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

Implementation Aspects of a Topographic Cellular Active Contour Algorithm

Dániel Hillier[†] [‡], Zsolt Czeilinger[‡] [⋆] and Csaba Rekeczky[§]

†Friedrich Miescher Institute for Biomedical Research, Basel, Switzerland
‡Faculty of Information Technology, Péter Pázmány Catholic University, Budapest, Hungary
⋆ György Gottsegen Institut for Cardiology, Haller u. 29., 1096 Budapest, Hungary
§Eutecus Inc., 1936 University Ave., Suite 360, Berkeley, USA
Email: daniel.hillier@fmi.ch, cuzi@deloitte79.axelero.net, rcsaba@digitus.itk.ppke.hu

Abstract—Topographic Cellular Active Contour (TCAC) algorithms are designed for solving difficult segmentation tasks at high speed. Implementation of a TCAC method on both parallel analogue and digital computing architectures is discussed in this paper. In the context of 3D echocardiography the performance of the TCAC algorithm is compared to other methods.

1. Introduction

Algorithms designed for machine vision applications such as medical imaging, surveillance, etc. in most cases need to segment the raw image flow and/or track a region in it. To solve this task, many algorithms were proposed stemming from the original Active Contour (AC) models [1] and others based on partial differential equations (PDEs) [2]. However the high computational cost associated to these methods can hamper their application to solve problems where fast processing is required.

Using an appropriate discretization approach these models can be mapped to a cellular structure described by nonlinear ordinary differential equations (ODEs), i.e. to a Cellular Nonlinear Network (CNN) [3] architecture. Topographic Cellular Active Contour (TCAC) [6] methods can be considered as a space and/or time discretized implementation of the original active contour or PDE models [4]. The Pixel Level Snakes (PLS) [5] is a TCAC technique where the contours are explicitly represented and evolve using discrete iteration steps to deform the contour towards local minimal distance curves based on a metric defined as a function of the features to be segmented. The Cellular Wave Computing [6] method (CWC) is based on the non-iterative implicit region propagation operator where the contours are defined by the boundaries of trigger waves.

This paper presents implementation details of the high speed CWC method in the context of 3D echocardiography where the task was to extract the endocardial boundary of the right atrium from 3D ultrasound data. The paper is organized as follows. The continuous time (non-iterative) and discrete time (iterative) implementation of the CWC method used for endocardial boundary extraction is presented in Section 2. Section 3 describes experiments and a comparison assessing the computational performance of the CWC method.

2. Constrained Wave Computing

Low-level image processing operators like filtering, edge detection, binary hole filling, feature extraction, etc. are computationally intensive. These operations are inherently pixel-parallel, i.e. identical, localized operations are performed on every pixel. Efficient image processing systems can be designed by associating each image pixel with an image processing circuitry and allowing local connections between neighboring processing cells. Each cell can have local memories and can perform basic arithmetic and logic operations on pixel values of their local neighborhood. CNNs [7] represent a powerful framework for this concept. In many CNN implementations, each individual cell circuitry is a generic realisation of Eq. (2), i.e. CNNs can be used to approximate solutions of PDEs. A number of different CNN processor implementations are available for parallel image processing [8] [9] on which various difficult image processing problems were solved at high speed [10].

Implicit models represent an important class of image segmentation approaches. Motivated by [11], [2], [12] a propagating wavefront can be defined by a geometric flow. The propagation velocity is made up of two terms, the regularity of the boundary and image derived information. The model of CWC is given by a reaction-diffusion type PDE:

$$\frac{\partial I_P(x,y,t)}{\partial t} = div\ grad(I_P(x,y,t)) + \\ + \mathcal{F}_1(I_{Control}(x,y,t_0)) + \mathcal{F}_2(I_P(x,y,t)) \quad (1)$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are nonlinear functions, $I(x,y,t) : [0,N]^2 \times [0,T] \rightarrow [0,M]$. $I_P(x,y,t)$ denotes an intensity image in which a black region is evolving from its initial state controlled by external image forces represented in $I_{Control}(x,y,t_0)$.

