Congreso Nacional de Control Automático 2006
18 - 20, Octubre, 2006, UNAM, México, DF
ISBN 970-32-4025-9

Stable Training Cellular Neural Networks For Computer Vision

Marco A. Moreno-Armendariz[1], Wen Yu[2], Juan Humberto Sossa-Azuela[1]
[1] Centro de Investigación en Computación-IPN, AV. Juan de Dios Bátiz S/N,
Unidad Profesional Adolfo López Mateos, tel. +52-55-5729-6000 ext. 56525, fax. +52-55-5729-6000 ext. 56607,
México D.F., 07738, México, marco_moreno@cic.ipn.mx
[2] Control Automatico, CINVESTAV-IPN, Av. IPN 2508, México D.F., 07360, México

***Abstract*— For visual servoing or path visual planning robotic tasks is very important to use a computer vision system with enough capacity to analize the images in real time. One common approach is use a PC computer to adquire or/and process the images from the scene and then generate the control commands for the robot. Unfortunatly for work in real time a big amount of computer resources are needed. In other hand, is important that the visual algorithms have the capacity to be adaptable for the changing enviroment conditions. In this paper we propose a stable learning algorithm for CNN looking to provide with more flexibility this kind of systems. Also we present a FPGA implementation to add real time operation.**
Keywords: Neural networks, Stability analysis, Computer vision.

I. INTRODUCCIÓN

Cellular Neural Networks (Chua and Yang, 1988; Chua and Roska, 1993; Harrer and Nossek, 1992) are arrays of dynamic artificial neurons (cells) with local translation invariant connections. This essential characteristic allows for hardware implementation of large networks on a single VLSI chip (Liñan et al., 2002). Cells are typically organized in a fairly large two-dimensional grid, so that CNNs can be used as massively parallel image processing tools, by associating each pixel of an image to one cell. Moreover, the sequential action of such a network with different parameters is capable of performing arbitrarily complex operations on the input image. These unique features make CNNs a natural candidate for computer vision.

In order to determine the templates automatically, the learning CNNs are studied by some researches based on others techniques. In (Radwan and Tazaki, 2004) the authors propose to use the Genetic Algorithm for designing the cloning template. Another way is using an hybrid structure called Recurrent Fuzzy Cellular Neural Networks (RFCNN), where each CNN template in the consequent parts are adjusted by the ordered derivative algorithm to minimize a given cost function (Lin and Lee, 1996). This ideas required another structure to obtain the values for the cloning templates. In this paper we transform the CNNs into the form of recurrent neural networks, and the learning don 't use additional structure.

Input-to-state stability is another elegant approach to analyze stability besides Lyapunov method. It can lead to general conclusions on stability by using input/state characteristics. By input-to-state stability (ISS) theory, we already prove in (Yu and Li, 2003) that the normal gradient law and backpropagation-like algorithm without robust modifications are L_∞ stable for discrete-time *feedforward* neuro identification. Neuro identification is in sense of black-box approximation. All uncertainties can be considered as parts of the black-box, *i.e.*, unmodeled dynamics are within the black-box model, not as structured uncertainties. Therefore the common used robustifying techniques are not necessary. By passivity theory, gradient descent algorithms for continuous-time recurrent neural networks were stable and robust to any bounded uncertainties without robust modification (Yu and Li, 2001). In this paper, ISS approach is applied to obtain the new off-line learning laws that do not need robust modifications.

In order to improve the processing time of the images we used a low level language to emulate in a FPGA a Discrete-time CNN. A simple application gives the effectiveness of the suggested algorithm.

II. PRELIMINARIES

The main concern of this section is to understand some concepts of input-to-state stability (ISS). Consider following discrete-time state-space nonlinear system

$$x(k+1) = f\left[x\left(k\right), u\left(k\right)\right] \tag{1}$$

where $u\left(k\right) \in \Re^m$ is the input vector, $x\left(k\right) \in \Re^n$ is a state vector,. f is general nonlinear smooth function $f \in C^\infty$. Let us now recall following theorem.

