Gabor filter approximation through a Neural Network applied to fingerprint images

Airam Carlos Pais Barreto Marques
Biologica Sistemas Ltda, Av. Almirante Barroso, 91 Sala 814 Centro, Rio de Janeiro, RJ, Brazil
acpairam@gmail.com

Antonio Carlos Gay Thomé
NCE/UFRJ – Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brazil
thome@nce.ufrj.br

***Abstract*— Fingerprints systems are used for access control and forensic applications to confirm or discover the identity of a person. With the emergence of digital computers, fingerprint images could be enhanced using modern digital image processing filters. Widely known, the Gabor filter greatly enhances the fingerprint image quality but it is associated with a high computational cost. This paper describes the use of a neural network trained to behave like a Gabor filter with a lower computational cost.**

Keywords— Gabor's filter approximation, Neural Networks, Fingerprints detection

I. INTRODUCTION

Fingerprints, by its uniqueness properties, have been one of the most used techniques for personal identification as well as for criminal investigations since the beginning of the 20th century.

The fingerprints are formed by furrows on the skin that form ridges in the contact areas, such as the touching areas of the fingers, the palms of the hands and soles of feet. These ridges are known as papillary ridges. The quality of a fingerprint image is generally enhanced by the usage of modern image processing filters. A widely known type of such a filter is the Gabor filter, which was firstly used for the treatment of fingerprint images by Hong et al. [1].

Despite the great improvement on the quality of an image that is possible to get using the Gabor's filter, there is an important drawback related to its high computational cost. In this paper we propose an approach that uses a neural network specially trained to behave like a Gabor's filter. The network is very simple, uses a small number of neurons and presents a lower computational cost compared to the conventional implementation of the Gabor's filter.

II. FINGERPRINT FUNDAMENTALS

In this section we present a brief overview on fingerprints and some of the computational methods used for image segmentation and minutiae extraction.

A. *Fingerprints*

Fingerprints are formed by furrows on the skin that form ridges in the contact areas, such as the touching areas of the fingers, the palm of the hands and soles of feet. These ridges are known as papillary ridges. Figure 1 shows a fingerprint image of a right index finger that was taken with ink over a small piece of paper.

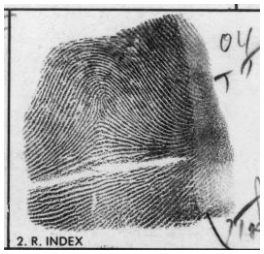

Fig. 1 – Image of a fingerprint that was collected with ink over a piece of paper

Fingerprint ridges have discontinuities, such as an endpoint or a bifurcation. These discontinuities are known as minutiae and were firstly used by Sir Francis Galton in 1892 [2]. As shown in figure 2, each minutia is related to its relative position and direction inside the fingerprint image.

A set of minutiae of a fingerprint identifies a person uniquely, which means that two different fingerprints, with very high probability, have different sets of minutiae.

Fig. 2 – Types of minutiae. (a) ridge end; (b) ridge bifurcation. The arrow shows the minutiae direction

B. *Automatic minutiae extraction*

With the evolution of the computers, the development of computational methods for feature extraction, called codifiers, and feature comparison, called matchers, became feasible. A codifier must extract features that can identify the fingerprint uniquely when used by the matcher, which compares two sets of features, and tells whether or not they were extracted from the same person.

Since the minutiae present a high probabilistic property of uniqueness, and the knowledge about their usage is well established, most of the codifiers are minutiae based, but other features can also be used for identification purposes, such as the pores present on the skin. A codification process generally covers the following phases:

Preprocessing: that consists on filtering the image in order to enhance the contrast of a potential feature or suppress noise;

Region of interest detection: that consists of detecting and extracting the regions containing fragments of the fingerprint. This phase is known as segmentation;

Feature Extraction: that consists of detecting and extracting features from the fragments of the fingerprint, such as minutiae, to be used by the classifier (matcher);

Post processing: that consists on applying validation strategies in order to eliminate or correct those wrongly extracted features, such as false minutiae.

This work focuses on the pre-processing phase, where we propose the use of a neural network to emulate the conventional implementation of the Gabor filter. The neural network implementation shows a small computational cost in terms of summations and multiplications. The other phases of the process are not subject of this paper, but for those interested, automatic methods for minutiae extraction can be found in [2], [3], [4], [5] and [6].