The solution of the PDE can be approximated using spatial discretization made in equidistant steps in both directions, $\Delta x = \Delta y = h = 1$. This way, $I_P(x,y,t)$ is mapped onto a CNN[13, 3] array such that the state value $I_X(i,j,t)$ of a CNN cell at a grid point i, j is associated with $I_P(ih, jh, t)$. Using Taylor-series expansion of $I_P(x,y,t)$ the CNN operator (template) corresponding to the second spatial derivative can be obtained [14]. The image derived

spatial constraint term $\mathcal{F}_1(.)$ is replaced by the linear combination of the values stored in $I_{Control}(i, j, t_0)$ and $\mathcal{F}_2(.)$ is replaced by the nonlinear function g that converts the diffusion into a region propagation process. The ordinary differential equation approximating the solution of Eq. (1) at the grid point i, j can be formulated as:

$$\frac{dI_X(i,j,t)}{dt} = -I_X(i,j,t) + \frac{c_1}{4}[I_P(i-1,j,t)+ + I_P(i+1,j,t) + I_P(i,j-1,t) + I_P(i,j+1,t)]+ + z(i,j) + g[I_X(i,j,t)] \quad (2)$$

where $I_P(i, j, t) = f[I_X(i, j, t)]$ and $g[.] = c_0 \cdot f[.]$; the nonlinear, sigmoid-type function f is defined as $f[I_X(i, j, t)] = 0.5 \cdot (|I_X(i, j, t) + 1| - |I_X(i, j, t) - 1|)$. Velocity of the propagating front is depending on the actual content of the image formulated in the term $z(i, j) = z_{const} + \sum_{k,l \in S_1(i,j)} b_{(k,l)} I_{Control}(k, l, t_0)$ where $S_1(i, j)$ represents the 3×3 neighborhood of the cell i, j, contributions of the neighbors are weighted by the values of $b_{(k,l)}$.

Setting constants in Eq. (2) to $c_0 = 3$ and $c_1 = 1$ and $b_{0,0} = -2$ and $z_{const} = 3.75$ an expanding binary wave process is generated (see [12] for an in-depth analysis). In the specific application field of endocardial boundary extraction from echocardiographic recordings, $I_{Control}$ is associated with the raw ultrasound data taken from the input data set at a specific time instant and the value of I_P at $t = 0$ is a small black patch inside the cavity. The non-iterative constrained wave process set up this way can extract the endocardium from the raw ultrasound data in a single instruction on a continuous time CNN processor[15, 16].

Using explicit forward Euler discretization Eq. (2) can also be solved in the discrete time domain. In that case Eq. (2) becomes:

$$I_X(i,j,n+1) = (1-\Delta t)\cdot I_X(i,j,n) + \Delta t \frac{c_1}{4}[I_X(i-1,j,n)+ + I_X(i+1,j,n) + I_X(i,j-1,n)+ + I_X(i,j+1,n)] + z(i,j) + \Delta t \cdot g[I_X(i,j,n)] \quad (3)$$

where $z(i, j) = z_{const} + \Delta t \sum_{k,l \in S_1(i,j)} b_{(k,l)} I_{Control}(k, l, n_0)$, $I_P(i, j, n + 1) = f[I_X(i, j, n + 1)]$; $g[.]$ and $f[.]$ remain as defined in Eq. 2. Iterating Eq. (3) in time with sufficiently small Δt approximates the solution of the continuous time model. The discrete time model can be solved on a much wider range of processors but it is obviously much slower than the continuous time model implemented on the ACEx processors [16].

The discrete-time CNN (DTCNN) model [17] is obtained from the standard CNN equations when $\Delta t = 1$ in the forward Euler discretization scheme. The DTCNN implementation of an algorithm might not converge to the original solution. DTCNNs or other similar pixel-parallel "single instruction multiple data" (SIMD) architectures [18, 19, 20] are particularly suitable to execute morphology operations on images in a pixel-parallel fashion.

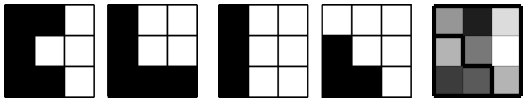

Figure 1: Local neighborhood operators used in the discrete time model of the CWC method. On all pixels of the eroded version of $I_P(i, j, n)$, "must be black" masks (rotated masks with 90 180 and 270 are not shown) set pixels of the patch image $I_P(i, j, n + 1)$ black that have a neighborhood of more than four black pixels (1^{st} and 2^{nd} mask images). "Valid black" masks (rotated versions not shown) select pixels having exactly three neighboring black pixels that are adjacent to each other (3^{rd} and 4^{th} images). The selected pixels correspond to the edge of the patch that is deformed under the guidance of control constraints to fit image features in $I_{Control}(k, l, n_0)$. Pixels in $I_P(i, j, n + 1)$ selected neither by the "must be black" nor by "valid black" masks are turned to white eliminating singular black pixels disconnected from the main patch. For the remaining edge pixels of the patch image, the average grayscale level of the neighboring pixels on the control image corresponding to the black and white pixels in the patch image are calculated (rightmost image). If the difference between the "white" and "black" average values is below a local difference threshold, the pixel is set to white, otherwise to black.