Theorem 1: For a discrete-time nonlinear system, the following are equivalent (Jiang and Wang, 2001).

- It is input-to-state stability (ISS).
- It is robustly stable.
- It admit a smooth ISS-Lyapunov function.

III. Stable Learning

A discrete-time cellular neural network's core operation is described by the following system of iterative equations:

$$x_{ij}(k+1) = \theta\phi\left(\begin{array}{c}\sum_{n,l\in N(ij)} A_{n-i,l-j}x_{ij}(k) \\ +\sum_{n,l\in N(ij)} B_{n-i,l-j}u_{ij}(k) + I\end{array}\right) \quad (2)$$

where x_{ij} is the state of the cell (neuron) in position ij, $i,j = 1\cdots n.$, that corresponds to the image pixel in the same position; u_{ij} is the input to the same cell, representing the luminosity of the corresponding image pixel, suitably normalized; A is a matrix representing the interaction between cells, which is local (as specified by the fact that summations are taken over the set N of indexes of neighborhood cells) and space-invariant (as implied by the fact that weights depend on the difference between cell indexes, rather than on their absolute values); B is a matrix representing forward connections issuing from a neighborhood of inputs. I is a bias. $N(i,j)$ is the set of indexes corresponding to cell ij itself and a small neighborhood (e.g. if neigborhood $r = 1$ then consider cell ij and its 8 nearest neighbors given a total of nine cells).

The function ϕ is nonlinear active function of the cellular neural network. In this paper we consider binary image processing, ϕ is sign function which is defined as

$$\phi(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

Remark 1: In order to apply stable learning algorithm, in this paper we propose a new CNN structure (2). The only different with the normal CNNs is that we introduce a positive parameter θ, $1 \geq \theta > 0$. When $\theta = 1$, it is the normal CNN. This modification does not destroy the good properties of CNN.

Remark 2: Another new idea of the CNN presented in this paper is that we use a Sigmoid function to approximate the sign function to realize off-line learning. $\phi(x)$ has the following form

$$\phi(x) = \frac{a}{1+e^{-bx}} - c$$

If a, b and c are selected suitable values, the two function are similar, see Fig. 1.

The operation performed by the network is fully defined by the so-called *cloning template* $\{A, B, I\}$. Moreover, under suitable conditions (Takashashi and Chua, 1998; Hanggi and Moschytz, 2000), and with time-invariant input u, a steady state is reached. This steady state depends, in general, on initial state values and input u. Images to be processed are fed to the network as initial state and/or input, and the result taken as steady state value, which realistically means a state value after some time steps (ranging normally from 10 to 100 according to the task).

Because there exist locality and space invariance in CNN weights, we transform CNN (2) into the following matrix

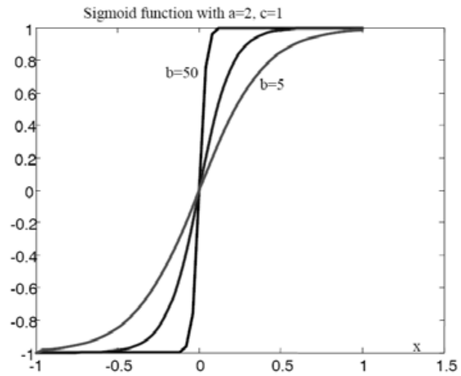

Fig. 1. Sigmoid function.

form

$$x_{ij}(k+1) = \theta\phi\left[\begin{array}{c} W_1(k)\lambda(x_{ij}(k)) + W_2(k)\lambda(u_{ij}(k)) \\ +W_3(k)\end{array}\right] \quad (3)$$

where the matrices $W_1(k) = [A_{n-i,l-j}]$, $W_2(k) = [B_{n-i,l-j}]$, $W_3(k) = [i]$ are the feedback template, control template, bias of i,jth CNN, or the weights of the neural network. They are $m \times m$ dimension vector, defined as

$$W_1(k) = \left[w_{11}^1 \cdots w_{1m}^1, w_{21}^1 \cdots w_{2m}^1, \cdots w_{m1}^1 \cdots w_{mm}^1\right]$$

