Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial
ISSN: 1137-3601
revista@aepia.org
Asociación Española para la Inteligencia Artificial
España

Minutiae Detection: An Image Exploring Agent-Based Model

Ariel Unanue, Adriana Zapico

Departamento de Computación
Fac. de Cs. Exactas, Fisico-Químicas y Naturales
Universidad Nacional de Río Cuarto
Enlace Ruta 8 y 36
5800 Río Cuarto Argentina
aunanue@exa.unrc.edu.ar, zapico@dc.exa.unrc.edu.ar

Abstract

Fingerprint Identification is one of the most precise methods to determine a person's identity. For years, this task has been carried out manually, but, due to the great number of fingerprints involved in a comparison, the implementation of an Automated Fingerprint Identification System (A.F.I.S.) emerged as a necessity. A critical process in automated fingerprint identification is minutiae detection, process in which the presence of image noise or filth would generate undesired spikes and breaks that would lead to a considerable number of spurious minutiae. To overcome this problem, before minutiae detection, a usual process applied is a post-processing of the thinned image or pruning step, being the latter not always successful. In this paper, we introduce a new minutiae detection method, without the need of a pruning step, based on image exploring agents. First, reactive agents are used over the thinned fingerprint image. They detect locations of interest that could be minutiae using an efficient coefficient, which we propose, and to our knowledge, it has not been applied before for minutiae detection. Then, several agents run through the image, starting at the locations of interest, to determine whether they are real minutiae or not. Having a more reliable method for minutiae detection would lead into a better performance of the whole recognition system.

Keywords: automated fingerprint identification, image exploring agents, pattern recognition, minutiae detection.

1 Introduction

From the beginning, and with different goals, man has been looking for methods to properly identify people. Unfortunately, documents that might be useful for this purpose, like passports, driver licenses, identity cards, etc., are easily alterable or duplicated. Hence, identification is disturbed, since some people might easily have multiple identities or alias to apply with fraud intention or to avoid justice.

Fingerprint identification is one of the more certain methods [1]. It has been applied since antique civilizations of Orient, where Emperor's fingerprint was a usual sign for certifying state documents. The use of fingerprint was increasing up to resulting in an important technique for personal identification [2].

However, manual identification is too tedious, too much time is needed, and today is nearly impossible of being applied because of the great number of fingerprints involved in a comparison [1]. Because of this, in the '60 it became obvious the necessity of developing and applying an Automated Fingerprint Identification System (A.F.I.S) [1, 3, 4].

Automated Fingerprint Identification is one of the most important biometric's technologies[1] at present. It has extended the use of fingerprint identification not only to forensic applications but to other civil ones like access control, Internet transaction validation, and Automatic Teller Machines (A.T.M.), among others [1, 3, 5, 6].

In the last years, several techniques have been developed, like those based on iris patterns, fingerprints and voice recognition to verify someone's identity. Fingerprint recognition has been used for a long time and it has been imposed like a robust person-identification method for two main reasons. First, in spite of the fact that the fingerprints may suffer little alteration, e.g. scars or burnt, they remain inalterable for life-time from the point of view of identification, and second, for *fingerprints uniqueness property* - as mentioned by Lee and Gaensslen [3] - determined by the characteristics of the ridges and valleys and their relationships.

Automated Fingerprint Identification System involves several processes, being a critical one the proper minutiae detection. This process becomes more difficult because of the presence of image noise or filth that might generate undesired spikes and breaks, resulting in a great number of false minutiae detected. Some authors [7] argued that the application of a pruning process might resolve this problem of false minutiae. However, spurious minutiae still might remain and be confused with true minutiae.

In this work, we introduce a new minutiae detection method, without the need of a pruning step, based on image exploring agents. First, reactive agents are used over the thinned fingerprint image. They detect *locations of interest* that could be minutiae using an efficient coefficient, which we propose, and to our knowledge, it has not been applied before for minutiae detection. Then, several agents run through the image, starting at the locations of interest, to determine whether they are real minutiae or not. This innovation results in an efficient method in terms of time and in terms of the Goodness Index [8]. The use of Image Exploring Agents for minutiae detection is not new [9, 10], but it has been used for minutiae detection from gray-level fingerprint images instead of thinned ones.

The rest of this paper is organized as follows. First, Section 2 presents background information on fingerprints, discussing their characteristics. It also describes the steps involved in a common Automated Fingerprint Identification System. Section 3 summarizes several approaches applied for minutiae detection. Section 4 presents the application of minutiae detection agents. Finally, some preliminary results are shown in Section 5, while some suggestions for future work are summarized in Section 6.