III. GABOR FILTER

The Gabor's filter is used in different contexts, such as texture segmentation, image compression and image enhancement. In Hong et al. [1], this filter was used to improve the quality of ridges and valleys in fingerprint images. Equation 1 shows the impulse response of a bi-dimensional Gabor filter.

$$G(x, y, f, \theta) = \exp\left(-0.5\left(\frac{x_r^2}{gx^2} + \frac{y_r^2}{gy^2}\right)\right) \cdot \cos(2\pi \cdot f \cdot y_r) \qquad (1)$$

where

$$x_r = x \cdot \cos(\theta) - y \cdot sen(\theta)$$
$$y_r = y \cdot \cos(\theta) + x \cdot sen(\theta)$$

From equation 1, we notice that the Gabor response is the product of a Gaussian with a cosine function. The Gaussian has standard deviations directly proportional to gx and gy, as can be seen from the normal distribution formula in Hoel et al. [7].

The variables x_r and y_r represent the same variables x and y rotated by an angle θ in counter-clock-wise fashion. The cosine function has frequency f and oscillates along yr. In figure 3 the impulse response of a Gabor's filter with gx = gy = 4, f = 0.1 and direction θ = 0 (oscillates along axis y) is shown.

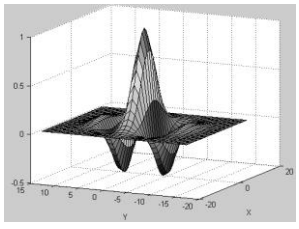

Fig. 3 – Gabor filter impulse response, using gx = gy = 4, f = 0.1 and θ = 0

The frequency response of the Gabor's filter is the convolution of the frequency responses of the Gaussian and cosine functions. The Gaussian function is a low-pass filter and the frequency response of the cosine function corresponds to two impulses symmetrically positioned regarding the origin, with a distance of f from the origin and direction orthogonal to θ. Therefore, the frequency response of the Gabor's filter will be a pass-band filter with the pass-band maximums positioned at a distance f from the origin with an angle orthogonal to the direction θ. Figure 4 shows the frequency response of the Gabor's filter with gx = gy = 4, f = 0.1 and θ = 0.

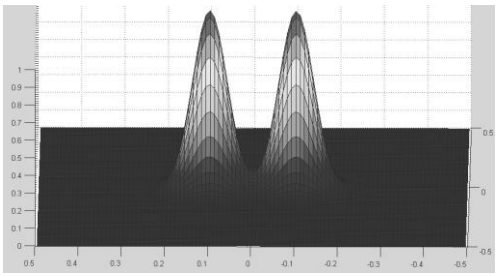

Fig. 4 – Frequency response of a Gabor filter using gx = gy = 4, f = 0.1 and θ = 0

Thus, the Gabor's filter acts improving the contrast of the sinusoids whose frequencies are around f and whose oscillation directions are approximately orthogonal to θ. Besides that, the filter also suppresses the noises around these sinusoids.

A small fingerprint fragment resembles a bi-dimensional sinusoid function, as observed in reference [3] and seen in figure 5.

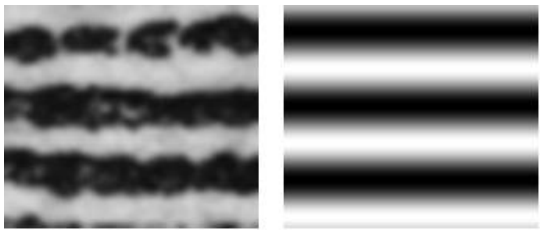

Fig. 5 – The left picture is a small fingerprint fragment. Observe that it resembles a sinusoid like the picture on the right

Because of this similarity the Gabor's filter can be used to improve the quality of fingerprint ridges. The main problem is to select the correct values for the filter's parameters, namely, gx, gy, f and θ.

Setting the value of gx and gy to 4, generally yields good results. Unfortunately, the other two parameters, namely f and θ, both of crucial importance to assure the quality of the filtering, are much harder to determine.

The parameter f must be chosen as the frequency of the sinusoid that best resembles the current fingerprint fragment. In practice this value can be computed through equation 2.

$$f = \frac{1}{2 \cdot b} \qquad (2)$$

where b represents the ridge width given in pixels.

Parameter θ represents the direction of the ridges flow, meaning that the sinusoid oscillates along the direction orthogonal to θ.

