Real-time Inspection of Metal Laminates by means of CNN's

Victor M. Preciado [a], Domingo Guinea [b], Rodrigo Montufar [b] and Jose Vicente [c]

[a] Escuela de Ingenieros Industriales, Univ. de Extremadura, Avda. de Elvas s/n, 06071 Badajoz, Spain

[b] Instituto de Automatica Industrial- Spanish Council for Scientific Research, Arganda del Rey, 28500 Madrid, Spain

[c] Escuela Politécnica, Univ. de Extremadura, Cáceres, Spain

ABSTRACT

Analog CNN array computer arises as an alternative to traditional digital processors in many industrial inspection like visual quality control of metal laminates, capable of make in a single chip Tera equivalent operations per second. A 4096 analog CNN processor array is able to perform complex space-time image analysis, being much faster than a camera-computer system in continuous inspection applications. Both chips have been implemented in CMOS technology and they are managed by a 32-bit high-performance low-cost micro-controller that closes the pan, tilt, lighting, focus and zoom loops required in the implementation of the active vision strategies. Several convolution masks for the Cellular Processors has been selected to detect particular changes in the texture, size, direction or orientation of the image entities, reprogramming "on the fly" the pixel resolution or shape when necessary. Laboratory results present these Cellular Processors and multiple resolution imager circuits as a promising architecture for visual inspection of industrial processes in real time.
The traditional image processing techniques (filtering, segmentation, interpretation) require a lot of computational effort due to data on each pixel are computed in a sequential way and the path of information is an analog/digital converter. The delay accumulation create in this process is unacceptable in real time image processing because of the information flow created in the usual vision tasks (e.g. automatic industrial inspection, vision problems in robotics, pattern analysis, etc.). Thus, the use of a massive parallel architecture working with analog signals avoid the previous problems. This is just the basis idea of Cellular Neural Network (CNN´s): An array of analogic dynamic processors which cells interact directly within a finite local neighborhood. The local CNN conectivity allow its realization as VLSI chips that can operate at a very high speed and complexity: 0.3TeraXPS performance for a 10x10mm2 chip using a 2-um technology in a robust implementation.

Keywords: Cellular Neural Network, Visual Processing, Industrial Inspection, Parallel Computation, Real Time Processing, Automatic Templates Generation, Genetic Algorithms.

1. INTRODUCTION

In this woks are going to be proposed a visual inspection systems based in the Cellular Neural Network paradigm. The main advantages in the use of this paradigm lay on the effective VLSI implementation of this kind of Neural Networks, being capable of operate at a very high speed and complexity: 0.3TeraXPS performance for a 10x10mm2 chip using a 2-um technology in a robust implementation. Many industrial inspection such visual quality control of metal laminates require billions of operations per second. Analog CNN array computer arises as an alternative to traditional digital processors, capable of make in a single chip Tera equivalent operations per second. A 4096 analog CNN processor array is able to perform complex space-time image analysis, being much faster than a camera-computer system in continuous inspection applications.

The traditional image processing techniques (filtering, segmentation, interpretation) require a lot of computational effort due to data on each pixel are computed in a sequential way and the path of information is an analog/digital converter. The delay accumulation create in this process is unacceptable in real time image processing because of the information flow created in the usual vision tasks (e.g. automatic industrial inspection, vision problems in robotics, pattern analysis, etc.). Thus, the use of a massive parallel architecture working with analog signals avoid the previous problems. This is just the basis idea of

Cellular Neural Network (CNN´s): An array of analogic dynamic processors which cells interact directly within a finite local neighborhood.

2. MODELLING THE DYNAMIC OF CELLULAR NONLINEAR NETWORK

This is just the basis idea of a Cellular Neural Network (CNN´s): An array of analogic dynamic processors which cells interact directly within a finite local neighborhood. The local CNN conectivity allow its realization as VLSI chips that can operate at a very high speed and complexity.

