_{out}. Changing L_{in} and L_{out} the motion of the curve can be obtained.

needed: $F_N = \vec{F} \cdot \frac{\nabla\phi}{|\nabla\phi|}$

$$\frac{\partial\phi}{\partial t} + F_N|\nabla\phi| \tag{5}$$

And the time evolution of the curve C is according to the following equation:

$$\frac{dC(t)}{dt} = F_N \cdot \vec{N} \tag{6}$$

Where $\vec{N}$ is the outward normal of the curve C.

2.1. Motion of curves on regular grid

In the level set method the curve C is implicitly represented as the zero level set of the function ϕ defined over a regular grid as shown in Figure 1. Henceforth we choose the level set function ϕ to be negative inside the curve and positive outside. We assume that ϕ is defined over a domain D$\in R^K$ ($K \geq 2$), without the loss of generality we assume that the grid is sampled uniformly. Two "neighboring" sets L_{in} and L_{out} can be unambiguously defined on this grid as follows:

$$L_{in} = \{\mathbf{x}|\phi(\mathbf{x}) < 0 \; and \; \exists\mathbf{y} \in N(\mathbf{x}) \; that \; \phi(\mathbf{y}) > 0\} \tag{7}$$
$$L_{out} = \{\mathbf{x}|\phi(\mathbf{x}) > 0 \; and \; \exists\mathbf{y} \in N(\mathbf{x}) \; that \; \phi(\mathbf{y}) < 0\} \tag{8}$$

Where $N(\mathbf{x})$ is defined as:

$$N(\mathbf{x}) = \{\mathbf{y} \in D | \sum_{k=1}^{K} |y_k - x_k| = 1\} \; \forall\mathbf{x} \in D \tag{9}$$

Two additional sets are required: F_{in} and F_{out}, they have the following properties:

$$F_{in} = \{\mathbf{x} \in D|\phi(\mathbf{x}) < 0 \wedge \mathbf{x} \notin L_{in}\} \tag{10}$$
$$F_{out} = \{\mathbf{x} \in D|\phi(\mathbf{x}) > 0 \wedge \mathbf{x} \notin L_{out}\} \tag{11}$$

F_{in} contains the points inside the curve and F_{out} contains the points outside the curve but not adjoining curve C. Naturally these sets can be described as follows:

$$L_{in} = \{\mathbf{x}|\mathbf{x} \in \Omega \; and \; \exists\mathbf{y} \in N(\mathbf{x}) \; that \; \mathbf{y} \in (D \setminus \Omega)\} \tag{12}$$

$$L_{out} = \{\mathbf{x}|\mathbf{x} \in (D \setminus \Omega) \; and \; \exists\mathbf{y} \in N\mathbf{x} \; that \; \mathbf{y} \in \Omega\} \tag{13}$$
$$F_{in} = \{\mathbf{x}|\mathbf{x} \in \Omega \; and \; \exists\mathbf{y} \in N(\mathbf{x}) \; that \; \mathbf{y} \in L_{in}\} \tag{14}$$
$$F_{out} = \{\mathbf{x}|\mathbf{x} \in (D \setminus \Omega) \; and \; \exists\mathbf{y} \in N(\mathbf{x}) \; that \; \mathbf{y} \in L_{out}\} \tag{15}$$

By switching grid points from one set to the other ($L_{in} \Leftrightarrow L_{out}$) the motion of the curve C can be obtained. Of course updating F_{in} and F_{out} is also necessary. To avoid confusion we note that the grid points in L_{in} and L_{out} are *not* on curve C. They are to define the neighborhood of the curve on the discrete grid and the curve is still represented implicitly by ϕ.

2.2. Local method to global method conversion, CNN concept

In [1, 7] Y. Shi and C. Karl developed a fast algorithm using local rules working on L_{in} and L_{out} and they scanned through the two sets whether there are points to be switched. Naturally our method works with global operators and spatial-temporal dynamics. It has 5 phases: **1.** initialization, **2.** preprocessing, **3.** accretion phase (evolution, update F_{out}, eliminate redundant parts from L_{in} update F_{in}), **4.** diminution phase (evolution, eliminate redundant parts from L_{out} update F_{out}, update F_{in}), **5.** check stopping condition.

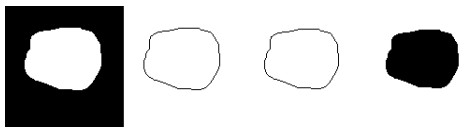

Figure 2: The initial sets: F_{out}, L_{out}, L_{in}, F_{in}

To utilize the advantages laying in the CNN paradigm a lot of curves have to be on the initial images F_{out}, L_{out}, L_{in}, F_{in}. The complexity of the algorithm is $O(n)$ where n is the size of one dimension of the data.

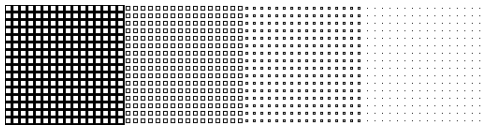

Figure 3: To maximize the speedup the initial images contain a lot of small curves.

