Image Processing Applications on ACE16k Using Iterative Annealing Method

S. Sevgen\(^1\) and M.E. Yüksel\(^2\)

\(^1\)Istanbul University, Istanbul/Turkey, sevgens@istanbul.edu.tr
\(^2\)Harran University, Sanliurfa/Turkey, erkan.yuksel@harran.edu.tr

Abstract—Cellular neural networks proved to be a useful parallel computing system for image processing applications. Cellular neural networks (CNNs) constitute a class of recurrent and locally coupled arrays of identical cells. The connectivity among the cells is determined by a set of parameters called templates. CNN templates are the key parameters to perform a desired task. One of the challenging problems in designing templates is to find the optimal template that functions appropriately for the solution of the intended problem. In this paper, we implement the IAOM on the analog CNN chip to find an optimum template by training a randomly selected initial template. We have been able to show that the proposed system is efficient to find the suitable template for some specific image processing applications.

Keywords—Cellular Neural Networks, Iterative Annealing, ACE16k, Template Training.

I. INTRODUCTION

The key feature of a Cellular Neural Networks (CNN), introduced in [1][2], is that it is a locally interconnected analog processor array. Since CNN has two dimensional (2D) grid structure, it is a suitable platform for developing image processing algorithms. Based on the mathematical modeling of CNNs, a programmable CNN, called CNN universal machine (CNN-UM) [3] has been developed. The CNN-UM is programmable array computer with real time and super computer power in a single chip. Since these chips have huge computational power and capability of parallel processing, it is possible to perform image processing tasks in a high speed in comparison to conventional architectures. The one of the hardware implementations of the CNN is Analog Focal Plane processor called ACE16k [4].

Program instructions called templates have most important role in the CNN applications. The dynamical behavior of a CNN is completely determined by the templates. The design of suitable templates is one of the fundamental tasks in CNN area. It is also important to find optimal values for template elements so that a CNN performs a desired task.

Template coefficients can be obtained by design or learning techniques. In the design techniques, the operation goal is translated into a set of local dynamic rules. Input and corresponding output are the fundamental elements of the learning techniques. In the learning techniques, the CNN templates are determined by trying to translate a global dynamical phenomenon into local relations among cells, and subsequently into a set of local rules, represented through the template parameters.

Fundamental learning/design methods can be classified as follows [5]:

- Decomposition techniques using coupled/uncoupled CNN and systematic methods for binary input/output functions
- Global optimization methods
- Genetic algorithms
- Fuzzy logic techniques
- Neural Networks techniques

Kozek T. and et al. used Genetic Algorithm for template learning [6]. Bahram M. and et al. developed a learning algorithm based on Back-Propagation [7]. Loncar A. and et al. developed a simulator system called SCNN which uses wide range of training algorithms [8]. All these methods are simulated to validate the accuracy of the trained templates using a CNN simulator or calculating the dynamics of the cells.

Since, VLSI (Very Large Scale Integration) implementation of ACE16k has some restrictions; the obtained templates by the simulation systems cannot give the accurate results on the chip. In order to overcome this problem, we have designed a template training system on ACE16k chip to obtain more stable templates. Iterative Annealing optimization method, a kind of Simulated Annealing, is implemented for the training system. The main advantage of using ACE16k chip is that the processing speed is much higher than the speed of the simulation systems. In addition, the hardware parameters are optimized to compensate the chip inaccuracies.

This paper is organized as follows. In section 2, the CNN architecture, ACE16k chip and Bi-i Cellular Vision System are examined. Section 3 deals with the optimization method using training procedure. In section 4, the training of templates on the CNN chip by implementing the Iterative Annealing method is explained. In section 5, the obtained templates and their application to different images are given. Finally, the concluding remarks are given in Section 6.