The dynamic of the array can be described by the following differential equations:

$$C \cdot \dot{V}x(t) = -R_x \cdot Vx(t) + A * Vo(t) + B * Vi(t) + I(t) \tag{1}$$

$$Vo = \frac{1}{2} \cdot (|Vx + K| - |Vx - K|) \tag{2}$$

$$N_r(i,j) = \{C(k,l) | max\{|k-i|, |l-j|\} \leq r\} \tag{3}$$

where *i j* refers to a grid point associated with a cell on the 2-D grid, and *k l* is a grid point in the neighborhood within a radius *r* of the cell *i j*. Equation (1) describes a nonlinear dynamical system. In this equation we can recognize several terms: the resistive term make the dynamic of the CNN to be stable at a value of 0 when the other terms don't exist, the term with A indicates the effect of the output image in the dynamic of the array, so this term is a feedback term that make the dynamic of the array complicated. This element improve the potenciality of the net in many application but it produce such a complicated dynamic that make the task of the templates designer difficult. The term with the B matrix is the effect of the input image in the dynamic of the array and the I image is an offset value. Due to the equation (2) is included in our system a piecewise linear function of saturation. In the equation (3) the concept of neighborhood is defined.

In conclusion, *C* is an input capacitor, *R* an input resistance and *I* an input bias current. V_{ykl} *(t)*represents the neural activity, whereas $V_{yij}(t)$ is the output of the network and V_{uij} is a fixed external input to the network. *B* and *A* are connection matrices that respectively describe the input and feedback connectivity.

In many applications, the cloning templates A, B and the threshold current I are space invariant when the template is space-invariant each cell is described by a simple identical cloning template defined by two (2*r*+1)x(2*r*+1) real matrices, as well as the constant term *I*. In this case every neuron has the same fixed synaptic weights, where N_r is the neighbourhood to which the synaptic connectivity of a neuron extends. So, the CNN will process local properties in the input image performing its convolution with a kernel defined by the cloning template. This feature makes the model very well adapted to image processing.

As the network will be devoted to this kind of tasks, it is convenient to represent equation (1) by the approximation of a difference equation of the form :

$$Vx[n+1] = Vx[n] + \frac{h}{C}\left(-\frac{1}{R}Vx[n] + A * Vo[n] + B * Vi[n] + I\right) \tag{5}$$

where *h* is the time step used to compute each iteration, *A* and *B* are the cloning templates. This equation is also used in the simulation of the net by means of a digital processor implicating Euler of Runge-Kutta 4 integrations methods.

In the equation than model the dynamic of the processors array can be seen a feedback term that difficult the design of the CNN by adjusting the classical methods of digital filtering. Besides, this term can't be omitted because is the element that gives it potentiality to the net.

3. CONSIDERATIONS IN THE DESIGN OF TEMPLATES

Nowadays several prototype CNN architectures have been implemented as VLSI chips, showing the capability of extremely high speed compared with traditional digital image processing tools. The proliferation of more and more sophisticated CNN architectures, and the increasing effort to implant practical system for industrial applications based in CNN chips, make necessary the programming of software development tools for template design.

Unfortunately, there is no methodology, even in the simplest case, that allows us to get the template associated to an determined signal processing. Consequently with the absence of a general method, the search of a solution to a determined problem could result fruitless due to an inadequate focus in the way of resolving the problem. Next a general method for the design of templates based on genetic algorithms and evolution programs will be proposed.

4. GENETIC GENERATION OF CLONING TEMPLATES

The discrete search space, the nonlinearity of the fitness function and the discontinuity of the space, make Genetic Algorithm (GA) a good candidate for this optimization task.

Present work presents a CNN system with the feature of automatic template generation from input/output training examples by means of a Genetic Algorithm. The simulator processes an input image from an initial state, and using a certain template, it will produce an output image performing a particular image processing operation. With the extra feature of automatic template generation the process is inverted, so that with an input image and the desired output image, the genetic algorithm is capable for searching a template able to generalize the operation.

To get an automatic design we must encode the template coding in the form of chromsomes. Now we must determine the chromosome length needed for the completely definition of our design parameters. One of the main features of CNN chips is its reliable and robust implementation as VLSI, that is due to the fact that local connectivity with radius unity is routed into a silicon wafer. So the number of parameters in a whole template set (neighborhood within a radius r=1) is 19; 9 parameters for each matrix (being A and B 3x3 matrices) and one more for the bias current I. Taking into account that analog VLSI chips have a certain template parameters accuracy, there will be utilized 8 bits for encoding each template parameter. So the utilized length of the chromosome for encoding the template parameters is 19 (parameters) x 8 (bits/parameters) = 152 bits.