3. 2D spatial-temporal level-set algorithms

In this section we unfold the details of each step in the implemented/simulated 2D algorithms. We have implemented two subtypes of the algorithm. The first is the "simple" level set meaning the curves can merge and split as the velocity field affect on them. The second is a bit more complex. Let's assume that we know the number of objects to be detected and we also have their rough location. Under these circumstances detecting multiple objects is pos-

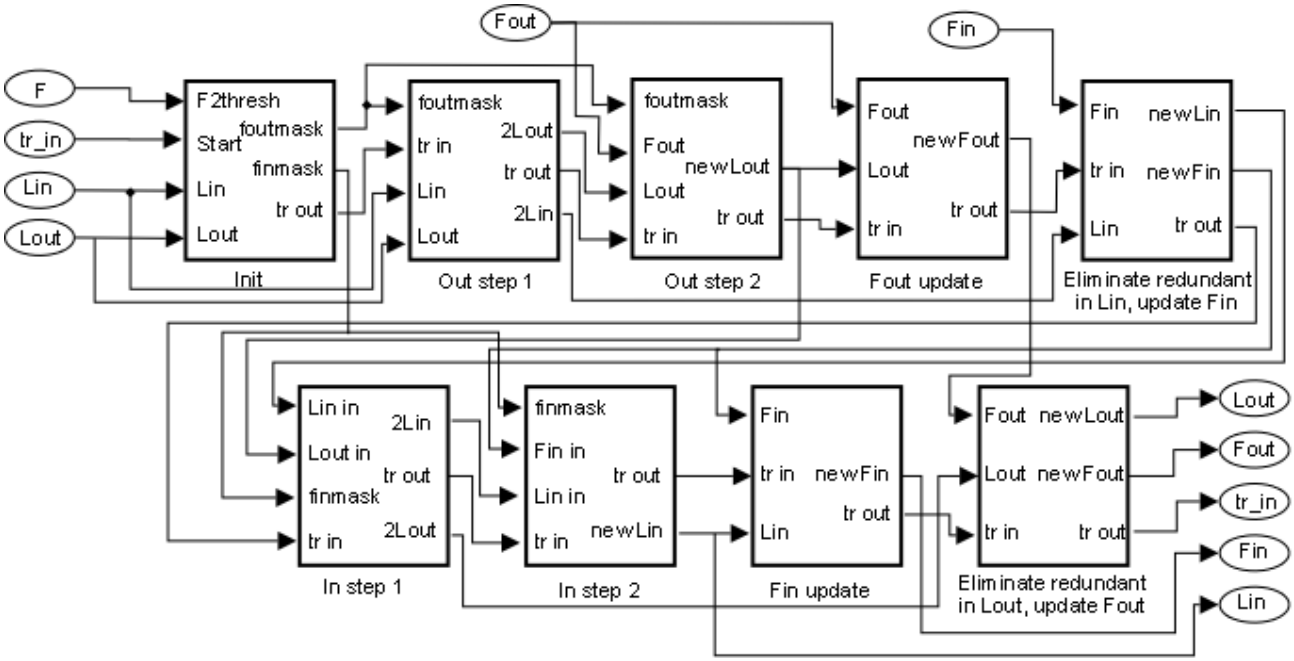

Figure 4: The main blocks of the kernel

sible meaning the separate initialized objects keep distance (KD).

We used the simplest templates: threshold, erosion and dilatation; and the following logics: "and", "or", "andnot".

3.1. Simple algorithm

The simple algorithm is fast but cannot distinguish the objects. We describe it in template and logic level. We work on binary images. The algorithm has 5 input image. Four of the five images are the initial state of the ϕ more precisely: F_{in}, L_{in}, L_{out}, F_{out}. The last one is the real input.

1. *initialization and preprocessing:* In this part F_{inmask} and $F_{outmask}$ are computed. $F_{outmask}$ is the mutual part of F and L_{out} and base of the accretion phase. $F_{outmask}$ is the mutual part of $\neg F$ and L_{out} and base of the diminution phase.

2. *accretion phase:* This is one major part of the algorithm. By means of $F_{outmask}$ new L_{out} is generated using dilation and logical operators. Then $F_{out}$is updated. Lastly redundant points from L_{in} are eliminated and F_{in} is updated.

3. *diminution phase:* This is the second major part of the algorithm. Dilating F_{inmask} and then using logical operators and the next piece of L_{in} is obtained. Then comes F_{in} afterward redundant points of L_{out} are eliminated lastly F_{out} is refreshed.

4. *stopping conditions:* There are two stopping condition. If either is fulfilled, the algorithm stops. Otherwise, the algorithm steps again on phase 2.

- $F(\mathbf{x}) \leq 0 \; \forall \, \mathbf{x} \; \in L_{out}$ and $F(\mathbf{x}) \geq 0 \; \forall \, \mathbf{x} \; \in L_{in}$
- The pre-specified number of iteration is reached.