The fingerprint fragment shown in figure 5 has ridge width of approximately 7 pixels and ridge flow direction close to 0. Using equation 2 with b = 7 (resulting in f ≅ 0.071), making θ = 0 and gx = gy = 4, a Gabor's filter that can be applied to the fingerprint fragment of figure 4 is obtained. The result of this process is shown in figure 6. Observe how the borders of the ridges were smoothed and the contrast between ridge and background was improved.

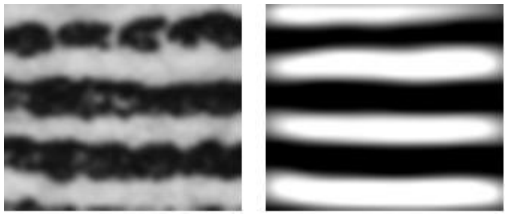

Fig. 6 – The left picture is the same shown in figure 5. The right picture is the result obtained after applying the Gabor filter on the left picture

In reference [1], the authors proposed a method where the image was divided into small non-superposed blocks and for each of them, the local ridge direction and width is estimated. After this, the image is scanned pixel by pixel and a convolution mask computed by equation 1 is applied, using the local ridge direction and width, estimated in the first step. In order to execute this second step efficiently, the authors proposed a pre-computed bank of Gabor's filters (bank of convolution masks) with different directions and widths rather than to compute the mask for each pixel.

In spite of the contrast improvement property of the Gabor's filter, the authors of reference [1] propose the use of a contrast enhancement filter before the use of the Gabor filter.

Figures 7 and 8 show the result of the application of a Gabor's filter using the method proposed in reference [1], over two gray scale 500 dpi resolution fingerprint images. Before applying the Gabor's filter, the images' contrasts were improved using local histogram equalization through an algorithm similar to the equalization presented in Gonzalez and Woods [8].

Fig. 7 – The left picture is a fingerprint fragment. The picture on the right is the result of the application of the method proposed in Hong, Wan and Jain [1]

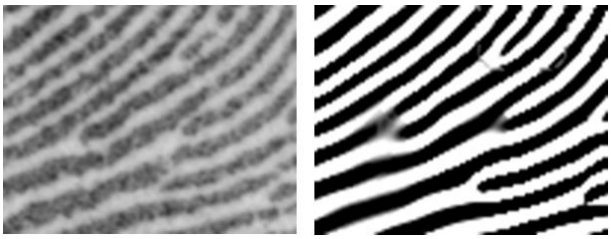

Fig. 8 – Another example of the application of the method proposed in Hong, Wan and Jain [1]

IV. Gabor Approximation Through a Neural Network

The previous section introduced the Gabor's filter, which can greatly enhance the quality of a fingerprint image. Nevertheless, the Gabor's filter shows two main drawbacks: its high computational cost and the need to compute the correct ridge width, what is not an easy task mainly within the context of noisy images.

In this section we propose a method that uses a neural network trained to behave like a Gabor filter. This approach is based on the idea initially proposed by Daugman [9] in which the behavior of the Gabor filter is approximated through the use of neural networks. Nevertheless, the strategies used to build the neural models are different from the strategy proposed in this work and, besides that, in that work [9] the Gabor filter is used in the context of image analyses and compression, while in this work the filter is used to improve the contrast and quality of fingerprint ridges.

The images referenced in this paper are regarded as being a matrix of integers varying from 0 to 255, representing, respectively, the darkest gray (black) and the brightest gray (white), with resolution of 500 dpi.

A. Training

The database used to train the neural network is composed of multiple sinusoid images created artificially, varying the phase and period, and introducing a pseudo-random noise. The sinusoids are bi-dimensional functions and oscillate along the y axis. The way the phase, period and noise are changed to create the sinusoid images is shown below:

- The phase varies within the interval [0, 2π), sampled in 40 uniforms intervals;

- The period assumes integer values varying within the interval [6, 20] assuming, therefore, 15 possible values.

- After the creation of the sinusoid image for a certain phase and period, the pseudo-random noise consists of the addition to each pixel of the image a value that oscillates randomly within the interval [-L, L]. The values that L can assume varies within the interval [0, 96] divided in 7 uniform intervals. It is worthy to say that when L is zero it means that no noise is added to the image.

The values used in the intervals described above were obtained empirically, through the training of several neural networks using different values. After the trainings were finished, the values that corresponded to the best network were retained.

As the phase assumes 40 possible values, the period 15 and the noise 7, the training database consists of 40x15x7=4200 images. Figure 9 shows 3 sinusoids with different values of phase, noise and period. Each image has dimension of 13x13 pixels to be equal to the window that is used to segment the fingerprint image.