This chromosome length can be drastically reduced in the case that the image processing task have a simetric behaviour (e.g. border detection, averaging, halftoning etc.), in a way that if we rotate a certain angle an input image and initial state image that give a concrete output, the new output image is the same that the before output by rotating the image the same angle. Thus, under these four constraints (a12=a21=a23=a32; a11=a13=a31=a33; b12=b21=b23=b32; b11=b13=b31=b33), the cloning template is reduced to 7 representative parameters (3 parameters for each matrix and another one for the bias current), and the chromosome is a 56-bits string.

$$A := \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \qquad B := \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix} \qquad I$$

a11,a12,a13,a21,…a32,a33,b11,b12,…,b32,b33,I => 19 parameters
n bits/parameter => 19 n bits

Fig. 1. Chromosome definition diagram.

The speed of convergence in the algorithm rely on the options set selected and the genetic operators values. So the options that can be changed in the program are the following:

Crossover selects genes from parent chromosomes and creates a new offspring. The way to do this is to choose randomly some crossover point and copy everything before this point from a first parent and then everything after this crossover point copy from the second parent. In this algorithm a one-crossover point with adaptable probability has been used.

After the crossover is performed, mutation take place. This is to prevent falling all solutions in population into a local optimum of solved problem by changing randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1

Population size determines what is the number of chromosomes in population (per generation). If there are too few chromosomes, GA have a few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA converges slows down. Performed work shows that after some limit (which depends mainly on encoding and the problem) it is not useful to increase population size, because it does not make solving the problem faster. In this genetic algorithm has been used a population size equal to 160 in the case of nonsimetric templates and equal to 60 on the case of *priori* knowledge of symmetric CNN dynamic.

The developed program allows to adjust the number of generations for the algorithm to converge, and also the number of necessary steps to have a good approximation of the difference equation (5). The number of generations depends, for the proposed applications, on the complexity of the image processing task and the steps to solve the difference equation.

This algorithm must search the template that minimize an energy function proportional to the to the difference between pixels from the current output image and the desired one. This function is the following:

$$f(\vec{p}) = \beta \cdot \sum_{pixels} \left| V_{uij} - V_{yij}(\vec{p}) \right| \tag{6}$$

where β=3 if the output image is completely white or black and β=1 otherwise.

For task with only a relatively small number of black (or white) output pixels, this function yields a high fitness for templates that create white (or black) image. To prevent these templates for becoming too dominant, they are penalized by multiplying their fitness values.

5. FLOW OF THE EVOLUTIVE ALGORITHM

Once the previous parameters have been adjusted at fit values and correct options have been select in the program, the algorithm flow is the following:

a) Loading the input image, initial state for iterative process and desired output image.

b) Random generation of the initial population.

c) Evaluation of each population individual by equation (6).

d) Arranging the individual attend to the previous evaluation .

e) Printing the current best output image, cloning template, number of iterarions and value of the evaluation function for the best current solution.

f) Obtaining a new generation of individuals by:

g) Selecting two parents by a roulette wheel selection.

h) Crossing this parents by a one-crossover process taking into account the crossover probability parameter.

i) Mutating the offspring keeping in mind the mutation rate.

j) Checking the stop condition (iteration number or evaluation function value).

k) Return to step 3.

