Detection of Symmetry Axis by a CNN-based Algorithm

G. COSTANTINI[1,2], D. CASALI[1]
[1]Department of Electronic Engineering
University of Rome "Tor Vergata"
Via del Politecnico, 1 – 00133 ROMA
ITALY
[2]Institute of acoustics "O. M. Corbino"
Via del Fosso del Cavaliere, 100 – 00133 ROMA
ITALY

Abstract: - In this paper a method for symmetry axis detection in binary images is presented. The method exploits the nonlinear dynamic behavior of Cellular Neural Networks (CNNs), in particular the propagation of bipolar waves. The image is represented in polar form, transforming the symmetry with respect to an arbitrarily oriented axis in a vertical symmetry: the position of the vertical axis corresponds to the angle of the original symmetry axis. The parallel CNN architecture is useful to speed up the computation, because of the high computational cost of the task.
The proposed algorithm is tested on a real image with good results.

Key-Words: - Cellular Neural Networks, symmetry.

1 Introduction

Symmetry axis detection in images is a task with high computational cost, so a simple method which is capable to speed up the computation, and robust to moderate asymmetries, could be very appreciable. An architecture suited for parallel VLSI image processing is represented by Cellular Neural Networks (CNNs) [1-3].

In this paper, we present a hybrid method for symmetry axis detection in binary images; it is based both on conventional digital processing and analog nonlinear processing using CNNs.

In a first step, the image is represented in polar form, transforming the symmetry with respect to an arbitrarily oriented axis in a vertical symmetry. The position of the vertical axis corresponds to the angle of the original symmetry axis.

In a second step, some CNN templates are used in sequence in order to detect the position of this vertical axis. In particular the proposed CNN processing exploits the propagation of bipolar waves, introduced by Petras and Roska in [4].

2 Overview of Cellular neural Networks

We e assume a CNN model defined on a rectangular (MxN) cell grid. $C(i,j)$ denotes the cell at the intersection of row i and column j. $N_r(i,j)$ denotes the neighborhood of cell $C(i,j)$, with radius r. The state equation of the CNN is the following [1]:

$$\frac{dx_{ij}}{dt} = -x_{ij} + \sum_{\substack{C(k,\ell)\\ \in N_r(i,j)}} \mathrm{A}_{ij,k\ell}\, f(x_{k\ell}) + \sum_{\substack{C(k,\ell)\\ \in N_r(i,j)}} \mathrm{B}_{ij,k\ell}\, u_{k\ell}$$

where x_{ij} is the state of cell $C(i,j)$. $\mathrm{A}_{ij,k\ell}$ and $\mathrm{B}_{ij,k\ell}$ denote the connection weight respectively from output and input of cell $C(k,l)$ to cell $C(i,j)$. The transfer function $f(\cdot)$ is the usual piecewise-linear function:

$f(x) = 0.5.$

3 Preprocessing

Let consider a binary image with a black object on a white background; assume that the object has a symmetry axis with an arbitrary orientation. The symmetry axis passes through the center of gravity, which is easily computed (the image is binary), so we concentrate only on the computation of its angle (Fig. 1). To extract the angle it is convenient to represent the image in polar coordinates, taking as origin the center of gravity. Transformation to polar coordinates can be carried out by a digital processor using the well-known conversion formulas; in fact, it is a pixel-wise computation which is easily parallelisable. If we assign the angle to the variable j (columns) and the distance from the center to the variable i (rows), we obtain a picture where all original symmetries become vertical symmetries; an example is shown in Fig. 2, where the arrow shows the pixel corresponding to the angle of the symmetry axis (clockwise angle orientation is considered).

In conclusion, the problem is simplified since it reduces to detect a vertical symmetry axis. The required number of rows depends on the size of the image and the position of the center of gravity: in the worst case, m is at most $\sqrt{r^2 + c^2}$, where r and c are the number of rows and columns of the original image, respectively. The number of columns, n, depends on the desired angular resolution: the resolution will be $360/n$ degrees. All the pixels of the new image without a correspondence in the original image can be filled with –1 (white). A similar approach has been adopted in [5]: the longitudinal motion stereo problem is transformed into a static stereo vision problem using polar representation of the images, and then solved using CNNs.