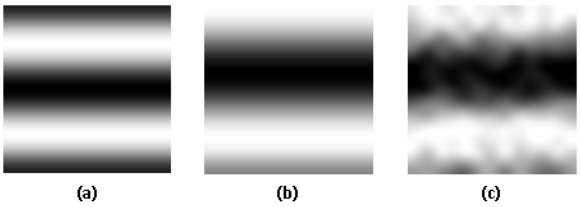

Fig. 9 – (a) A sinusoid with period of 7 pixels, phase 0 and without noise. (b) A sinusoid without noise, with phase of 45^0 and period 10. (c) A sinusoid equal to (b) with noise oscillating between -32 and 32

The input vector of the network is not composed by all pixels of the image (169); it consists of a vector of dimension 13, each one containing the average of the gray levels of each line of the 13 x 13 image.

The desired outputs of the network (training targets) are generated by the Gabor's filter applied to the same image, using the correct parameter f, since the period of each image is known, and orientation θ = 0. The target for the neural network is taken as the Gabor result for the window central pixel.

The neural network model chosen to perform the filtering is the MLP, with only 3 neurons on the hidden layer and 1 neuron on the output layer. The propagation function chosen for all the neurons is the hyperbolic tangent.

There are two considerations to highlight at this point:

The output of the Gabor's filter convolution is another image whose values may not be restricted to the interval [0, 255], so if the output is lower than 0 or greater than 255 it is truncated to 0 or 255, respectively. But for the training, the image values are normalized between –1 and 1, whereas –1 corresponds to 0 and 1 to 255;

Each position of the input vector is divided by 255 to assure that the input values vary in the interval [0, 1].

The network was trained using the RProp (**R**esilient back **Prop**agation) algorithm [10] and [11], with goal 0.001. The training reached a mean squared error of 0.0018, failing to achieve the goal, in 350 training epochs.

B. Filter implementation

After the network training, the filter can be implemented as following:

1st step: Apply the equalization to the input image in order to improve its contrast.

2nd step: Estimate the fingerprint ridges orientation dividing the input image in several blocks of 32x32 pixels, with a superposition of 16 pixels. The estimation is performed using the image gradient directions as proposed in reference [1].

3rd step: For each pixel of the image, repeat the steps 4 and 5.

4th step: Center a window of 13x13 pixels over the current pixel. Rotate the window by the local orientation estimated in step 2. Extract the gray level average for each line of the rotated window, obtaining the input vector with 13 positions. Divide each position by 255.

5th step: Use the vector obtained in the previous step as the input of the network and computes its output. Map the output of the neuron, which ranges from -1 to 1, into the interval [0, 255].

In order to execute the 4th step efficiently, a bank of pre-computed rotations for 8 different angles is used. Then, for each pixel, the pre-computed rotation closest to the estimated orientation is used.

C. Experiments and results

Figure 10 shows an image that was filtered using the proposed method in the previous section and compares it with the images generated by a Gabor's filter with mask of 13x13 and 17x17 pixels.

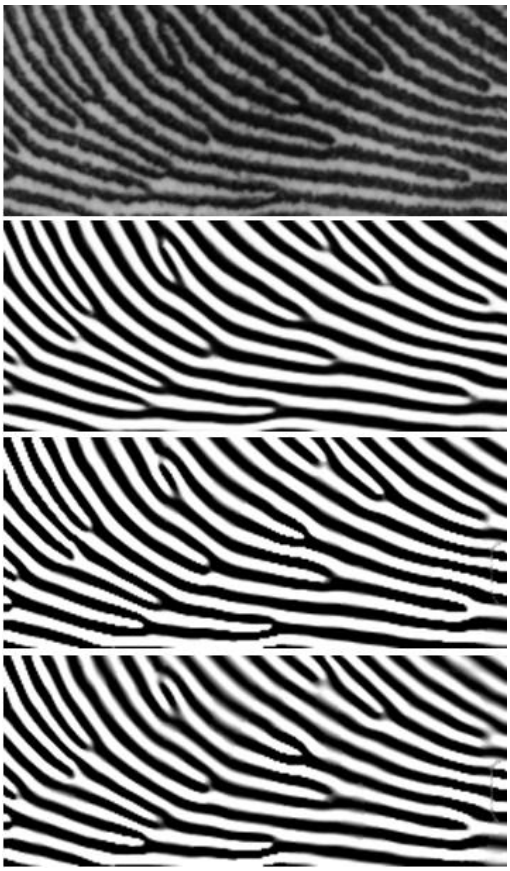

Fig. 10 – From top to bottom: original image; image filtered with the proposed method; image filtered with a Gabor filter with a mask of 17x17 pixels; image filtered with a Gabor filter with a mask of 13x13 pixels.