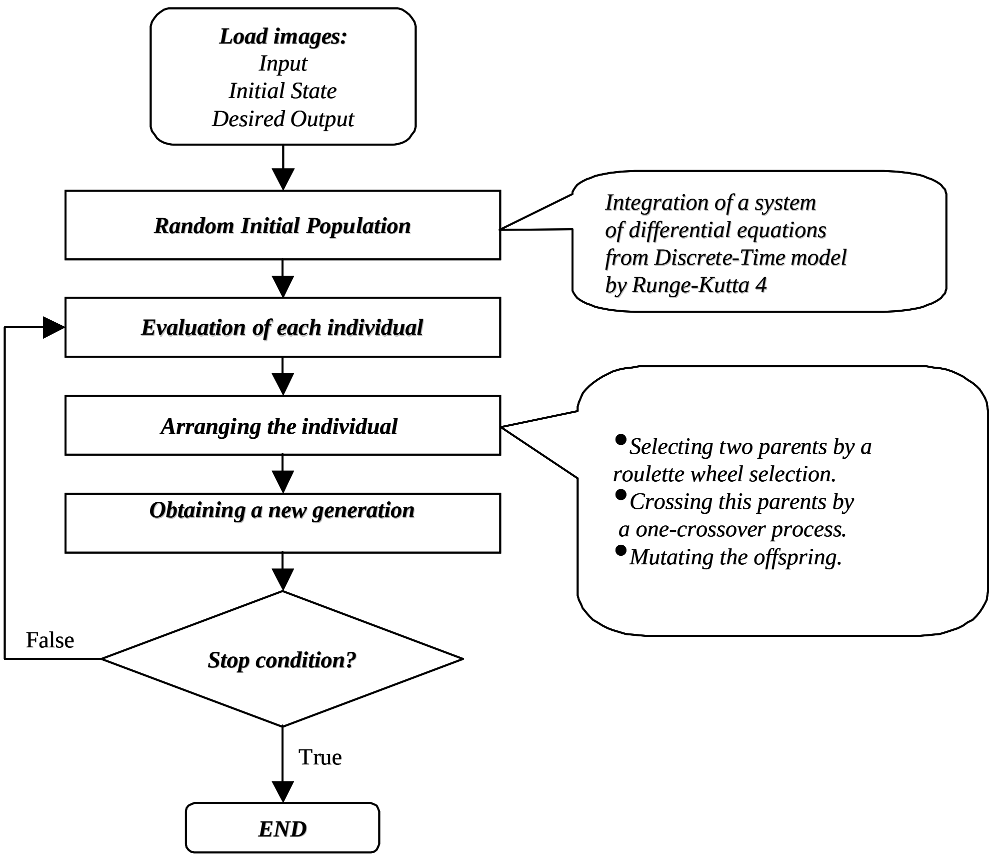


Fig. 2. Automatic template generation using GA´s

5.1. Genetic Templates in Border Detection

In the following example the genetic algorithm has been used to learn some templates able to perform a border detection.

$$A := \begin{pmatrix} -4.121 & 3.070 & -4.121 \\ 3.070 & 3.729 & 3.070 \\ -4.121 & 3.070 & -4.121 \end{pmatrix} \quad B := \begin{pmatrix} -0.443 & -0.223 & -0.443 \\ -0.223 & 3.564 & -0.223 \\ -0.443 & -0.223 & -0.443 \end{pmatrix} \quad I := 1.588$$

Fig. 4. Final Template Solution for the task above.

It can be seen that the generated template differs from the well known border detection template performing the same task. The CNN simulator parameters used are: Crossover probability=80%, Mutation rate=4%, Number of iteration=200, Population size=60, Transient steps=10, Symmetric template option checked.

In fig. 2. the input image has been drawing containing various kind of geometric figures, it has been made to make a parallel optimization with respect to robustness in a single stage.

6. CONSIDERATIONS IN THE SINGLE TEMPLATE GENERATION

Nowadays CNN architectures implemented as VLSI chips shows the aptitude of extremely high speed compared with traditional digital image processing tools. The proliferation of more and more sophisticated CNN architectures, and the increasing effort to implant practical system for industrial applications based in CNN chips, make necessary the programming of software development tools for template design.

This work presents a CNN simulator with the feature of automatic template generation from training examples by mean of a Genetic Algorithm [3]. The simulator processes an input image from an initial state, and using a certain template, it produce an output image performing a particular image processing operation. With the extra feature of automatic template generation the process is inverted, so that, with an input image and the desired output image, the genetic algorithm is able to search a template that perform that operation. The discrete search space and the nonlinearity of the fitness make Genetic Algorithm (GA) a good candidate for this optimization task.