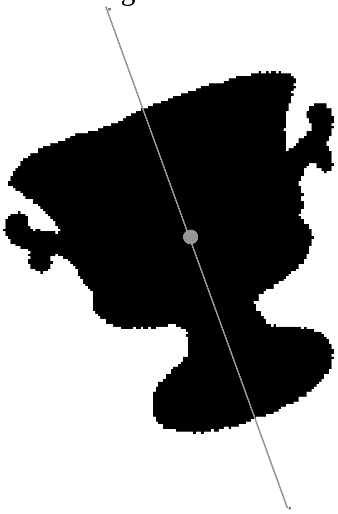

Figure 1. Original image with 200x200 pixels. The axis of symmetry and the centre of gravity are shown.

Figure 2. Image in Fig. 1 converted to polar coordinates with 1° per horizontal pixel.

Resolution is 149x360 pixels. The arrow corresponds to the angle of the symmetry axis (70° clockwise).

2 CNN processing

After the preprocessing phase, we have only images with vertical symmetry axis. Let consider the binary image mapped on a rectangular CNN with m rows and n columns.

The cell input u_{ij}, $i = 1,\ldots, m$, $j = 1,\ldots, n$, represents the pixel value (-1: white; +1: black).

To detect a vertical symmetry axis, the processing steps described in Fig. 3 can be used.

First we apply the following template, called wave:

$$\mathbf{A} = \begin{pmatrix} 0.3 & 0.3 & 0.3 \\ 0.3 & 0.8 & 0.3 \\ 0.3 & 0.3 & 0.3 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} \quad I = 0$$

It is a slight modification of the template proposed in [4] to show the phenomenon of bipolar waves propagation in CNNs. The difference consists in the B template, here used as a vertical edge detector (in [4] only the central element of B is non-zero and set to 1). The state equation of cell c(i,j) becomes:

$$\frac{dx_{ij}}{dt} = -x_{ij} + \sum_{\substack{C(k,\ell) \\ \in N_r(i,j)}} A_{ij,k\ell}\, f(x_{k\ell}) + u_{i,j-1} - u_{i,j+1}$$

The difference $d(i,j) = u_{i,j-1} - u_{i,j+1}$ will be non-zero only if the pixel (i, j) lies on the boundary between black and white regions. This difference can be positive or negative depending on the type of transition: $-1 \rightarrow +1$ or $+1 \rightarrow -1$. Using this template, with zero initial conditions, the network will have the following time-evolution:

1) At the beginning, cells with input difference $d(i,j) \neq 0$ evolve toward saturation, while the other cells remain in the zero state;

2) The cells with saturated output drive to saturation the neighbouring cells with $d(i,j) = 0$;

3) The saturation regions will expand until all the cells have an output equal to ±1.

If there is a vertical symmetry axis, the boundary pixels, from which the expansion starts, will be equidistant from it. Since the black regions expand at the same velocity of white regions, these regions will meet along the symmetry axis, which extends from the top to the bottom of the image. As can be seen in Fig.4(a), in the steady-state output image there can be false boundaries between black and white regions, due to non perfect symmetry in the original image. Nevertheless, experimental results show that if a boundary is almost vertical and extends on the whole image, it represents a symmetry axis with high probability.

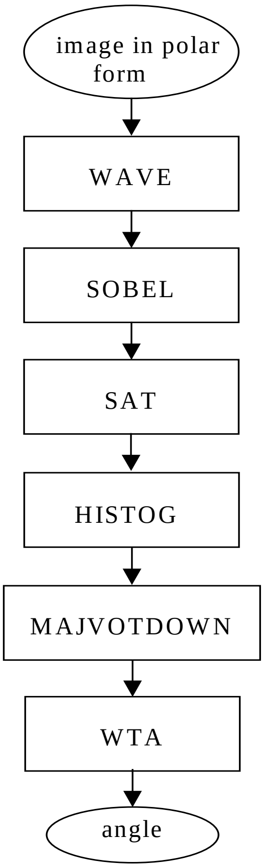

Figure 3. Summary of the proposed CNN processing.