From the images of figure 10, it can be observed that, in spite of the similarities between them, the image obtained through the proposed method has a quality slightly higher than the image obtained through the Gabor of 13x13 pixels and it is very close to the one filtered with the Gabor of 17x17 pixels.

The filters were tested in an AthlonXP 2000+ with a clock of 1.7 GHz, using images obtained from NIST Special Database 27 [12]. The processing times for each filter over an image with 800x768 pixels are exhibited in table 1.

TABLE 1 – PROCESSING TIME COMPARISON BETWEEN THE PROPOSED METHOD (NEURAL) AND THE GABOR FILTERS OF 13X13 AND 17X17 PIXELS

Method	**Processing time**	**Percentage processing time compared with the proposed method time**
Neural	2.08 seconds	100%
Gabor of 13x13 pixels	2.53 seconds	122%
Gabor of 17x17 pixels	3.64 seconds	17l%

Figures from 11 to 14 show other images filtered with the proposed method.

Fig. 11 – A fingerprint image and the respective image obtained through the Gabor filtering simulation using the proposed approach

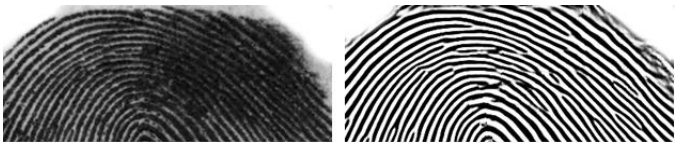

Figure 12 – A fingerprint image and the respective image obtained through the Gabor filtering simulation using the proposed approach

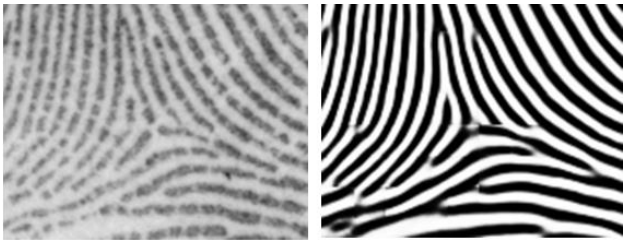

Figure 13 – A fingerprint image and the respective image obtained through the Gabor filtering simulation using the proposed approach

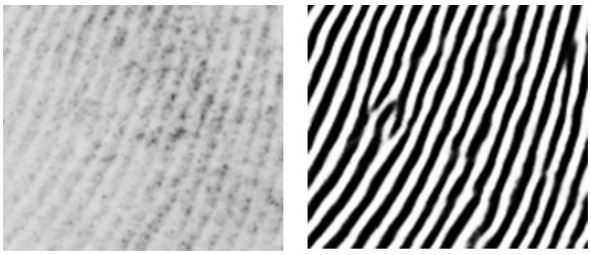

Figure 14 – A fingerprint image and the respective image obtained through the Gabor filtering simulation using the proposed approach

V. COMPLEXITY ANALYSIS

In the previous section, the absolute time performance of the traditional and the neural Gabor filter were compared. Despite the importance of this comparison, it is not the best way to compare the performance of computational costs, because it heavily depends on the implementation algorithm and the computer used for the comparison.

So, it seems to be much more interesting to compare them with respect to the number of multiplications and summations consumed by each method.

Let NMTG be the Number of Multiplications used by the Traditional Gabor implemented via convolution masks. In the same way, NSTG is the number of sums used by the same implementation.

Let NMNG and NSNG be, respectively, the number of multiplications and summations used by the Neural Gabor implementation.

In order to simplify the computation of these values, it is assumed that the image to be filtered and the convolution mask are square and their number of rows and columns are respectively, n and k.