6.1. Search Space

The number of parameters in a whole template set (neighborhood within a radius *r*=1) is 19. Taking into account that analog VLSI chips have a certain template parameters accuracy, 8 bits are utilized to encode each template parameter. So the chromosome length utilized to encode the template parameters is 152 bits (19 parameters x 8 bits/parameters).

This chromosome length can be drastically reduced in the case that the image processing task have a symmetric behavior (e.g. border detection, averaging, halftoning etc.). Thus, under symmetric behavior the cloning template is reduced to 7 representative parameters (3 parameters for each matrix and another one for the bias current), and the chromosome is a 56-bits string.

The speed of convergence in the algorithm rely on the options set selected and the genetic operators values. The options that can be changed in the program are the following: Crossover probability, Mutation rate, Population size and Transient options. Besides, the developed program allows to adjust the number of generations for the algorithm to converge, and also the number of necessary steps to have a good approximation of the difference equation (5).

This algorithm must search the template that minimize an energy function proportional to the to the difference between pixels from the current output image and the desired one. This function is the following [4]:

$$f(\vec{p}) = \beta \cdot \sum_{pixels} \left| V_{uij} - V_{yij}(\vec{p}) \right| \qquad (6)$$

where β=3 if the output image is completely white or black and β=1 otherwise.

7. GENETIC GENERATION OF CASCADE-TEMPLATES

Going beyond, there is many advanced applications with CNN´s where several templates are working consecutively for a concrete purpose. In this case the information to encode is the 19 parameters of each stage and two images (input and initial states).

The way it has been done is by a first stage with input image equal to the image to process, and an initial state selected from a fixed set of initial states. This initial states has been selected from numerous samples of simple templates, concluding that almost all the initial states are including in a reduced collection of images (the input one, white, black and grey scale, etc.)

7.1. Search Space Dimension and Pruning

Thus, the number of bits needed for encoding each stage in a cascade-template case are the same than in the single-template case plus 8 bits needed for encoding both input and initial images number (4 bits for encoding each image number from the collection. The search space for the genetic search is Nx(152+8) bits in the general case, being N the number of stages.

We can drastically reduce the dimension of this cube if the symmetric condition is utilized, obtaining a search space of Nx(56+8) bits. However, the search space is too big for assuring the convergent of the evolutionary process. So, a heuristic method has to be used to make the search well-conditioned for the algorithm to perform. This consist in the use of a library of well-known templates like initial parameters of each stage.

8. MULTI-TEMPLATE TREE AND GENETIC PROGRAMMING

In this part of the work, a general methodology for developing a general case of analogic algorithms utilizing combinations of single templates to perform complex image processing tasks is presented. Some concepts in this section are related with AI theory.

8.1. Notation and Representation of the Tree

The situation is that each template can be seen like an operator which acts directly on two images, and this can be represented exactly by a diagram of the type called *tree*.

The notation utilized is the *prefix* or Polish notation due to Lukasiewicz. According to this notation an operator *O* which performs an action on two objects *x* and *y* is represented *O x y*. In our case, O consist of a single template, x is an input image and y is the initial state for the CNN differential equation to be solved. No parenthesis are needed and the principle operator for any term appear at the head of that term.

In the case of CNN analogic algorithms binary operator are the templates which need two external input images for its process to be performed (i.e. logic operators, CCD or objects extractor templates, etc.). Unary operators only need one image because either input image or initial state image are previously preset to a black, gray or white intensity level. In this way we can express the tree in fig.1 like (parenthesis are not needed, only used for making the expression clearer): *Tem.4(Tem.2(Tem.1(X, B), A), Tem.3(X, A)).*

The convenience of this notation can be seen in the random generation of trees representing analogic algorithms. The individuals in the initial random population and the offspring produced by each genetic operation must be all syntactically valid executable programs, we can assure this by checking that the next useful theorem, due to Rosenbloom is carried out in each new node of the tree.