The solution of the discrete time CWC model formulated in Eq. (3) can be approximated on such architectures as follows.

For every pixel in $I_P(i, j, n)$, spatial constraints on the local, nearest neighborhood of every pixel are checked in a properly defined sequence to find the new value of $I_P(i, j, n + 1)$. The internal constraints are represented by hit-and-miss masks. Hit-and-miss is a general binary morphological operation that can be used to look for particular masks (patterns) of foreground and background pixels in an image. Internal constraints ensure that the patch will not contain singular white pixels, and fill up the deeper concavities along the boundary of the patch. "Must be black" mask checking (see Fig. 1) sets pixels of the patch image black that have a neighborhood of more than four black pixels.

"Valid black" masks select pixels having exactly three neighboring black pixels that are adjacent to each other. The selected pixels correspond to the edge of the patch that is deformed under the guidance of control constraints to fit image features in $I_{Control}(k, l, n_0)$. Pixels in $I_P(i, j, n + 1)$ selected neither by the "must be black" nor by "valid black" masks are turned to white in eliminating singular black pixels disconnected from the main patch. Grayscale spatial constraints are applied only to the result of internal constraint checking, i.e. to the current edge pixels of $I_P(i, j, n)$. The algorithm calculates the average grayscale level of the neighboring pixels on the control image corresponding to the black and white pixels in the patch image, respectively.

If the difference between these "white" and "black" average values is below a local difference threshold, it sets the pixel to white, otherwise to black.

The grayscale average comparison is the essential contour extraction step. By computing the intensity difference between the inner and outer side of the region boundary, the algorithm approximates the component of the local gradient orthogonal to the boundary of the patch. The patch is expanding if the intensity gradient is greater than a local difference threshold, and shrinking when it is less. In the specific application of echocardiography, where the cardiac wall to be segmented is represented by locally bright regions in $I_{Control}(k, l, n_0)$, the patch approaching from the center of the cavity "sees" a gradual increase in the average intensity until the intensity plateau of the cardiac wall is reached. This change of gradient amplitude is detected by this method.

3. Experiments and Results

The continuous time version of CWC was implemented on the 128×128 pixels resolution, massively parallel processor (ACE16k [16]). The discrete time version of CWC was implemented in C language and ran on a PC (Pentium 4 3GHz with 1Gb RAM) to serve as a reference for performance evaluation.

The CWC method implemented on the massively parallel processor achieved 500 frames/s (fps) performance outperforming the Pentium 4 implementation that reached 40 fps. The performance advantage of the continuous time implementation of CWC is caused by application of the so-called constrained trigger wave operator [12] that is a dynamic operator solving the contour detection problem in a single instruction.

Echocardiography is a specific application field used to evaluate the computational performance of the CWC method against other methods (see Table 1). Most methods were designed to segment the LV and use a priori learned shape information. The time required for shape learning was not included in the figures of Table 1. Also, the reader should keep in mind, that the proposed method was applied to extract the right atrium whereas other methods were applied to the rather different, geometrically much simpler problem of extracting the left atrium.

The integer performance measurements executed in a standardized test environment (SPEC®, SPECint2000, www.spec.org) were used to bring processor performances to a common ground. Content of this comparison table should be taken with great caution and it is not intended as a rigorous performance analysis. 2D and 3D methods were brought to a common ground via calculating the number of processed frames per second (fps). For 3D methods, if the published performance result was in volumes per second, the volume per second value was multiplied by the lowest spatial resolution to get fps.

We are aware that even after this transformation, fps gives slightly distorted information about the performance of the original algorithm. Therefore we did not go further in transforming results to a common measure, because further spatial scale changes would not have taken into account the unknown "slow down" ratio specific to each different algorithmic approach when processing data sets with different resolutions.

Acknowledgments

The authors acknowledge the support of the whole team of Analogic Ltd. for providing support to the Bi-i system. We express our thanks to V. Binzberger and D. L. Vilarino for their contributions, and to A. Szatmári and to T. Roska for their general support.

References

[1] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (4) (1988) 321 – 331.

[2] R. Malladi, J. Sethian, B. Vemuri, Shape modeling with front propagation: a level set approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (2) (1995) 158–175.

[3] L. O. Chua, T. Roska, The CNN paradigm, IEEE Trans. on Circuits and Systems 40 (1993) 147–156.

[4] C. Rekeczky, T. Roska, Calculating local and global PDEs by analogic diffusion and wave algorithms, Proceedings of the European Conference on Circuit Theory and Design 2 (2001) 17–20.

[5] D. L. Vilarino, C. Rekeczky, Pixel-level snakes on the CNN-UM: Algorithm design, on-chip implementation and applications, International Journal of Circuit Theory and its Applications 33 (2005) 17–51.