In case of $r = 1$, $m = 3$. If $r = 2$, $m = 5$. $\lambda(x_{ij}(k))$ is a special selection function, it is defined as

$$\lambda(x_{ij}(k)) = \left[\begin{array}{c} x_{i-m,j-m}\cdots x_{i-m,j-m}, x_{i-m+1,j-m}\cdots \\ x_{i-m+1,j-m}\cdots x_{i+m,j-m}\cdots x_{i+m,j-m}\end{array}\right]^T$$
$$x_{i-m,} = 0 \text{ if } i \leq m$$
$$x_{,j-m} = 0 \text{ if } j \leq m$$

The desired images are represented as (3), which is bounded-input and bounded-output (BIBO) stable, *i.e.*, $x(k)$ and $u(k)$ are bounded. The desired state of $x_{ij}(k)$ is defined as $x_{ij}^*(k)$. According to the Stone-Weierstrass theorem (Cybenko, 1989), the desired images (3) can be written as

$$x_{ij}^*(k+1) = \theta\phi\left[W_1^*\lambda\left(x_{ij}^*(k)\right) + W_2^*\lambda(u_{ij}(k)) + W_3^*\right] + \mu(k) \quad (4)$$

where W_1^*, W_2^* and $W_3^*$are constant weights which can minimize the modeling error $\mu(k)$. Since ϕ is bounded functions, $\mu(k)$ is bounded as $\mu^2(k) \leq$ $\bar{\mu}$, μ is an unknown positive constant. The neuro identification error is defined as

$$e_{ij}(k) = x_{ij}(k) - x_{ij}^*(k)$$

From (4) and (3)

$$\begin{aligned} e_{ij}(k+1) &= \theta\phi\left[\begin{array}{c} W_1(k)\lambda(x_{ij}(k)) \\ +W_2(k)\lambda(u_{ij}(k)) + W_3(k)\end{array}\right] - \\ &\quad \theta\phi\left[\begin{array}{c} W_1^*\lambda(x_{ij}(k)) \\ +W_2^*\lambda(u_{ij}(k)) + W_3^*\end{array}\right] - \mu(k)\end{aligned}$$

Using Taylor series around the points of $W_1(k)\lambda(x_{ij}(k))+W_2(k)\lambda(u_{ij}(k))+W_3(k)$

$$\begin{array}{l}\phi\left[W_1(k)\lambda(x_{ij}(k))+W_2(k)\lambda(u_{ij}(k))+W_3(k)\right]\\ -\phi\left[W_1^*\lambda(x_{ij}(k))+W_2^*\lambda(u_{ij}(k))+W_3^*\right]\\ =\phi'\widetilde{W}_1(k)\lambda(x_{ij}(k))+\phi'\widetilde{W}_2(k)\lambda(u_{ij}(k))\\ +\phi'\widetilde{W}_3(k)+\varepsilon(k)\end{array}\tag{5}$$

where $\widetilde{W}_1(k)=W_1(k)-W_1^*$, $\widetilde{W}_2(k)=W_2(k)-W_2^*$, $\widetilde{W}_3(k)=W_3(k)-W_3^*$, $\varepsilon(k)$ is second order approximation error. ϕ' is the derivative of nonlinear activation function ϕ at the point of $W_1(k)\lambda(x_{ij}(k))+W_2(k)\lambda(u_{ij}(k))+W_3(k)$. So

$$\begin{array}{c}e_{ij}(k+1)=\theta\phi'\widetilde{W}_1(k)\lambda(x_{ij}(k))\\ +\theta\phi'\widetilde{W}_2(k)\lambda(u_{ij}(k))\\ +\theta\phi'\widetilde{W}_3(k)+\theta\zeta(k)\end{array}\tag{6}$$

where $\zeta(k)=\varepsilon(k)-\mu(k)$. The following theorem gives a stable off-line learning algorithm of CNN.