II. CNN ARCHITECTURE AND BI-I CELLULAR VISION SYSTEM

A. CNN Structure

Standard CNN consists of \(M \times N\) rectangular array of cells. The smallest part of CNN is called cell \((C(i,j))\) with Cartesian
coordinates \((i,j)\) \((i=1,2,3...M,\ j=1,2,3...N)\). Each cell can be defined by the following linear and non-linear mathematical equations \([9]\):

\[
\frac{dx_{ij}}{dt} = -x_{ij} + \sum_{C(k,j)=S,(i,j)} A(i, j; k, l)x_{klj} + \sum_{C(k,j) \in S,(i,j)} B(i, j; k, l)u_{klj} + z_{ij}
\]

\[
y_{ij} = f(x_{ij}) = \frac{1}{2}|x_{ij} + 1| - \frac{1}{2}|x_{ij} - 1|
\]

where,

\(x_{ij} \in \mathbb{R}\); State variable of cell \(C(i,j)\),
\(y_{klj} \in \mathbb{R}\); Outputs of cells,
\(u_{klj} \in \mathbb{R}\); Inputs of cells,
\(z_{ij} \in \mathbb{R}\); Threshold,
\(A(i, j; k, l)\); Feedback operator,
\(B(i, j; k, l)\); Control operator.
\(y_{ij}\); Output equation.

The sphere of influence, \(S_r(i, j)\), of the radius \(r\) of cell \(C(i,j)\) is defined to be set of all neighborhood cells satisfying the property \([9]\)

\[
S_r(i, j) = \left\{ (k, l) \mid \max_{1 \leq k, l \leq N} \| k-i \| \| l-j \| \leq r \right\}
\]

The total number of the template parameters in a CNN is 19 when \(r=1\) (a threshold parameter \(z_{ij}\), 9 parameters \(a_{klj}\), 9 parameters \(b_{klj}\)). The general structure of the CNN templates is as follows

\[
A = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix}, \quad B = \begin{bmatrix}
b_{11} & b_{12} & b_{13} \\
b_{21} & b_{22} & b_{23} \\
b_{31} & b_{32} & b_{33}
\end{bmatrix}\quad \text{and } Z = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}
\]

**B. ACE16k Chip**

ACE16k is a CNN-UM implementation. CNN-UM is an analog and logic computer that consists of many interconnected parallel processor units on its main processor. ACE16K can be basically described as an array of 128x128 identical, locally interacting, analog processing units designed for high speed image processing tasks. The system contains a set of on-chip peripheral circuitries that, on one hand, allow a completely digital interface with the host, and on the other provide high algorithmic capability by means of conventional programming memories where the algorithms are stored \([4]\).

Although ACE16K is essentially an analog processor (computation is carried out in the analog domain), it can be operated in a fully digital environment. For this purpose, the prototype incorporates a bank of Digital-to-Analog (for input) and Analog-to-Digital (for output) converters at the images I/O port \([4]\).

ACE16K is conceived to be used in two alternative ways. First, in applications where the images to be processed are directly acquired by the optical input module of the chip, and second, as a conventional image co-processor working in parallel with a digital hosting system that provides and receives the images in electrical form \([4]\).

**C. Bi-i Cellular Vision System**

The Bi-i cellular vision system which contains ACE16k chip and Digital Signal Processor (DSP) is a high-speed, compact and intelligent camera for training (Figure 1). Most important interface of Bi-i is 100 Mbit Ethernet. Programs to be run on Bi-i are loaded over Ethernet, and the host computer can write or read data to or from the Bi-i over Ethernet. Instant Vision Libraries and Bi-i SDK (Software Development Kit) are set of \(C++\) programming library for developing Bi-i applications. These libraries can be used with the development environment for the DSP and ACE16k called Code Composer Studio. Functions in the SDK are operations on different components of the Bi-i hardware such as operating the CMOS sensor. TACE_IPL library is an image processing library for ACE16k chip \([10]\) \([13]\).

![Figure 1: Bi-i Cellular Vision System](image)

**III. ITERATIVE ANNEALING**

Iterative Annealing (IA), a kind of Simulated Annealing, is an optimization method specially developed for CNN \([11]\). The algorithm of Iterative Annealing is shown below:

1. Choose initial values \(x_0, s_{max}, j_{max}, T_0, \tau, j = 0\)
2. Calculate step size \(\nu = (\tau / T_0)^{j_{max} - j_{max}}\)
3. \(T = T_0, i = 0\)
4. \(y_i^k = x_i^k + u^k \cdot T; u^k : \text{Unit distribution U[-0.5, 0.5]}\)
5. If \(f(y_i^*) < f(x_i^*)\) then \(x_{i+1} = y_i^*\)
6. Reduce temperature \(T = \nu \cdot T\)
7. \(i = i + 1\)
8. If \(i < (s_{max} / j_{max})\) then Go To 4

