International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 7 Issue 3 Ser. I ǁ 2019 ǁ PP. 51-61

Cellular Neural Network and Application in Animation Proccessings

Thanh-Tung Cap[1], Hien-Trinh Nguyen[2]
[1]*Department of Mathematics, University of Education, Thai Nguyen, Viet Nam*
[2]*Department of Computer Science, University of Information And Communication Technology, Thai Nguyen, Viet Nam*
Corresponding Author: Thanh-Tung Cap

ABSTRACT: *The processing of motion images, in healthcare in general and in scanning chest images in particular, has an extremely important meaning in reality because it is necessary to accurately determine location, size and rule of movement of objects (tissue, tumour…) for diagnosis, surgery or radiation treatment. The processing often makes use of IR (image registration) techniques, the virtue of which is to determine compatibility of pixels in 2 consecutive images in an image series. There are numerous methods of image registration but the popularly applied one is based on the criteria of optical flow speed. With regard of chest images, especially for analysis and diagnosis of lung tumour, we often have considerable discrepancies due to the complexity of movement (random and quick variations in respiration) and flexibilty of cells (which leads to periodical change of brightness of each pixels based on respiratory rhythm and various diffraction). Some approaches are proposed such as CCLG (Combine Compressible Local Global formulation) to deal with the problem, but only with small displacement optical flow. In our research we study the approach of LDOF (Large Displacement Optical Flow) and the solution of CNN technology for parallel processing of large data.*
Keywords:*Cellular Neural Network, Optical Flow, Image Registration, CNN, LDOF, Medical Image*

Date of Submission: 01-10-2019 Date of acceptance: 16-10-2019

I. INTRODUCTION

Motion image analysis is a very important and complicated part in image processing, especially for healthcare image. In image processing, for studying motion objects, IR (image registration) approach are often proposed. Image registration is finding compatibility of every pixel in 2 succeeding images of a collected image chain". The image chain is taken in subsequent moments during object surveying. There are many techniques applied in IR but the mainly used criteria is determing pixel based on speed of optical flow. Optical flow expresses speed of motion object in a image. Motion image processing in healthcare paid a special attention to optical flow mathematic model, especially scanned chest imge to find and analyse activities of tissue, tumour, defect… and recommend suitable treatment regiment.

Cellular neural networks (CNN) has been developed since 1988 [1][2] and has gained considerable achievements, CNN is a neuron network and is designed as a simulation of retina in human eye. The largest diffrence between CNN and ordinary neuron network lies in the fact that each cell only interacts and exchange information directly with its neighbours and this itnteraction is neutral, constant (identical) and isotropic, so that it is easy to realize those relations and processes by VLSI. The most popular application of CNN is to solve real time image processing problem because CNN is very efficient in parrallel processing for large data volume.
In this article, we analyse and propose optical flow analysing model based on CNN techonogy which provides faster and more accurate solution than the traditional approach based on sequence calculation.

II. METHODOLOGY

2.1 Basic equation of Optical Flow and discrete process

As a common sense, in motion image processing, it is important to determine compatibility of pixel between two consecutive images. The speed of each pixel in motion picture sequence (Optical flow) is an important characteristic to solve this problem. For scanned image, the strength of light of each pixel I (x, y, t) is correspondent with the density of cell tissue $\rho(x,y,t)$ where x, y is coordinate of pixel in space Ω and t is time.So the mathematic expression of pixel is as follows :

$$I(\mathbf{x},t) \sim r(\mathbf{x},t) \qquad (1)$$

Where: $t \in [0,1]; \qquad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \in \Omega; \qquad \Omega \in \square^2$

In most image processing methodes for scanned images, it is assumed that light strength of each pixel, corresponding with tissue density, is constant with time: $I(\mathbf{x},t) \square \rho(\mathbf{x},t) = const$, that means:

$$I(x+u.dt, y+v.dt, t+dt) = I(x,y,t) \qquad (2)$$

Taylor deployment for left side:

$$I(x+u.dt, y+v.dt, t+dt) = I(x,y,t) + I'(x,y,t)\frac{h}{1!} + ... + O(h^n)$$

Eliminating high-derivative components (considered as Extremely small) and assuming h=1(pixell)the equation (2) will be:

$$I'(x,y,t) \approx 0 \quad <=> \; I_x \frac{\partial x}{\partial t} + I_y \frac{\partial y}{\partial t} + I_t \approx 0$$

$$I_x.u + I_y.v + I_t \approx 0$$

it mea*n*s that : $\nabla I.V + I_t \approx 0 \qquad (3)$

Where : $\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}; \qquad V = \begin{bmatrix} u \\ v \end{bmatrix}$