Theorem 2: If the cellular neural network (3) is used to process images, the following gradient updating law can make the identification error $e_{ij}(k)$ bounded (stable in an L_∞ sense)

$$\begin{array}{l}W_1(k+1)=W_1(k)-\eta(k)\phi'\lambda(x_{ij}(k))e_{ij}^T(k)\\ W_2(k+1)=W_2(k)-\eta(k)\phi'\lambda(u_{ij}(k))e_{ij}^T(k)\\ W_3(k+1)=W_3(k)-\eta(k)\phi' e_{ij}^T(k)\end{array}\tag{7}$$

where $\eta(k)$ satisfies

$$\eta(k)=\left\{\begin{array}{c}\dfrac{\eta}{1+\|\phi'\lambda(x_{ij}(k))\|^2+\|\phi'\lambda(u_{ij}(k))\|^2+\|\phi'\|^2}\\ 0\end{array}\right.$$

if $\frac{1}{\theta}\|e_{ij}(k+1)\|\geq\|e_{ij}(k)\|$, 0 otherwise and $0<\eta\leq 1$. The average of the identification error satisfies

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{k=1}^{T}e_{ij}^2(k)\leq\frac{\eta}{\pi}\overline{\zeta}\tag{8}$$

where $\pi=\dfrac{\eta}{1+\kappa}\left[1-\dfrac{\eta\kappa}{1+\kappa}\right]$, $\kappa=\max_k\left(\|\phi'\lambda(x_{ij}(k))\|^2+\left\|\phi'\lambda(u_{ij}(k))\right\|^2+\left\|\phi'\right\|^2\right)$, $\overline{\zeta}=\max_k\left[\zeta^2(k)\right]$, $T>0$ is identification time.

Demostración: Select Lyapunov function as

$$V(k)=\left\|\widetilde{W}_1(k)\right\|^2+\left\|\widetilde{W}_2(k)\right\|^2+\left\|\widetilde{W}_3(k)\right\|^2$$

where $\left\|\widetilde{W}(k)\right\|^2=\sum_{i=1}^n\widetilde{w}(k)^2=tr\left\{\widetilde{W}^T(k)\widetilde{W}(k)\right\}$. From the updating law (7)

$$\widetilde{W}_1(k+1)=\widetilde{W}_1(k)-\eta(k)\phi'\lambda(x_{ij}(k))e_{ij}^T(k)$$

So

$$\begin{array}{l}\Delta V(k)=V(k+1)-V(k)\\ =\left\|\widetilde{W}_1(k)-\eta(k)\phi'\lambda(x_{ij}(k))e_{ij}^T(k)\right\|^2-\left\|\widetilde{W}_1(k)\right\|^2\\ +\left\|\widetilde{W}_2(k)-\eta(k)\phi'\lambda(u_{ij}(k))e_{ij}^T(k)\right\|^2-\left\|\widetilde{W}_2(k)\right\|^2\\ +\left\|\widetilde{W}_3(k)-\eta(k)\phi' e_{ij}^T(k)\right\|^2-\left\|\widetilde{W}_3(k)\right\|^2\\ =\eta^2(k)\|e_{ij}(k)\|^2\|\phi'\lambda(x_{ij}(k))\|^2\\ -2\eta(k)\left\|\phi' x_{ij}(k)\widetilde{W}_1(k)e_{ij}^T(k)\right\|\\ +\eta^2(k)\|e_{ij}(k)\|^2\left\|\phi'\lambda(u_{ij}(k))\right\|^2\\ -2\eta(k)\left\|\phi'\lambda(u_{ij}(k))\widetilde{W}_2(k)e_{ij}^T(k)\right\|\\ +\eta^2(k)\|e_{ij}(k)\|^2-2\eta(k)\left\|\phi'\widetilde{W}_3(k)e_{ij}^T(k)\right\|\end{array}$$