If we know a priori that there is one symmetry axis and we want to know its position, we can look for the longest and "more vertical" edge, and this will give us the position of the axis, also in presence of imperfect symmetry as in Fig. 1. To extract the axis, we perform an edge detection on the output of wave, giving a binary image $\mathfrak{I}$ where the boundaries are black and all the other pixels are white. This can be obtained using the following two templates in sequence ($\mathbf{A}$ = 0, I = 0 in both cases):

$$\mathbf{B}_{\mathbf{SOBEL}}=\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad \mathbf{B}_{\mathbf{SAT}}=\begin{pmatrix} 0 & 0 & 0 \\ 0 & g(.) & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Template $\mathbf{B}_{\mathbf{SOBEL}}$ is a CNN version of the Sobel method [6]; template $\mathbf{B}_{\mathbf{SAT}}$ is non linear with $g(u)$ = 1 if $|u| \geq 1$, $g(u)$=-1 otherwise. The effect of these two templates is shown in Fig.4(b): the longest, almost continuous, vertical line corresponds to the symmetry axis; it is at column 70, giving an angle of 70° clockwise.

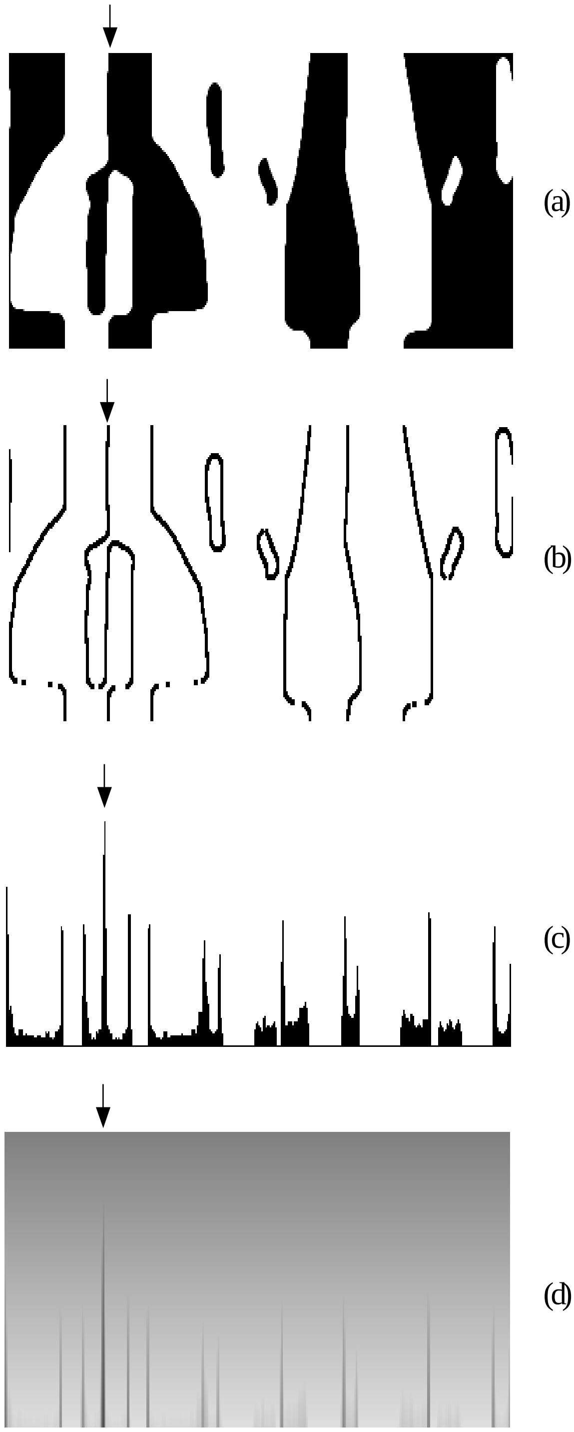

Figure 4. Output image after application of the CNN templates (the arrow corresponds to the angle of the symmetry axis):
(a) Result of WAVE template
(b) Result of SOBEL and SAT
(c) Result of HISTOG
(d) Result of MAJVOTDOWN.