With equation (3) we can not find unique solution for optical flow problem (an equation with two variables u,v of speed V). We must add another condition for this problem. Horn- Schunk assume that speed of neighbouring pixels are similar, or speed variations of pixels are flexbible [9], so the authors introduced flexibility coefficient (smoothing) α for speed calculation problem. By idea of Horn- Schunk, we merge equation (3) with assumption of unchangable speed of neighbouring pixels in to convex function (4) and find value of V to minize this function:

$$\min_{\mathbf{v}} \frac{1}{2}\int_{\Omega} h(x)^2 + \alpha^2 \sum_{i=1}^{2} \|\nabla v_i\|^2 d\mathbf{x} \qquad (4)$$

Where $h(x) = \tilde{N} I.V + I_t$

$\alpha > 0$ is smoothing coefficient. We can find V to minize function (4) by Variational Calculus. As (4) is a convex square functional so the necessary and sufficient condition for optima V is to satisfy corresponding Euler-Lagrange equation:

$$\frac{\P f}{\P x} - \frac{d}{dt}\frac{\P f}{\P v} = 0 \qquad (5)$$

The boundary condition is :

$$(\nabla v_i + Ih\mathbf{e}_i)^T \mathbf{n} = 0; \;\; i = 1,2 \qquad (6)$$

Where $\mathbf{e}_i$ is the unit vector of the axis I,**n** is the normal vector to the boundary surface.In (5) and (6) we use those marks:

$$f = h^2 + \alpha^2 \sum_{i=1}^{2} \|\nabla v_i\|^2$$

$$(h = \nabla I.V + I_t)$$

After calculating the derivatives of (5) and reducing, the Euler-Lagrange equation has the form:

$$\begin{cases} I_x(I_x.u + I_y.v + I_t) - \alpha^2 \nabla^2 u = 0 \\ I_y(I_x.u + I_y.v + I_t) - \alpha^2 \nabla^2 y = 0 \end{cases} \qquad (7)$$

Where ∇^2 is the Laplace operator, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

Can rewrite (7) in short form : $h.\nabla I - \alpha^2 \begin{bmatrix} \nabla^2 u \\ \nabla^2 v \end{bmatrix} = 0$

In [9], Horn - Shunk presented the approximate calculation for image derivatives as follows:

$$I_x = \frac{1}{4}(I_{i,j+1,t} - I_{i,j,t} + I_{i+1,j+1,t} - I_{i+1,j,t} + I_{i,j+1,t+1} - I_{i,j,t+1} + I_{i+1,j+1,t+1} - I_{i+1,j,t+1})$$
$$I_y = \frac{1}{4}(I_{i+1,j,t} - I_{i,j,t} + I_{i+1,j+1,t} - I_{i,j+1,t} + I_{i+1,j,t+1} - I_{i,j,t+1} + I_{i+1,j+1,t+1} - I_{i,j+1,t+1}) \quad (8)$$
$$I_t = \frac{1}{4}(I_{i,j,t+1} - I_{i,j,t} + I_{i+1,j,t+1} - I_{i+1,j,t} + I_{i,j+1,t+1} - I_{i,j+1,t} + I_{i+1,j+1,t+1} - I_{i+1,j+1,t})$$

In addition, the Laplace operator for the velocity components $\tilde{N}_u^2$, $\tilde{N}_v^2$, calculated as follows:

$$\tilde{N}^2 u = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) - u_{ij} \quad (9)$$

Put $\bar{u}_{ij} = \frac{1}{4}(u_{i-1,j} + u_{i,j-1} + u_{i,j+1} + u_{i+1,j})$

Calculate the same way for $\nabla_v^2$and $\bar{v}_{ij}$. Substitute (9) into (7) we have:

$$\begin{cases}(I_x^2 + \alpha^2).u_{ij} + I_x(I_y.v_{ij} + I_t) - \alpha^2\bar{u}_{ij} = 0\\ (I_y^2 + \alpha^2).v_{ij} + I_y(I_x.u_{ij} + I_t) - \alpha^2\bar{v}_{ij} = 0\end{cases} \quad (10)$$

2.2 CNN network and compute Optical Flow based on CNN

The CNN network is a dynamically linked system of identical elements, each element is on an MxN grid node and has the equations described as follows:

– Status equation:

$$\dot{x}_{ij} = -x_{ij} + \sum_{k,l \in S_r} A_{ij,kl}(.y_{kl}(t).y_{ij}(t)) + \sum_{k,l \in S_r} B_{ij,kl}(u_{kl}(t).u_{ij}(t)) + z \quad (11)$$

– Output equation:

$$y_{ij}(t) = f(x_{ij}) \quad (12)$$

u_{ij}, x_{ij}, y_{ij} are accordingly input, state and output of cell at the grid node (i,j) in the MxN network. $k,l \in S_r$ denotes the parameters of the grid node at (k,l) which is neighbours of (i, j) in influence area S_r. The influence area S_r is determined by the radius r, for example r=1 then S_r is dimension 3x3 in 2D space (In 3D it is a sphere with radius r). $A_{ij,kl}$; $B_{ij,kl}$; z are accordingly are feedback, control ans bias tempates of cell (i,j). $A_{ij,kl}$; $B_{ij,kl}$ is for general case of neural and non-linear templates.