From (6) we have

$$\begin{array}{c}\frac{1}{\theta}e_{ij}(k+1)=\phi'\widetilde{W}_1(k)\lambda(x_{ij}(k))\\ +\phi'\widetilde{W}_2(k)\lambda(u_{ij}(k))\\ +\phi'\widetilde{W}_3(k)+\zeta(k)\end{array}\tag{9}$$

Using (9) and $\eta(k)\geq 0$,

$$\begin{array}{l}2\eta(k)\left\|\phi' x_{ij}(k)\widetilde{W}_1(k)e_{ij}^T(k)\right\|\\ -2\eta(k)\left\|\phi'\lambda(u_{ij}(k))\widetilde{W}_2(k)e_{ij}^T(k)\right\|\\ -2\eta(k)\left\|\phi'\widetilde{W}_3(k)e_{ij}^T(k)\right\|\\ \leq -2\eta(k)\left\|e_{ij}^T(k)\right\|\left\|\frac{1}{\theta}e_{ij}(k+1)-\zeta(k)\right\|\\ =-2\eta(k)\left\|e_{ij}^T(k)\frac{1}{\theta}e_{ij}(k+1)-e_{ij}^T(k)\zeta(k)\right\|\\ \leq -2\eta(k)\left\|e_{ij}^T(k)\frac{1}{\theta}e_{ij}(k+1)\right\|+2\eta(k)\left\|e_{ij}^T(k)\zeta(k)\right\|\end{array}$$

If $\left\|\frac{1}{\theta}e_{ij}(k+1)\right\|\geq\|e_{ij}(k)\|$, $\left\|e_{ij}^T(k)\frac{1}{\theta}e_{ij}(k+1)\right\|\geq\|e_{ij}(k)\|^2$, since $0<\eta\leq 1$, ,

$$\begin{array}{l}\Delta V(k)\leq\eta^2(k)\|e_{ij}(k)\|^2\|\phi'\lambda(x_{ij}(k))\|^2\\ +\eta^2(k)\|e_{ij}(k)\|^2\|\phi'\lambda(u_{ij}(k))\|^2\\ -\eta(k)\|e_{ij}(k)\|^2+\eta(k)\|\zeta(k)\|^2\\ =-\eta(k)\cdot\\ \left[\begin{array}{c}1-\\ \eta\dfrac{\|\phi'\lambda(x_{ij}(k))\|^2+\left\|\phi'\lambda(u_{ij}(k))\right\|^2+\left\|\phi'\right\|^2}{1+\|\phi'\lambda(x_{ij}(k))\|^2+\|\phi'\lambda(u_{ij}(k))\|^2+\|\phi'\|^2}\end{array}\right]\cdot\\ e_{ij}^2(k)+\eta_k\zeta^2(k)\\ \leq -\pi e_{ij}^2(k)+\eta\zeta^2(k)\end{array}\tag{10}$$

where $\pi=\dfrac{\eta}{(1+\kappa)^2}$, $\kappa=\max_k\left(\|\phi'\lambda(x_{ij}(k))\|^2+\left\|\phi'\lambda(u_{ij}(k))\right\|^2+\left\|\phi'\right\|^2\right)$. Since $\pi>0$

$$n\min\left(\widetilde{w}_i^2\right)\leq V_k\leq n\max\left(\widetilde{w}_i^2\right)$$

where $n\times\min\left(\widetilde{w}_i^2\right)$ and $n\times\max\left(\widetilde{w}_i^2\right)$ are $\mathcal{K}_\infty$-functions, and $\pi e_{ij}^2(k)$ is an $\mathcal{K}_\infty$-function, $\eta\zeta^2(k)$ is a $\mathcal{K}$-function, so V_k admits the smooth ISS-Lyapunov function as in *Definition 2*. From *Theorem* 1, the dynamic of the identification error is input-to-state stable. The "INPUT" is corresponded