81
9. \( j = j + 1 \)
10. If \( (j < j_{\text{max}}) \) Then Go To 3

The function \( f(\tilde{x}) \) represents the error measure which must be minimized and \( \tilde{x} \) represents the D-dimensional parameter vector. \( s_{\text{max}} \) is the maximum number of iteration steps and \( j_{\text{max}} \) determines how many reruns, have to be carried out. Having the physical effect of Annealing in mind, we call \( T_0 \) the start temperature. The minimal temperature \( \tau \) determines the accuracy of the parameter vector at the global minimum. At every step the temperature is chilling, leading to a decreasing search area until \( T \) reaches \( \tau \). Then the process restarts with \( T=T_0 \). Finally a global minimum is found [13].

IV. ON CHIP TRAINING WITH ITERATIVE ANNEALING

Iterative Annealing (IA) method was modified to work on a PC with the ACE16k chip. This means that we can obtain templates which are stable and robust without inaccuracies of CNN-UM hardware realization. ACE16k chip as an external process unit obtains output images for variable template configurations during training process. IA algorithm consists of two loops. The inner loop contains annealing procedure. The outer loop controls iterative behavior. The function to minimize is an error measure calculated between a given reference image \( R \) and an output image \( O \) obtained from the chip [12]. This function is

\[
 f(R, O) = \frac{\sum_{i=1}^{a} \sum_{j=1}^{b} |r_{i,j} - o_{i,j}|}{g \times a \times b}
\]  \hspace{1cm} (4)

Here \( a \) and \( b \) are the image dimensions, \( r_{i,j} \) and \( o_{i,j} \) the pixel grey values. \( g \) is a factor for normalization to obtain values between 0 and 1.

We can modify the Iterative Annealing algorithm to adapt to the chip by adding the following steps into the inner loop: 1. Templates are generated dynamically out of the parameter vector to load and perform them directly on the chip. 2. The chip output image saved for computing the distance to a reference image. This algorithm generates templates using parameter sets. Then, it loads and runs these templates to ACE16k chip, saves output images and compares them with desired output using error measure function [12].

V. EXPERIMENTAL RESULTS

We have developed on chip training system by using Iterative Annealing method based on procedure in section IV. Starting the annealing at the temperature of \( T_0 = 10 \) and cooling off each step by a factor of \( \nu = 0.91 \) for 100 steps are the initialize parameters in all optimization procedures. Initial template values are chosen randomly.

In order to test performance of the system, we have tried to learn gray level edge detection template. Resulting template set is given below:

\[
 A = \begin{bmatrix} 0 & 0 & 0 \\ 4.54 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -2.98 & -0.47 & -2.26 \\ 2.07 & 5.66 & 1.74 \end{bmatrix} \quad I = -0.96
\]  \hspace{1cm} (5)

We have applied the trained template to ACE16k chip to show the accuracy of the template. We have created two sample binary images (128x128 pixels) as shown in Figure 2a, Figure 3a and Figure 4a. Outputs of template are given in Figure 2b, Figure 3b and Figure 4b, respectively.

![Image 1](https://via.placeholder.com/150)

Figure 2: Test – 1
(a) Input image (b) Output image

![Image 2](https://via.placeholder.com/150)

Figure 3: Test – 2
(a) Input image (b) Output image

![Image 3](https://via.placeholder.com/150)

Figure 4: Test – 3
(a) Input image (b) Output image
VI. CONCLUSION

In this paper we have implemented a template training system using Iterative Annealing optimization technique on ACE16k CNN chip in Bi-i Cellular Vision System. We have obtained gray level edge detection template on the ACE16k chip and given test results. Output images that we had obtained have proved accuracy of the trained edge detection template and efficiency of developed template training system.

ACKNOWLEDGMENT

This work was supported by Scientific Research Projects Coordination Unit of Istanbul University. Project number YADOP – 5781.

REFERENCES