A sequence of symbols S in prefix notation is a well-formed expression if and only if:

rank (S)=-1;

rank (sub-expression on the left of S)≥0;

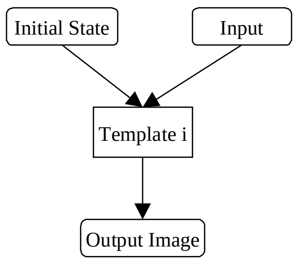
where rank is defined by:

rank (binary operator)=1;

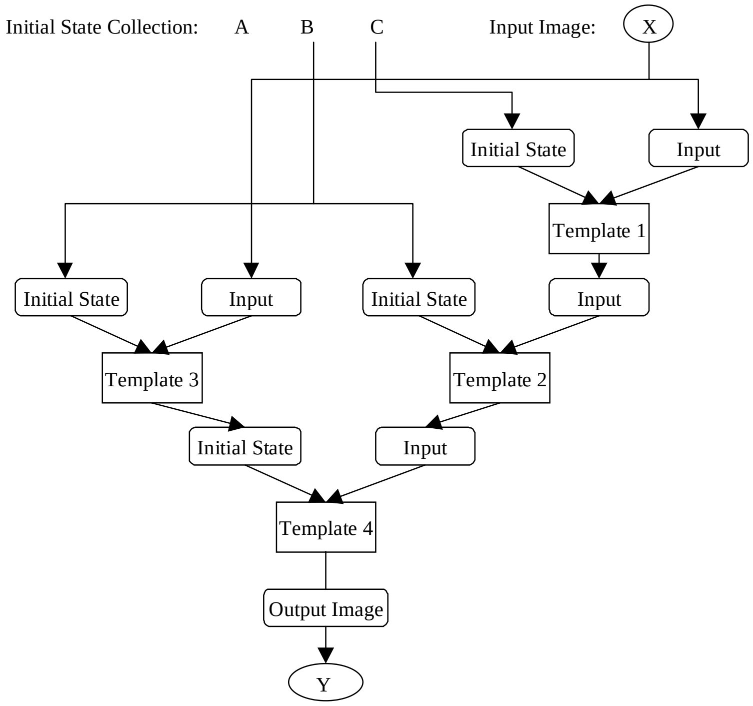
rank (unary operator)=0;

rank (constant)=-1;

rank (S1 concatenated with S2)=rank (S1)+rank (S2).



Tree node configuration.



Tree example

Fig. 3. Tree and node example.

8.2. Search Space and its Reduction by mean of an Heuristic Method

The representation in the machine can be done by constructing a table in which all variables (images) and operators (template) are listed. For each symbol we give three items of information: operator and two variables represented like (S,L,R).

In any case L and R can themselves be trees; that is, these variables can be a pointer to another operator in the table, that is, indices showing where in the table these are to be found, and therefore the relevant operators. The terminal symbols of the tree are the variables and constants of the expression and have no L, shown by setting L=0.

Once the tree has been encoding we must performance the genetic search for selfprogramming the analog algorithm. In this case we start in a library of well-known templates and a collection of initial states. In this way we have search an algorithm able to selfprogramme.

9. APPLICATIONS IN INDUSTRIAL INSPECTION

In this point some real examples of the analog CNN array computer for visual quality control of metal laminates are going to be shown. The task consist of trying to extract defects and imperfections of a metal laminate automatically from an input image in gray scale.

In this paper the solution proposed consist of a simple template like it is shown below:

$$A := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad B := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad I := K$$

Fig. 6. Proposed template for metal laminate defects detection.

In this simple template the control matrix applied to the input image has all its elements equal to zero, so the input image doesn't have influence in the CNN output image. The metal laminate image are loaded into the initial condition of the CNN, it can be seen that, what the central positive element in the feedback matrix A produces is a continuos increasing or decreasing in the pixel value to its saturation if the pixel value is major or minor than K respectively. In another way, the value K fix the threshold that one pixel must overcome to be converted into white; in other case it is converted into black.