[6] D. Hillier, V. Binzberger, D. L. Vilarino, C. Rekeczky, Topographic cellular active contour techniques: Theory, implementations and comparisons, International Journal of Circuit Theory and Applications 34 (2) (2006) 183–216.

[7] L. Chua, T. Roska, Cellular neural networks and visual computing, Cambridge University Press New York, NY, 2002.

[8] A. Zarandy, M. Foldesy, P. Szolgay, S. Tokes, C. Rekeczky, T. Roska, Various implementations of topographic, sensory, cellular wave computers, IEEE International Symposium on Circuits and Systems (ISCAS) (2005) 5802–5805.

[9] P. Dudek, P. Hicks, A general-purpose processor-per-pixel analog SIMD vision chip, Circuits and Systems

First author and reference	Keyword description of the underlying algorithm	Microprocessor	CINT 2000	Spatial resolution of input (x*y(*z))	Fps
Bosch [21]	Active appearance models	Pentium III 800 MHz	365	768*576(*16)	2,67
Wolf [22]	Multi-scale edge detection	Pentium III 1 GHz	423	768*576	1
Proposed method	topographic, discrete time	Pentium 4 3 GHz	1149	128*128(*90)	40
Proposed method	topographic, continuous time	Parallel (ACE16k)	Analog proc	128*128(*90)	500

Table 1: Computing performance comparison. Common ground for comparison is the number of processed frames per second (fps), derived from published data. Processing power of processors on which the given method was implemented were brought to a common ground using the CINT2000 figures from www.spec.org. Spatial resolution is given for reference, third number in parenthesis denotes resolution in the third spatial dimension. The reader should take this comparison with great caution and is referred to the text for further comments.

I: Regular Papers, IEEE Transactions on [see also Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on] 52 (1) (2005) 13–20.

[10] C. Rekeczky, I. Szatmari, D. Balya, G. Timar, A. Zarandy, Cellular multiadaptive analogic architecture: a computational framework for UAV applications, Circuits and Systems I: Regular Papers, IEEE Transactions on [see also Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on] 51 (5) (2004) 864–884.

[11] V. Caselles, R. Kimmel, G. Sapiro, Geodesic Active Contours, International Journal of Computer Vision 22 (1) (1997) 61–79.

[12] C. Rekeczky, L. O. Chua, Computing with Front Propagation: Active Contour And Skeleton Models In Continuous-Time CNN, The Journal of VLSI Signal Processing 23 (2) (1999) 373 – 402.

[13] L. O. Chua, L. Yang, Cellular neural networks: Theory, IEEE Trans. on Circuits and Systems 35 (1988) 1257–1272.

[14] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, F. Puffer, Simulating nonlinear waves and partial differential equations via CNN - part I: Basic techniques, IEEE Trans. on Circuits and Systems - I. Fundamental Theory and Applications 42 (10) (1995) 807–815.

[15] G. Liñán, S. Espejo, R. Domínguez-Castro, A. Rodríguez-Vázquez, Architectural and Basic Circuit Considerations for a Flexible 128× 128 Mixed-Signal SIMD Vision Chip, Analog Integrated Circuits and Signal Processing 33 (2) (2002) 179–190.

[16] A. Rodriguez-Vazquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, S. E. Meana, ACE16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs, Circuits and Systems I: Regular Papers, IEEE Transactions on [see also Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on] 51 (5) (2004) 851 – 863.

[17] H. Harrer, J. A. Nossek, Discrete-time cellular neural networks, International Journal of Circuit Theory and Applications 20 (1992) 453–468.

[18] P. Dudek, S. J. Carey, A general-purpose 128x128 simd processor array with integrated image sensor, Electronics Letters 41 (12) (2006) 678–679.

[19] A. Lopich, P. Dudek, Global operations in SIMD cellular processor arrays employing functional asynchronism, Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006. International Workshop on (2006) 18–23.

[20] P. Foldesy, A. Zarandy, C. Rekeczky, T. Roska, High performance processor array for image processing, Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on (2007) 1177–1180.

[21] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, J. H. C. Reiber, Automatic segmentation of echocardiographic sequences by active appearance motion models, Medical Imaging, IEEE Transactions on 21 (11) (2002) 1374 – 1383.

[22] I. Wolf, M. Hastenteufel, R. D. Simone, M. Vetter, G. Glombitza, S. Mottl-Link, C. F. Vahl, H. P. Meinzer, ROPES: a semiautomated segmentation method for accelerated analysis of three-dimensional echocardiographic data, Medical Imaging, IEEE Transactions on 21 (9) (2002) 1091 – 1104.