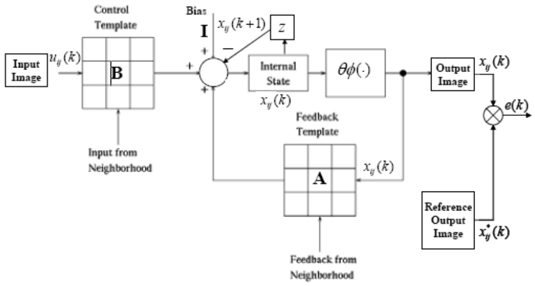

Fig. 2. Off-line training of CNN.

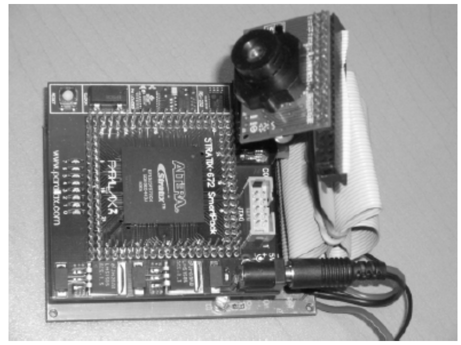

Fig. 3. Implementation of the digital camera M3188A on FPGA.

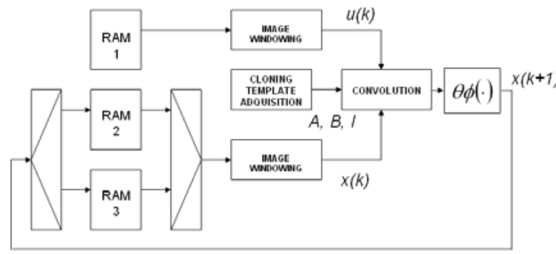

Fig. 4. Emulation of Discrete-time CNN on FPGA.

to the second term of the last line in (10), *i.e.*, the modeling error $\zeta(k) = \varepsilon(k) - \mu(k)$, the "STATE" is corresponded to the first term of the last line in (10), *i.e.*, the identification error $e_{ij}(k)$. Because the "INPUT" $\zeta(k)$ is bounded and the dynamic is ISS, the "STATE" $e_{ij}(k)$ is bounded. If $\frac{1}{\theta}\|e_{ij}(k+1)\| < \|e_{ij}(k)\|$, $\Delta V(k) = 0$. $V(k)$ is constant, $W_1(k)$, $W_2(k)$ and $W_3(k)$ are constants. Since $\|e_{ij}(k+1)\| < \theta\|e_{ij}(k)\|$, $e_{ij}(k)$ is also bounded. (10) can be rewritten as

$$\Delta V(k) \leq -\pi e_{ij}^2(k) + \eta\zeta^2(k) \leq \pi e_{ij}^2(k) + \eta\overline{\zeta} \qquad (11)$$

Summarizing (11) from 1 up to T, and by using $V(T) > 0$ and $V(1)$ is a constant

$$\begin{array}{c} V(T) - V(1) \leq -\pi\sum_{K=1}^{T} e_{ij}^2(k) + T\eta\overline{\zeta} \\ \pi\sum_{K=1}^{T} e_{ij}^2(k) \leq V(1) - V(T) + T\eta\overline{\zeta} \leq L_1 + T\eta\overline{\zeta} \end{array}$$

(8) is established. ■

Remark 3: The condition $\frac{1}{\theta}\|e_{ij}(k+1)\| \geq \|e_{ij}(k)\|$, $1 \geq \theta > 0$ is dead-zone. When $\theta \to 0$, this condition disappears because $\frac{1}{\theta}\|e_{ij}(k+1)\| \geq \|e_{ij}(k)\|$ is always correct. When $\theta = 1$, CNN (2) becomes the normal CNN, but in order to assure the learning process be stable, we need that the training occurs when the identification error increases. When $1 > \theta > 0$, the smaller θ is, the smaller the dead-zone becomes. The off-line training is shown in Fig. 2.