As the traditional filter is implemented via convolutions, the number of multiplications and summations are equal and their value is given by equation 3.

$$NMTG = NSTG = n^2 \cdot k^2 \qquad (3)$$

In the neural implementation, both NMNG and NSNG depend, additionally, on the number of neurons m used on the hidden layer. These values are given by equations 4 and 5.

$$NMNG = n^2 \cdot m \cdot (k+1) \qquad (4)$$

$$NSNG = n^2 \cdot \left((k+1) \cdot m + m + 1 + k^2\right) \qquad (5)$$

The ratio between NMTG and NMNG, given by equation 6, shows the number of multiplications saved by the neural Gabor approach, using k = 13 and m = 3.

$$TM = \frac{NMTG}{NMNG} = \frac{n^2 \cdot k^2}{n^2 \cdot m \cdot (k+1)} = \frac{k^2}{m \cdot (k+1)} =$$

$$= \frac{13^2}{3 \cdot (13+1)} = \frac{169}{42} = 4.02 \qquad (6)$$

Thus, from equation 6 we can conclude that the neural approach strongly reduces the number of multiplication operations.

In the same way, equation 7 shows the ratio between NSTG and NSNG.

$$TS = \frac{NSTG}{NSNG} = \frac{n^2 \cdot k^2}{n^2 \cdot \left((k+1) \cdot m + m + 1 + k^2\right)} =$$

$$= \frac{k^2}{(k+1) \cdot m + m + 1 + k^2} = \frac{169}{14 \cdot 3 + 14 + 169} =$$

$$= \frac{169}{225} = 0.75 \qquad (7)$$

From equation 7, it can be observed that the number of summations are increased in the neural approach, resulting in 1/0.75 = 1.33 more summations than the traditional method.

VI. Conclusions

The Gabor's filter, a widely known digital filter used to enhance fingerprint image quality, has two main drawbacks: its high computational cost, and the difficult task of correctly estimating the width of fingerprint ridges.

In this paper, a method to solve these two problems is proposed. The proposed approach uses a neural network trained to behave like a Gabor filter. The network is trained using an input vector obtained from a set of artificially created sinusoid images. The input vector does not have explicit information about ridge width.

Analyzing the experiment results presented in the previous section, we note that the network output is very similar to the output provided by the Gabor's filter with a mask of 17x17 pixels. Since the network only receives 13 grey level values extracted from the window of size 13 x 13, it is reasonable to conclude that the network is able to estimate, by itself, the ridge width information.

The response time of the proposed approach is around 22% faster than the one provided by the conventional implementation of the Gabor filter with a matrix of 13x13 pixels and 70% for the one with mask of 17x17 pixels. The neural Gabor also provides good results and requires less amount of memory, since it does not need a pre-computed table of convolution masks.

Although the proposed approach performs 1.33 times more operations of sums when compared to the traditional approach, the number of multiplications is 3.75 times lower.

References

[1] L. Hong, Y. Wan and A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no.8, (1998), 777--789.

[2] Americo Lobo Neto, Extração automática de minúcias de impressões digitais**,** Bachelor monograph (Bachelor in Computer Science) – Universidade Federal Fluminense - Brasil, 2000.

[3] Anil K Jain and Sharath Pankanti, "Fingerprint Classification and Matching"**,** 1997. Accessed in: May 21st 2003. http://www.research.ibm.com/ecvg/pubs/sharat-handbook.pdf.

[4] Dario Maio and Davide Maltoni, "Direct Gray-Scale Minutiae Detection In Fingerprints"**,** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 1, Jan 1997.

[5] V. K. Sagar, D. B. L. Ngo, K. C. K Foo, "Fuzzy Control for Fingerprint Feature Selection"**,** *ACCV'95 Conference*, Vol. 3, pp. 767-771, Dec 5-8 1995.

[6] W. F. Leung *et al*, "Fingerprint Recognition Using Neural Network", *Proc. of the IEEE workshop Neural Network for Signal Processing*, 1991, pp. 226-235.

[7] Paul G. Hoel, Sidney C. Port and Charles J. Stone, *Introdução à teoria da probabilidade*, 1st ed. Interciência, 1978.

[8] Rafael C. Gonzalez and Richard E. Woods, *Processamento de Imagens Digitais*, 1. ed. Edgard Blücher Ltda, 2000.

[9] John Daugman, "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression**",** *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 36, No 7, pp. 1169-1179, 1988.

[10] D. E. Rumelhart, G. E. Hinton and R. J. Willians, "Learning representations by back-propagation errors", Nature, pp. 323:533-536, 1986.

[11] Simon Haykin, *Redes Neurais: princípios e prática***,** Porto Alegre: Bookman, 2nd ed., 2001.

[12] Michael D. Garris, R. Michael McCabe, Nist Special Database 27 – Fingerprint Minutiae From Latent and Matching Tenprint Images, 2000. Available in the following address: (accessed in September of 2003) http://www.nist.gov/srd/PDF%20files/nistsd27.pdf