The documents [1] [2] have all clearly demonstrated the convergence of the solution and the stability of the system.

+ Case $A_{ij,kl}$; $B_{ij,kl}$ are single and linear template, they are rewritten as$A_{ij,kl}$, $B_{ij,kl}$, and :

$$\dot{x}_{ij} = -x_{ij} + \sum_{k,l \in S_r} A_{ij,kl}.y_{kl}(t) + \sum_{k,l \in S_r} B_{ij,kl}.u_{kl}(t) + z \quad (13)$$

$$y_{ij} = \frac{1}{2}\left(|x_{ij}(t)+1| - |x_{ij}(t)-1|\right) \quad (14)$$

Figure 1 shows a diagram of a cell, with a radius r = 1.

+ Case DCNN (Delayed Cellular Neural Network), formula (11) converts to:

$$\dot{x}_{ij} = -x_{ij} + \sum_{k,l \in S_r} A_{ij,kl}.y_{kl}(t) + \sum_{k,l \in S_r} A^t{}_{ij,kl}.y_{kl}(t-t) + \sum_{k,l \in S_r} B_{ij,kl}u_{kl}(t)) + \sum_{k,l \in S_r} B^t{}_{ij,kl}.u_{kl}(t-t) + z \quad (15)$$

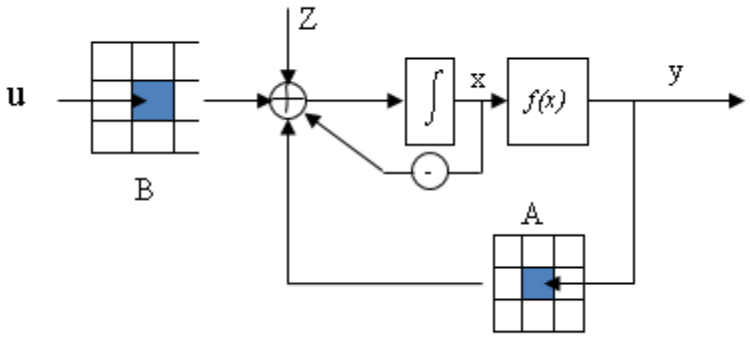


Figure 1. Diagram of a cell.

In the application, there are some types of linear CNN as follows:

+ CNN Zero feedback: A=0

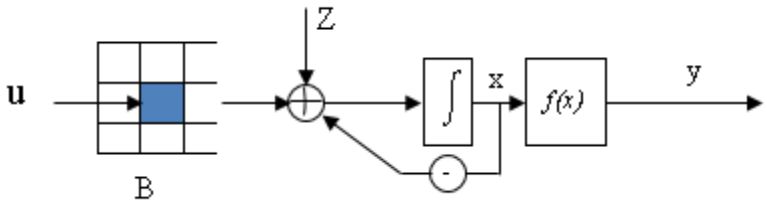


Figure 2. Activity diagram of a cell without feedback.

With z=0, equation (15) is:

$$\dot{x}_{ij} = -x_{ij} + \sum_{k,l \in S_r} B_{ij,kl}.u_{kl}(t) + \sum_{k,l \in S_r} B^t{}_{ij,kl}.u_{kl}(t-t) \quad (16)$$

+ CNN autonomous (Zero input): B=0

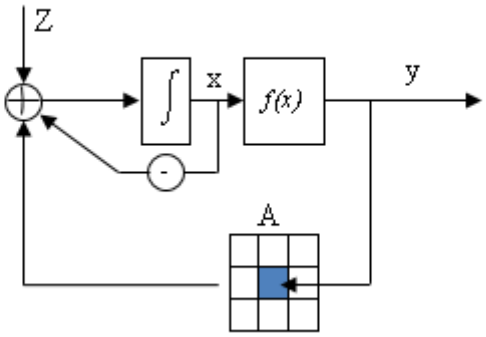


Figure 3. Activity diagram of a cell without input (Autonomous)

+ CNN uncoupled, template A has $a_{ij}=0$ if $i \neq j$.

Going back to the discrete Euler - Lagrange equation in (10), it is shown that calculating process for (u, v) can be implemented on CNN by finding the stability solution of the following differential system by virtual time as follows:

$$\begin{cases} \dfrac{du_{ij}}{d\tau} = (I_x^{\,2} + \alpha^2).u_{ij} + I_x(I_y.v_{ij} + I_t) - \alpha^2\bar{u}_{ij} \\ \dfrac{dv_{ij}}{d\tau} = (I_y^{\,2} + \alpha^2).v_{ij} + I_y(I_x.u_{ij} + I_t) - \alpha^2\bar{v}_{ij} \end{cases} \quad (17)$$