IV. Real time Vision

IV-A. FPGA Implementation

In this section we explain the technology used to implement our real time computer vision. The main components are:

1. ALTERA Stratix Field programmable gate array (FPGA).
2. Digital camera M3188A by Quasar.
3. Low level language called Very High Speed Integrated Circuit Hardware Description Language (VHDL).

An FPGA is based on these technologies, but the interconnections can be defined by the user via Configurable Logic Blocks (CLBs). So, when the design is finished the code is loaded onto the FPGA, then the CLBs are configurated to carry out this specific design. The CLBs are reconfigurable and interconnected blocks, so they can be re-used. This property allow to the user obtain a very complex designs. To acquire the real images (left side of the images) we used a digital camera M3188A connected to the ALTERA Stratix (see Fig. 3).

For developing the emulation of the Discrete-time CNN (2) we design the general diagram shown in Fig. 4, then we programming each module in the low level language VHDL. The RAM components are used to store the images until the state $x(k+1)$ achieves the stationary state or a predefine number of iterations is reached. The stationary state means that there is not any change in the images contained between the $x(k)$ and $x(k+1)$ states. For our application we select the number of iterations that each cloning template will be applied.

IV-B. PC Implementation

We used Matlab to simulate a discrete-time CNN (2). Our program performs three basic steps for each test image.

1. Capture or load an image into a matrix variable. and normalize with values of -1 and 1 (white and black).
2. Process the image with the CNN structure.
3. Normalize the resulting in black and white image matrix to the values 255 and 0, respectively.

The processing time is measured by applying the above steps to each test image. For a fair comparison with respect to the FPGA system, we do not include the time required for loading, saving and displaying the images, thus considering only the processing time.

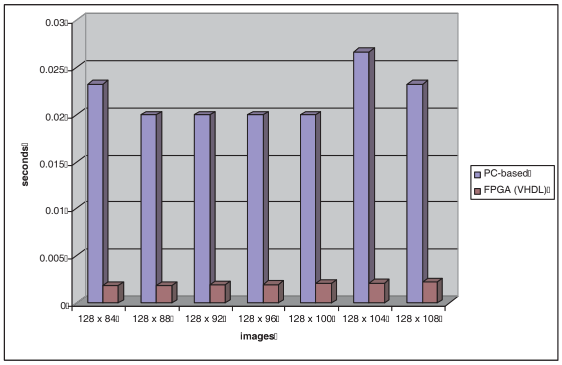

Fig. 5. Comparision between executing times with FPGA and PC-based

IV-C. Experimental Results

In this subsection we select a cloning template used for eliminate the small noise(SOK) in the image (12), this task in very important in different computer vision tasks which are desired templates for the training.

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; I = 0 \qquad (12)$$

To test our updating law to update the values of the CNN, we use the structure shown in Fig. 5, the goal is find an cloning template (A, B, I) to process the input image and get the reference output image. The learning process is:

1. Select a pair of input-output images. Also we have an third image that represents the current output of our CNN.
2. Using this pair of images our Cellular Neural Network adjust the values of the cloning template (A, B, I), the error signal is the difference between the current output image of the CNN and the given reference output image, we select the following initial conditions.

$$A^o = 0{,}1 \quad B^o = \begin{bmatrix} 0{,}2 & 0{,}1 & 0{,}2 \\ 0{,}2 & 0{,}3 & 0{,}2 \\ 0{,}3 & 0{,}1 & 0{,}3 \end{bmatrix} \quad I^0 = 0{,}2$$

3. Now we apply the stable gradient updating laws, the final values for the cloning template are:

$$A^* = 1{,}0 \quad B^* = \begin{bmatrix} 0 & 1{,}0 & 0 \\ 1{,}0 & 2{,}0 & 1{,}0 \\ 0 & 1{,}0 & 0 \end{bmatrix} \quad I^* = 0$$

IV-D. Processing time

Finally, we compare the time required to process an image with our FPGA implementation versus PC. We consider different sizes of images and we apply the SOK cloning template. The result is shown in Fig. 5.