3.2. KD algorithm

This variant can detect multiple objects at the expense of the running time. In the case of N separate objects the algorithm needs 4N piece of the initializing images $(F^1_{in}, F^2_{in}, \ldots F^N_{in}; L^1_{in} \ldots L^N_{in}; L^1_{out} \ldots L^N_{out}; F^1_{out} \ldots F^N_{out})$ and the running time is N times longer compared to the simple algorithm (initialize / preprocess to object 1 / one step of the simple algorithm / preprocess to obj. 2 / one step of the simple algorithm / . . . / preprocess to object N / one step of the simple algorithm / check stopping condition).

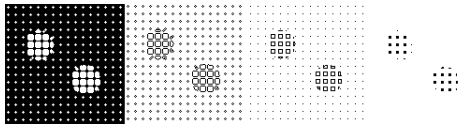

Figure 5: 1^{st} iteration F_{out}, L_{out}, L_{in}, F_{in}

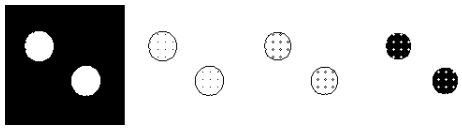

Figure 6: 4^{th} iteration F_{out}, L_{out}, L_{in}, F_{in}

Table 1: iterations required depending on the number of initial curves

no. of curves	1	2x2	4x4	5x5	8x8	16x16
no. of iterations	31	16	17	15	11	5

4. 3D spatial-temporal level-set algorithms

In this section we describe a 3D CNN model. Here one cell has 27 neighboring cell, including itself too. The algorithm is the same described in the previous section using logical operations, erosion and dilation but the major topics are the templates to realize these operations. Here the templates are 3 dimensional arrays. Using nearest neighborhood the templates have 3^3 elements according to the 3 spatial dimension (height, width, depth). $3 \times 3 \times 3$ B template of the 6 connection erosion is:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

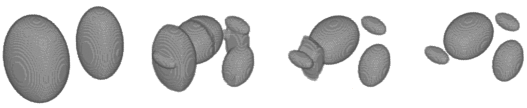

Figure 7: 3D evolution of our CNN algorithm

5. Simulation results

With our colleguages, we implemented the algorithm in Simulink using the SimCNN toolbox [8]. The simulation results are presented in Figure 7-10. It can be seen that in the case of a lot of small initial curves the transient is much faster, it lasts only for 5 iteration.

In the case of an image flow, where the frames are highly correlated, the algorithm after a sort delay, which comes from the usually big difference between the zero level set of the function ϕ and the velocity field (the image to be detected or segmented), it converges after a few steps compared to the size of the image. In the case of 128x128 image size after the first image the required number of iterations are maximum 5.

6. Conclusion

In this paper we presented 2D and 3D level set based CNN algorithms for multiple object detection. In 3D case, instead of using 2D layers we applied 3D CNN templates. Our simulation results show the applicability of the algorithms. In the future we are planning to implement the algorithm on a GPU (graphical programming unit) where real time processing can be achieved.

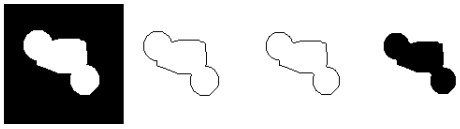

Figure 8: 15^{th} iteration F_{out}, L_{out}, L_{in}, F_{in}

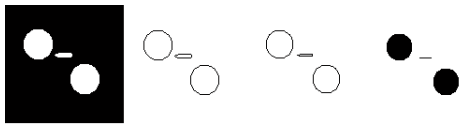

Figure 9: 31^{st} iteration F_{out}, L_{out}, L_{in}, F_{in}

Acknowledgments

The Operational Program for Economic Competitiveness (GVOP KMA) is gratefully acknowledged. The authors are also grateful to Gergely B. Soós and József Veres for the discussions and their suggestions.

References

[1] Y. Shi and W. Karl, "A Fast Level Set Method Without Solving PDEs," *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2005.*, vol. 2, 2005.

[2] S. Osher, J. Sethian, and L. R. Center, *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*. National Aeronautics and Space Administration, 1987.

[3] L. O. Chua and T. Roska, *Cellular neural networks and visual computing, Foundations and applications*. Cambridge University Press, 2002.

[4] Gy. Cserey, Cs. Rekeczky, and P. Földesy, "PDE based histogram modification with embedded morphological processing of the level-sets," *Journal of Circuits, System and Computers*, vol. 12, no. 4, pp. 519–538, 2003.

[5] D. Vilarino and C. Rekeczky, "Pixel-level snakes on the CNNUM: algorithm design, on-chip implementation and applications," *Int. Journal of Circuit Theory and Applications*, vol. 33, no. 1, pp. 17–51, 2005.

[6] S. Osher and R. Fedkiw, "Level set methods- An overview and some recent results," *Journal of Computational Physics*, vol. 169, no. 2, pp. 463–502, 2001.

[7] Y. Shi, *Object based dynamic imaging with level set methods*. PhD, Boston Univ. College of Eng., 2005.

[8] G. Soós, A. Rák, J. Veres, and G. Cserey, "GPU powered CNN simulator (SIMCNN) with graphical flow based programmability," in *IEEE International Workshop on Cellular Neural Networks and their Applications, (CNNA 2008)*, 2008. sent.