The main problem in this proposal is to determine this threshold K for extracting the desired pixels avoiding to extract pixels from the background. So we must give to K a value slightly major than the maximum value of a pixel from the background. With this purpose we select randomly a small area from the metal laminate and compute the global maximum value of every pixels with the following template:

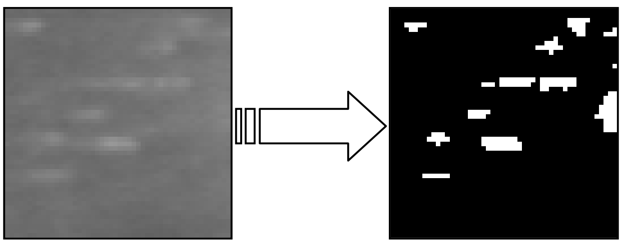
$$A := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad B := \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad I := 0.05$$

Fig. 7. Propagating template for calculating global maximum.

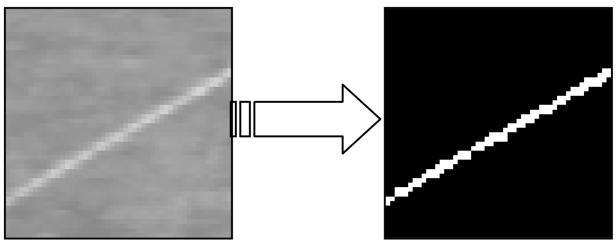
Before taking this value like the maximum value from the background, we must assure that the area randomly selected doesn't contain any defect. For making it we calculate the gradient of this area and compare the global maximum value with the averaging value of the image evaluated with a recursive low-pass filter. If the difference between both values is high, we consider that the area contains any defect and select another new small area.

When the global maximum value of the background is calculated, the parameter K take a value slightly major than this one. By means of this process the following input/output imagen can be obtained:

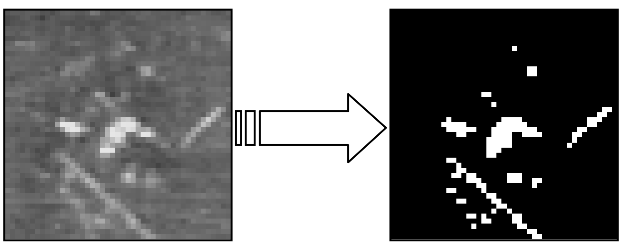
Input Images *Output Images*



Little scrapes in the surface



Single scratch in the metal surface



Pricked surface

Laboratory results present these Cellular Processors as a promising architecture for visual inspection of industrial processes in real time.

10. CONCLUSIONS

Actual CNN VLSI chips shows an extremely high speed, e.g. relatively small chips with 20x20 dimensions implemented in a 0.7 μm CMOS technology with τ=5 μs cell time constant, which means about 10000 op/sec, supposing a global convergence time of 20 τ in worst case (propagating template). This suppose an unprecedented magnitude order jump in computation time for image processing tasks.

The design template task has been usually focused like an extrapolation of the traditional methods for digital image processing, or more recently, has been used complex mathematical techniques like mathematical morphology and PDE related methods. Every case, the success of the template design rely on the designer experience. Automatic template generation by GA´s suppose a general methodology for this task, so the CNN programmers are released of the heavy charge that suppose template design and, in this way, effort can be inverted in develop analogic algorithm for solving general vision tasks.

ACKNOWLEDGEMENTS

This work has been funded by the spanish CICYT proyect SIVA (Integrated System for Active Vision). We thank also our partners from CNM of Seville and the Institute of Optics.