Hence, we need to look for long vertical lines: we should find the line which contain more black pixel. In order to do this, we first move all pixels downward, by means of the non linear template

HISTOG, which generates a one-dimensional histogram of a black-and-white image [7]:

$$\mathbf{A} = \begin{pmatrix} 0 & b & 0 \\ 0 & 1 & 0 \\ 0 & a & 0 \end{pmatrix} \qquad \mathbf{B} = 0 \qquad I = 0 \qquad (1)$$

With:
$a = -3$ if $y_{ij}-y_{kl} > 1.5$
$a = 0$ elsewhere
$b = 3$ if $y_{ij}-y_{kl} < -1.5$
$b = 0$ elsewhere
Result is shown in Fig 4(c).
Finally, we apply the MAJVOTTAKER

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad I = 0 \qquad (2)$$

This is a variation of the first template of the sequence called “majority vote taker” [7]. The purpose of these operations is to find long vertical lines. Using template (2) we obtain an accumulation of pixel values along each column toward the bottom of the image (Fig. 4(d)). At the steady-state, the output $y(m,j)$ of cell $C(m,j)$ will be proportional to the number of black pixels in column j (Fig. 4(d)). Finally we take the pixel j corresponding to the largest value $s(j)$. In Fig. 5 the values $s(j)$ vs. j are plotted; the maximum corresponds to 70°. The largest value can be automatically extracted using a Winner Take All (WTA) network, realizable with a CNN [8]. It is worth noting that in the polar representation of the image we will always find a pair of angles, differing by 180°, corresponding to the symmetry axis of the original image. This redundancy can be exploited to make the detection more robust to small asymmetries.

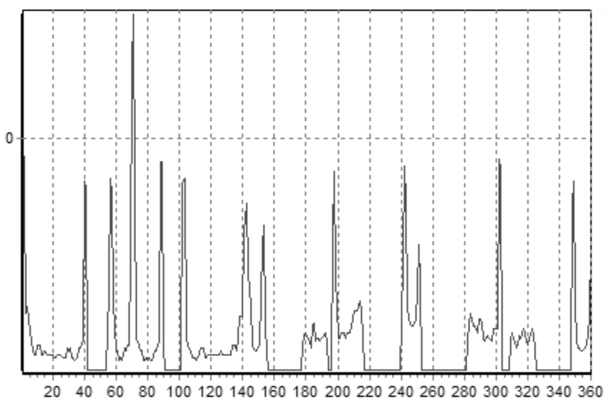

Figure 5. Values of cells at the last row of image in Fig. 4(d). The position of the maximum value corresponds to the angle of the symmetry axis (70°).

4 Conclusion

We presented a method for symmetry axis detection in binary images. Symmetry axis detection is a task with high computational cost, so our method exploits the nonlinear dynamic behavior of Cellular Neural Networks (CNNs), in particular the propagation of bipolar waves to reduce the computation time.

The method is performed in two steps.

In a first step, the image is represented in polar form, transforming the symmetry with respect to an arbitrarily oriented axis in a vertical symmetry. The position of the vertical axis corresponds to the angle of the original symmetry axis.

In a second step, some CNN templates are used in sequence in order to detect the position of this vertical axis. In particular the proposed CNN processing exploits the propagation of bipolar waves.

The good results obtained from tests executed on real images show the effectiveness of the method.

References:

[1] L.O. Chua, L. Yang, “Cellular neural networks: Theory”, *IEEE Transaction on Circuits and Systems*, vol. CAS-35, October 1988, pp. 1257-1272

[2] L.O. Chua, L. Yang, “Cellular neural networks: applications”, *IEEE Transaction on Circuits and Systems*, 1988, vol. 35, pp. 1273-1290

[3] *International Journal of Circuit Theory and Applications*, Special issue on CNN Technology, vol. 30, pp. 87-393, 2002

[4] I. Petras, T. Roska, “Application of direction constrained and bipolar waves for pattern recognition”, *Proc. of 6th IEEE Int. Workshop Cellular Neural Networks and their Applications, CNNA 2000*, pp. 3-8, 2000

[5] S. Taraglio, A. Zanela, A. Pellecchia, “The longitudinal motion stereo problem: a CNN approach”, *Proc. of 6th IEEE Int. Workshop Cellular Neural Networks and their Applications, CNNA 2000*, pp. 375-379, 2000

[6] W.K. Pratt, *Digital Image Processing*, Second Edition, Wiley, 1991

[7] *CNN software library*, Neuromorphic Information Technology Lab., Budapest, 2001, http://lab.analogic.sztaki.hu

[8] G. Seiler, J.A. Nossek, "Winner-take-all cellular neural networks", *IEEE Transaction on Circuits and Systems -Part II*, vol. 40, March 1993, pp. 184-190.