In this case, we construct a 2-layer CNN network structure (Figure 4) to process 2D images for two components (u, v) of velocity **v**. Deploying the formula (17), we get it:

$$\begin{cases} \dfrac{du_{ij}}{d\tau} = \sum_{kl\in S_r} A^1_{ij,kl}.u_{kl}(t) + \sum_{kl\in S_r} A^{12}_{ij,kl}.v_{kl}(t) + Z^1_{ij} \\ \dfrac{dv_{ij}}{d\tau} = \sum_{kl\in S_r} A^2_{ij,kl}.v_{kl}(t) + \sum_{kl\in S_r} A^{21}_{ij,kl}.u_{kl}(t) + Z^2_{ij} \end{cases} \quad (18)$$

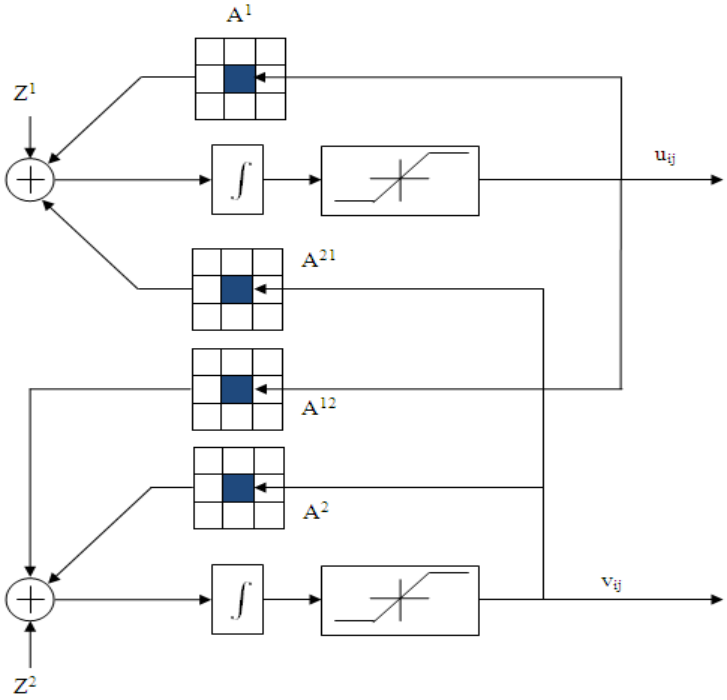


Figure 4. Two-layer CNN model calculates the component velocity (u, v) of the point (ij)

With A^{12} và A^{21} are feedback templates of layer 1 (with u as status of CNN) for layer 2 (with v as status of CNN) and versus. And Z^1, Z^2 are bias of layers 1 and 2. . We can determine weighted connection matrix of cells as follows:

$$A^1 = \begin{bmatrix} 0 & -a^2/4 & 0 \\ a^2/4 & I_x^2 + a^2 & -a^2/4 \\ 0 & -a^2/4 & 0 \end{bmatrix} \quad A^{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_x.I_y & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & -a^2/4 & 0 \\ a^2/4 & I_y^2 + a^2 & -a^2/4 \\ 0 & -a^2/4 & 0 \end{bmatrix} \quad A^{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_y.I_x & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Z_{ij}^1 = I_x.I_t \;; \qquad Z_{ij}^2 = I_y.I_t$$

As shown in above, the calculation of image derivatives I_x, I_y, $I_t$are necessary. We use cell network DCNN (Delayed Cellular Neural Network) to approximately calculate those derivatives. Use formula (16) to calculation for the status x_{ij} for obtaining steady state:

$$\frac{dx_{ij}}{dt} = - x_{ij} + \sum_{k,l\in S_r} B_{ij,kl}.u_{kl}(t) + \sum_{k,l\in S_r} B^t{}_{ij,kl}.u_{kl}(t-t)$$
$$= - x_{ij} + \sum_{k,l\in S_r} B_{ij,kl}.I_{kl}(t) + \sum_{k,l\in S_r} B^t{}_{ij,kl}.I_{kl}(t-t)$$

Put $x = I_x$, we has : $B = B^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 1 \end{bmatrix}$

Put $x = I_y$, we has : $B = B^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & 1 & 1 \end{bmatrix}$

Put $x = I_t$, we has : $B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$; $B^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & -1 & -1 \end{bmatrix}$

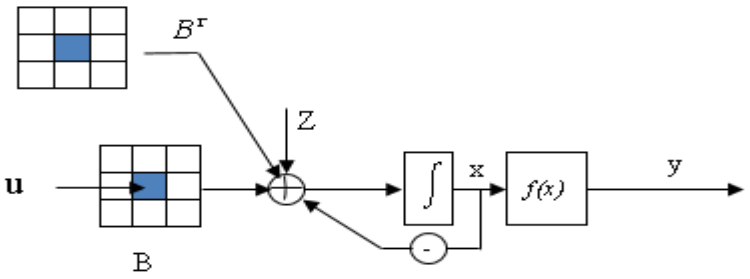


Figure 5. Model of DCNN cell computes the image derivative

We have a two-layer cell network to calculate the image derivative:

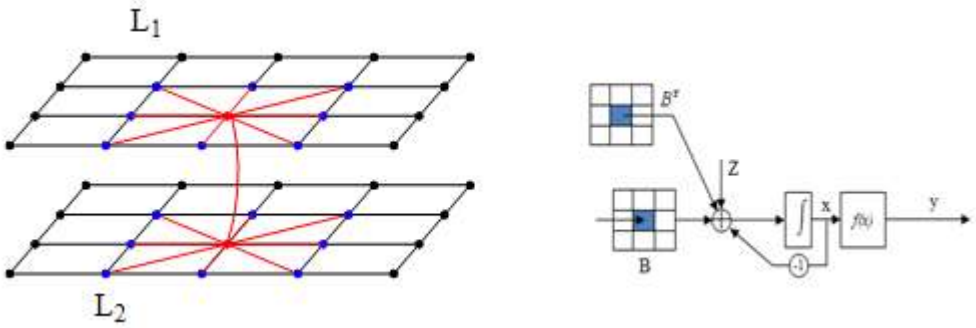


Figure 6. Two layers of CNN calculate the image derivative

In which, layer L1: uses autonomous CNN format, and layer L2: CNN has no feedback.

2.3 Culculate optical flow with large displacement (LDOF)

In solving problem (10) we must often use finite difference methode to approximate image derivatives, whose accuracy depend on displacement of pixel. For large displacement images (that is, images that move too fast during the survey period), it is obvious that the error is too large, leading to image verification (Image Registration - IR) become seriously inaccurate. To solve this problem, there are many solutions that are brought up, together, they often find ways to combine the image verification (IR) process with the image inpainting method.That strategy begins with the idea of incorporating unknown image information into IR process. The most simple example is IR for $P_0$and P_1. We make some modifications by inserting some unknown image data into time space from $P_0$and P_1. This leads to LDOF (Large Displacement Optical Flow) formula:

$$\min_{v,I} \frac{1}{2} \int_H (\tilde{N} I.v)^2 + a^2 \sum_{i=1}^{2} \| \tilde{N} v_i \|^2 + b^2 (I_t)^2 \, dH \qquad (19)$$

With $\begin{cases} I(x(0),0) = P_0(x) \\ I(x(T),T) = P_1(x) \end{cases}$

Here velocity v=(u,v)^T ; $\tilde{v} = [u, v, 1]^T$

$H = W \times [0,T]$; W is the image space domain; $\tilde{N} v_i = \left(\frac{\partial v_i}{\partial x_1}, \frac{\partial v_i}{\partial x_2}, \frac{\partial v_i}{\partial t} \right)^T$

As is well known, the solution of the optimal problem results in the satisfaction of the Euler-Lagrange equation. Through the implementation steps, the following results are obtained:

$$\begin{bmatrix} h.I_x & - a^2 \Delta u \\ h.I_y & - a^2 \Delta v \\ div(\mathbf{f}) & - b^2 p_u \end{bmatrix} = 0 \qquad (20)$$

To solve the system (20) by CNN, we can separate them into two groups of equations:

a)

$$\begin{cases} I_x (I_x.u + I_y.v + I_t) - \alpha^2 \nabla^2 u = 0 \\ I_y (I_x.u + I_y.v + I_t) - \alpha^2 \nabla^2 y = 0 \end{cases}$$

b) $div(h.\tilde{v}) + \beta^2 I_{tt} = 0$

Equations (a) can be solved to compute velocity components (u, v) when knowing image I (and its derivatives in 2D space are: $I_x; I_y$). In contrast, with (b) if we know (u,v) we can also identify image I. From the above comment, to solve the problem, we can construct two modules M_1 and M_2, in which M_1 computes into the velocity **v**=(u,v) satisfies the equation group (a), while M_2 calculates I satisfies the equation group (b). M_1 and M_2 are applied CNN technology to solve PDE differential equations. The algorithm is modeled as shown in Figure 7. The stop condition (end of the loop) is determined by the convergenceflag for each coordinates (element at position (i, j) in image matrix I):

$$convergenflag[i][j] = \begin{cases} \dfrac{dx_{ij}}{dt} = 0 \\ and \;\; y_{kl} = \pm 1 \end{cases}$$

Where x, y : state and output of cell

In fact, when running the program, instead of checking convergenceflag, we just need to select the number of loops large enough.