REFERENCES

1. L. O. Chua, L. Yang. "Cellular Neural Networks: Theory". IEEE Trans. on Circuits and Systems. Vol. 35, No. 10. pp. 1257-1272. October 1988.
2. L. O. Chua, L. Yang. "Cellular Neural Networks: Applications". IEEE Trans. on Circuits and Systems. Vol. 35, No. 10. pp. 1273-1290. October 1988.
3. L. O. Chua, T. Roska. "The CNN Paradigm". IEEE Trans. on Circuits and Systems. I: Fundamental Theory and Applications. Vol. 40, No. 3. pp. 147-156. March 1993.
4. S. Espejo, R. Carmona, R. Domínguez-Castro, A. Rodriguez-Vazquez, "A VLSI oriented continuous-time CNN model", Int. Journ. Of Circuit Theory and Applications, vol. 24, pp. 341-356 (1996).
5. "Cellular neural network design for solving specific image-processing problems", Cellular Neural Networks, pp. 191-199, De. Wiley (1993)
6. J. M. Cruz and L. O. Chua, "Design of high speed high density CNN´s in CMOS technology", Int. J. Circuit Theory and Applications, pp. 555-572, vol. 20, 1992.
7. P.Szolgay, I.Kispal and T.Kozek, "An Experimental System for Optical Detection of Layout Errors of Printed Circuits Boards Using Learned CNN Templates", Proceedings of the International Workshop on Cellular Neural Networks and their Applications (CNNA-92), pp. 203-209, Munich, 1992.
8. T.Sziranyi and M.Csapodi, "Texture classification and Segmentation by Cellular Neural Network using Genetic Learning", Computer Vision and Image Understanding, Vol. 71, No. 3, pp. 255-270, September 1998.
9. P.L.Venetianer, F.Werblin, T.Roska and L.O.Chua, "Analogic CNN Algorithms for some Image Compression and Restoration Tasks", IEEE Transactions on Circuits and Systems, Vol. 42, No. 5, 1995.
10. J. R. Koza: "Detailed Description of Genetic Programming". Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.
    R. Caponetto, M. Lavorgna, A. Martinez and L. Occhipinti, "Cellular Neural Network Simulator for Image Processing Applications", 1998 Fifth IEEE International Workshop on Cellular Neural Network and Their Applications Proceedings, pp.360-365. April 1998.

11. Martin Hanggi and George S. Mostchytz, “Stochastic and Hybrid Approaches Toward Robust Templates”, 1998 Fifth IEEE International Workshop on Cellular Neural Network and Their Applications Proceedings, pp.366-371. April 1998.
12. V. Preciado, D. Guinea, J. Vicente, M.C.Garcia-Alegre and A.Ribeiro, ”CNN´s design by GA´s”, European Congress on Computational Methods in Applied Sciences and Engineering Proceedings, to be published in 2000.
13. J. M. Cruz and L. O. Chua, “A 16 × 16 Cellular Neural Network Chip: the First Complete Single-Chip Dynamic Computer Array with Distributed Memory and Grayscale I/O”, *Analog Integrated Circuits and Signal Processing*, 15 (3), 1998, 227-238.
14. A. Paasio, A. Dawidziuk, K. Halonen and V. Porra, “Minimum size 0.5 μm CMOS Programmable 48 × 48 CNN Test Chip”, *Proc. of the 1997 European Conference on Circuit Theory and Design*, 1997, 154-156.
15. S. Espejo, R. Domínguez-Castro, G. Liñan and A. Rodríguez-Vázquez, “64 × 64 CNN Universal Chip with Analog and Digital I/O”, *IEEE 1998 Int. Conf. On Electronic Circuits and Systems*, 1998
16. Rinn, R., Lorbach, N., Lücking, F., (1999) ,”"First automatic surface inspection for hot strip mills",Advance Steel, 1999,
17. Ceracki, P., Lücking, F., Reizig, H.-J. (1999),"On-line defect recognition in hot strip by automatic surface inspection", *Stahl und Eisen* 119, No.4, pp. 77-81.
18. Kaiser, S. (1999),”HTS-2 technology - a synonym for inspection of steel strip", QCI Newsletter 2, 1999, pp. 6-7.
19. McManus, George J. (1999), "Automatic Surface Inspection of Sheet", AISE Stee Technology
20. F.Torres, J.M.Sebastián, R.Aracil, L.M.Jiménez, O.Reinoso , “Automated Real-Time Visual Inspection System for High-Resolution Superimposed Printing". Imaging and Vision Computing, vol 16, Issue 12-13, pp. 947-958, 1998.
21. L.M.Jiménez, F.Torres., "Detection of Defects in Aluminium Surface through Computer Vision Techniques" Proc. of QCAV' 97, vol. 1, pp. 381-386, 1997.
22. J. Beyerer, “Model-based analysis of groove textures with applications to automated inspection of machined surfaces.”, Measurement 15, pp. 189-199, 1995.
23. J. Beyerer and F. Puente León, “Suppression of inhomogeneities in images of textured surfaces.”, Optical Engineering, Vol. 36, No. 1, pp. 85-93, 1997.