V. CONCLUSIONS

In this paper, we proposed a new stable technique to obtain a cloning template to extract required information from an input image for cellular neural networks. This technique could be used to improve the performance for a mobile robot control (**?**). As we can see in Figure IV-D, is almost ten times faster process the images with the CNN implemented in a FPGA with VHDL language, we can process 45 images per second. This are very important for robotic tasks. The main contributions of this paper are: 1) we proposed a stable updating law for CNN, 2) since images processing is not done by a PC which is connected to the mobile robot (wired or wireless), the autonomy of mobile robots can be improved with this technique, 3) we give real time algorithm for images process, it is possible to extend this algorithm to our optical stereo vision model (Moreno-Armendariz et al., 2003).

VI. ACKOWLEDGMENT

M.A. Moreno-Armendariz thanks to SIP-IPN grant on project 20062090.

REFERENCES

L.O. Chua, L. Yang (1988), Cellular Neural Networks: Theory, *IEEE Trans. Circ. Syst.*, CAS-35 1257–1272.

L.O. Chua, T. Roska (1993), The CNN Paradigm, *IEEE Trans. Circ. Syst.*, CAS-I-40, 147–156.

G. Cybenko (1989), Approximation by Superposition of Sigmoidal Activation Function, *Math.Control, Sig Syst* 2 (1989) 303-314.

H. Harrer, J.A. Nossek (1992), Discrete-time Cellular Neural Networks. *Int. J. of Circ. Th. Appl.*, 20, 453–467.

Hanggi, M. Moschytz, G.S. (2000), An analysis of CNN settling time, *IEEE Trans. Circ. Syst. I*, 47, (1), 9–24.

Z.P. Jiang and Y. Wang (2001), Input-to-State Stability for Discrete-Time Nonlinear Systems, *Automatica*, 37, 857–869.

C.T.Lin and G.Lee (1996), *Neural fuzzy systems: A neural-fuzzy synergism to intelligent systems*, Prentice-Hall Inc., NJ.

L. Liñan, S. Espejo, R. Domínguez-Castro, A. Rodríguez-Vázquez (2002), ACE4K: An Analog I/O 64x64 Visual MicroProcessor Chip with 7-bit Analog Accuracy,*Int. J. of Circ. Th. Appl.*, 30, 89–116.

M.A. Moreno-Armendariz, J. Aguilar-Luna, E. Gómez-Ramírez, X. Vilasís-Cardona (2003), An optical stereo vision model, *Proceedings of the IASTED Internacional Conference on Circuits, Signals, and Systems*,México.

E. Radwan, E. Tazaki (2004), Rough sets and genetic algorithms in learning cellular neural networks cloning template for decision making system, *International Journal of Neural Systems*, 14, 57–68.

Takashashi, N.; Chua, L.O. (1998), On the complete stability of nonsymmetric cellular neural networks, *IEEE Trans. Circ. Syst. I*, 45, 7, 754–758.

W. Yu, X. Li (2004), Fuzzy identification using fuzzy neural networks with stable learning algorithms, *IEEE Transactions on Fuzzy Systems*, 12, 3, 411–420.

W.Yu and X.Li (2003), Discrete-time neuro identification without robust modification, *IEE Proceedings - Control Theory and Applications*, 150, 3, 311–316.

W.Yu and X. Li (2001), Some Stability Properties of Dynamic Neural Networks, *IEEE Trans. Circuits and Systems, Part I*, 48, 1, 256–259.