Designing M1 and setting up algorithm (a) as shown in (3). Now we deploy for M2 (equation (b)):

$$div(h.\tilde{v}) = h.div(\tilde{v}) + \nabla h.\tilde{v} = (I_x.u + I_y.v + I_t).div(\tilde{v}) + \nabla h.\tilde{v}$$

To reduce complexity in calculation, we set h as constant to x,y,t ; then equation (b) becomes :

$(I_x.u + I_y.v + I_t)(u_x + v_y) + \beta^2 I_{tt} = 0$

Put $I_x.u + I_y.v = \Phi$; and $u_x + v_y = \Gamma$; We have: $\Gamma.\Phi + \Gamma.I_t + \beta^2.I_{tt} = 0$ (*)

To approximate I by the time axis, we discrete the time domain, considering the time axis as created by the grid points with grid distance as Δ_t. Then there are the approximate formulas for first and second derivative derivatives as follows:

$$I_t = \frac{I_{i,j,t+1} - I_{i,j,t-1}}{2.\Delta_t} \qquad (21)$$

$$I_{tt} = \frac{I_{i,j,t+1} + I_{i,j,t-1} - 2.I_{i,j,t}}{\Delta_t^2} \qquad (22)$$

Replace (*) and find stability solution I:

$$\frac{dI_{i,j,t}}{dt} = \Gamma.\Phi + (\Gamma/2.\Delta_t + \beta^2/\Delta_t^2).I_{i,j,t+1} + (\beta^2/\Delta_t^2 - \Gamma/2.\Delta_t).I_{i,j,t-1} - 2.\frac{\beta^2}{\Delta_t^2}.I_{i,j,t} \qquad (23)$$

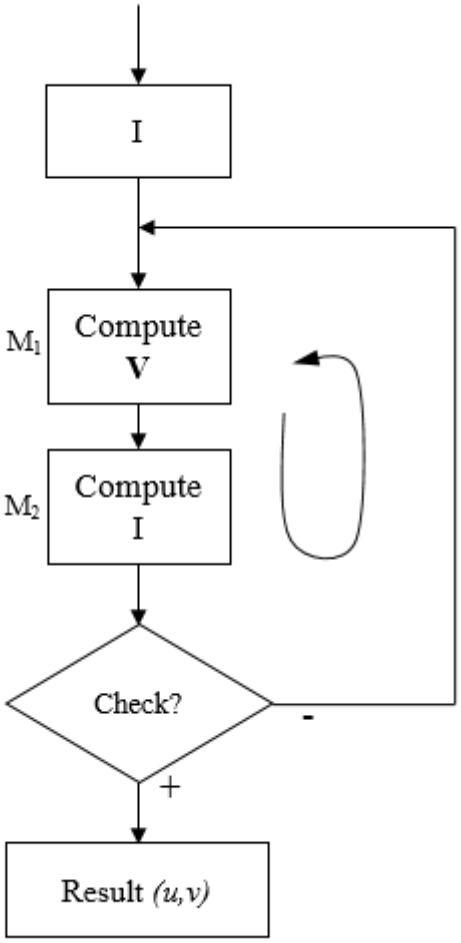


Figure 7. The algorithm solving the equations LDOF

Put $\Delta_t = 1$. The differential equation (23) is solved by CNN with the model and the template defined asfollows:

$B = 0$; $Z = \Gamma.\Phi$

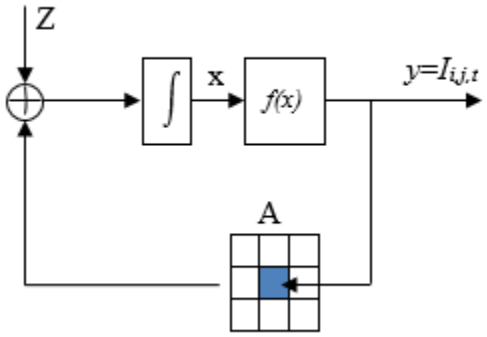


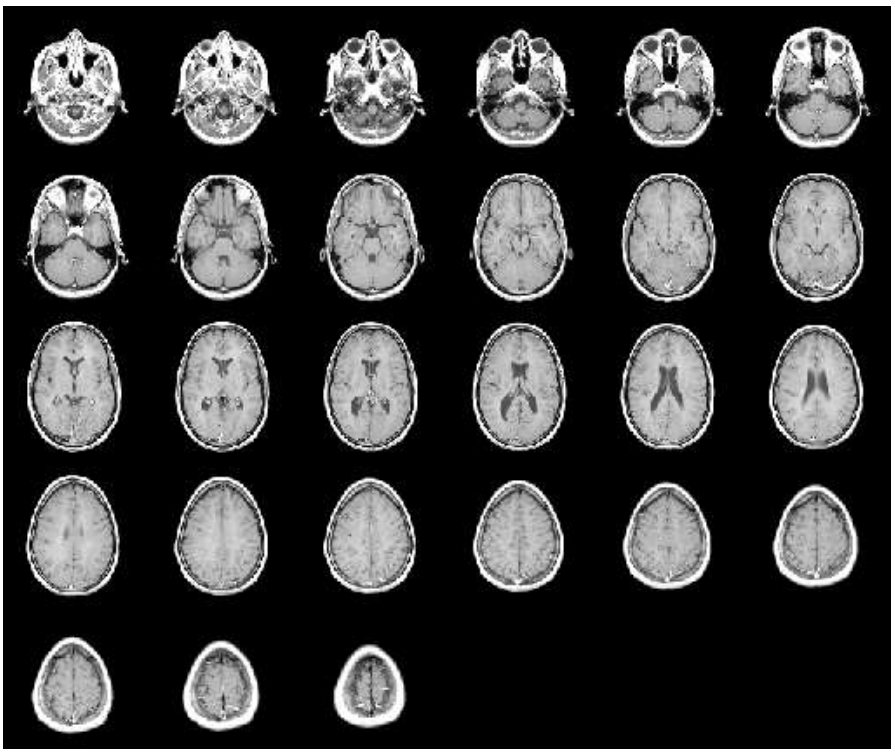
Figure 8. CNN for solving the equations LDOF

$$A = \begin{bmatrix} 0 & 0 & 0 \\ \beta^2 + \frac{\Gamma}{2} & -2.\beta^2 & \beta^2 - \frac{\Gamma}{2} \\ 0 & 0 & 0 \end{bmatrix}$$

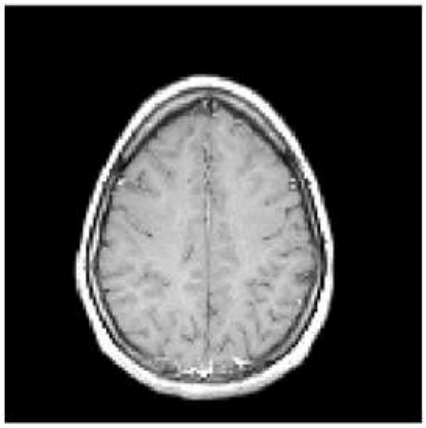
III. RESULT AND DISCUSSION

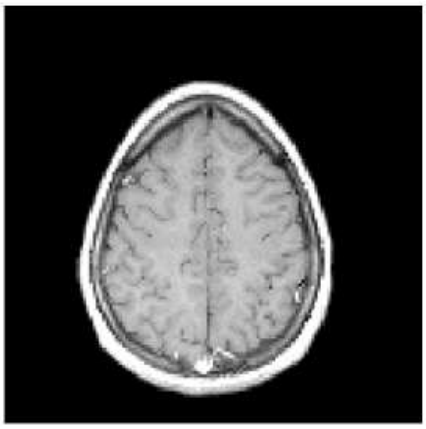
3.1 Experiments

Test running with motion brain image data based on biological rhythm. The MRI image obtained a sequence of 27 frames.

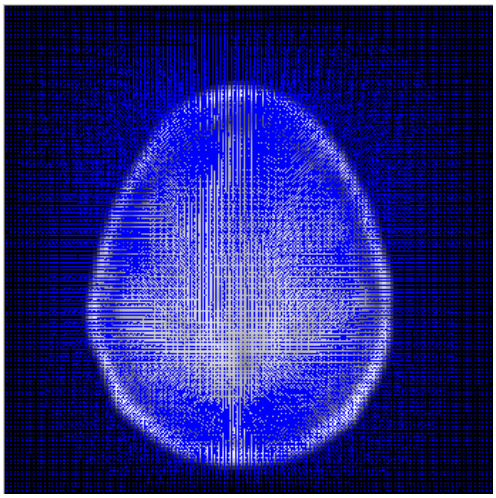


It is possible to choose randomly 2 consecutive 2 image for optical flow experiments.CNN simulation is on Matlab code. Experiment for 21st and 22nd frame:

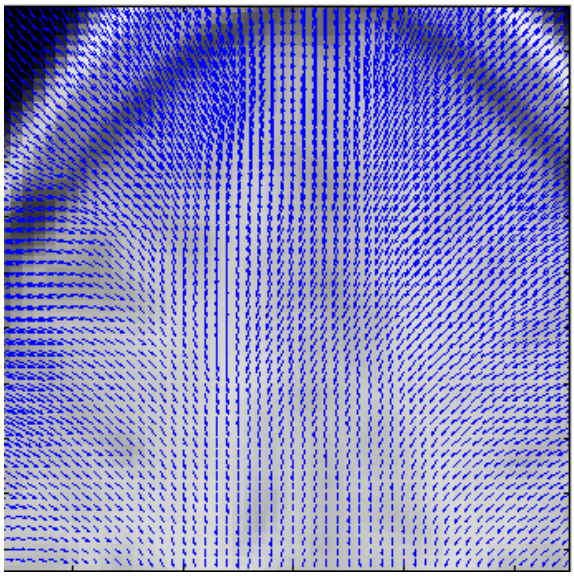




Calculating optical flow, the resulting velocity vectors are displayed as follows:



Zoom in:



In general, medical animation processing is particularly interested in applying the mathematical model of Optical flow to detect and analyze the operation of tissues, tumors, deformities,etc… to help find out processing of recommending an appropriate treatment regimen.

3.2 Conclusion and Open Problems

Motion image processing in healthcare has drawn attention of researchers to satisfy high demand in the field of image diagnosis and treatment (X- ray, PET, MRI, CT scanning…) Chest scanned image ischaracterized by soft object, rapid movement, complex disformation, discrepancy bu artificial influence… so the analysis and processing require high accuray and high speed. Optical flow based on gradient calculation hence is suitable for its flexibility and accuracy.

CNN network with above mentioned templates can be used to solve optical flow speed calculating for medical images (especially chest scanned image). Compared with previous methodes (sequentialprocessing, repeating algorithm…) this approach has many advantages: calculation speed of CNN network does not depened on size of imge but only on transition phase of electric cuircuit so it is very fast. Here we make a summation of research orientations and appoaches to optical flow in healthcare. We also deal with some problems and solutions (motion imge and large displacement, large data processing, fast speed…). Special attention is paid to application of CNN technology, a new but very potential tendency in motion image processing.

Nevertheless there are some problems to be addressed such as: the stability of CNN network, the requirement of sophisticated simulations, compressed image processing, large displacement in 3- dimension spce, processing and installation in 4- dimension space (3 spatial dimensions and 1 time dimension)

REFERENCES

[1]. L.O Chua, L. Yang , “Cellular Neural Network : Theory and Application”, IEEE Trans. on Circuits and System, Vol 35, pp 1257-1290, 1988.
[2]. T.Roska, LO. Chua, “The CNN Universal Machine: An Analogic Array Computer”, IEEE Trans. On Circuit and System, Vol 40, pp.163-173, 1993.
[3]. Leon O. Chu and T. Roska, "Cellular neural networks and visual computing: foundation and applications", Cambridge University Press, 2002 USA.
[4]. S.Beauchemin and J. Barron, "The computation of optical flow" , Surveys.September 1995 27(3); pp : 433-437.
[5]. Angela Slavova, "Cellular Neural Networks: Dynamics and Modelling", Kluwer Academic Publishers 2005
[6]. F.Meyer, "Iterative Image Transformation for an Automatic Screening of Cervical Cancer" , Journal of Histochemistry and Cytochemistry - 2000
[7]. Yan Ha, "Method of edge detection based on Non-Linear Cellular Automata", Proceedings of the 7th World Congress on Intelligent Control and Automation. 27/7/2008- Chongquing.China.
[8]. T.Corpetti, E.Memin, A.Santa-Cruz, D.Heitz & G.Arroyo, Electronic Imaging 1, 1998 pp: 347-352
[9]. B.Horn & Schunck , "Determining optical flow ", Artificial Intelligence, 17 . 08-1981; pp: 185- 203
[10]. R. Wildes, M.Amabile, Ann-Marie Lanzillotto & Tzong-Shyng Leu. "Recovering Estimates of Fluid Flow from Image Sequence Data" , Academic Press 08.2000.
[11]. Lucas B & Kanade T 1981. Procceding of DARPA IU Workshop, pp 121-130
[12]. E. Castillo, Yin Zang, T.Guerrero, "Compressible image registration for thoracic computed tomography image" , R 2005 Physics in medicine and Biology 51 pp 1-15
[13]. A. Bruhn & W.Joachim , "Combining Local and Global Optical flow methods" , International Journal of computer Vision 61(3), pp 211-231,2005 Springer Science+Business Media Inc. Manufacture in The Netherlands 4.2004.
[14]. J Barron and N.Thacker, "Tutorial:Computing 2D and 3D optical flow" , Tina Memo No.2004-012, Internal document of Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 2004.
[15]. R. Adrian, "Twenty year of particle image velocimetry", Experiments in Fluid, 2005, 39(2) pp:159-169.
[16]. S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In 32nd International Conference on Machine Learning, pages 448–456, Feb. 2015.
[17]. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag., 29(6):82–97, 2012.
[18]. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
[19]. E. Haber, L. Ruthotto, and E. Holtham. Learning across scales - A multiscale method for convolution neural networks. In AAAI Conference on AI, volume abs/1703.02009, pages 1–8, 2017.
[20]. E. Haber and L. Ruthotto. Stable architectures for deep neural networks. Inverse Probl., 34:014004, 2017.
[21]. W. E. A Proposal on Machine Learning via Dynamical Systems. Comm. Math. Statist., 5(1):1–11, 2017.
[22]. A. Dundar, J. Jin, and E. Culurciello. Convolutional Clustering for Unsupervised Learning. In ICLR, Nov. 2015.
[23]. Y. Chen and T. Pock. Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. IEEE Trans. Pattern Anal. Mach. Intell., 39(6):1256–1272, 2017.
[24]. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. arXiv preprint arXiv:1806.07366, 2018.
[25]. P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and G. Carlier. Deep Relaxation: Partial Differential Equations for Optimizing Deep Neural Networks. arXiv preprint 1704.04932, Apr. 2017.

Thanh-Tung Cap" Cellular Neural Network and Application in Animation Proccessings" International Journal of Research in Engineering and Science (IJRES), vol. 07, no. 3, 2019, pp. 51-61