2 Background on Fingerprints

A fingerprint may be seen (Figure 1) as a graphic that follows the course of the ridges and valleys that form it [1]. Ridges are the stick up borders of the skin, while valleys are the spaces between ridges.

As it was mentioned, there exists a biunivoque relation between persons and their fingerprints, being this property based on the characteristics of the ridges, valleys and their relationships. In manual fingerprint identification, these characteristics are called *characteristic points* and they have an inappreciable value for identification, since a fingerprint is totally characterized by them. Five types of

[1] Biometrics is the science of identifying or verifying the identity of a person based on physiological or behavioral characteristics [5].

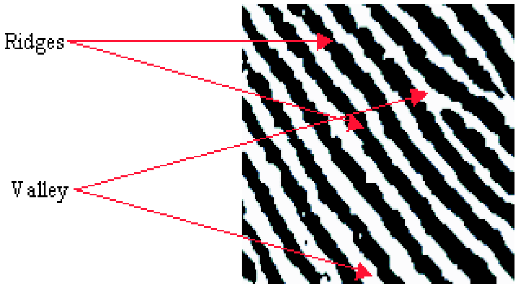

Figure 1: Fingerprint Image

characteristic points have been described in [2], while eighteen different types have been enumerated in [6]. Between those points, Automated Fingerprint Identification only distinguishes two types (Figure 2):

ridge ending: is a point where a ridge ends abruptly,

ridge bifurcation: is a point where a ridge splits into two ridges,

since other characteristic points may be seen as a combination of them [1]. In Automated Fingerprint Identification, these two characteristic points are called *minutiae*.

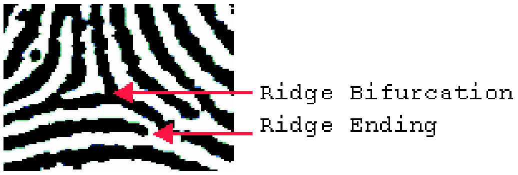

Figure 2: Minutiae

Automated Fingerprint Identification

Minutiae are used in automated fingerprint identification to make possible people's recognition. But, before being able to do recognition, several processes have to be done to the fingerprint. Different authors propose a variety of these processes. Indeed, there exists an ample bibliography about the different alternatives (eg. [1, 7, 11, 12]). Following, one of the most common chain of processes is briefly detailed. It involves:

Image acquisition: from plain scanners (for inked fingerprint) or through live-scanners. Through them, a digital gray-level image of the fingerprint is obtained for being processed.

Image Enhancement: It is usual that during the acquisition step, some noise appears resulting in cutting of the ridges and other undesirable effects. Because of this, a process that improves the quality of the fingerprint is needed in order to make more robust the whole automated identification process.

Thresholding (Binarization): In this step, the image is transformed from a gray-level image into one with only black and white values.

Thinning (Skeletonization): Here, the wide of the ridges is refined up to one pixel. This makes possible the minutiae detection, but it might also generate several spikes and breaks that might be considered as minutiae when in fact they are not.

Post-processing of the Skeleton: In order to decrease the amount of spurious minutiae detected, this stage intends to eliminate the majority of spikes and breaks generated in the previous stages.

Minutiae Detection: This is a crucial stage, in which fingerprint's minutiae are detected, it is desirable to minimize the number of spurious minutiae detected. Minutiae are stored as templates in order to a latter comparison.

Each stage may involve several substages, a detailed description may be seen in [7].

Once these stages are finished, matching is possible to be done. In the matching process two fingerprints are analyzed, or strictly speaking, the templates obtained in the minutiae detection process are compared to determine if both template correspond to the same fingerprint (*match*) or not (*no match*).

3 Minutiae Detection

As it was mentioned, a critical step in automated fingerprint identification is the proper minutiae detection from the fingerprint image [8]. The performance of the whole system relies on it. There exists several methods for minutiae detection, for example, those based on:

Artificial Intelligence: different approaches may be considered, applying:

- Neural Networks for detecting the presence of minutiae in the thinned fingerprint image, e.g. [12].
- Fuzzy Logic and Neural Networks to detect minutiae on the gray-level image, e.g. [11].
- Image Exploring Agents and Reinforcement Learning to minutiae detection from a gray-level fingerprint image, e.g. [9].
- Image Exploring Agents and Genetic Programming to minutiae detection from a gray-level fingerprint image, e.g. [10].

Mask-based: These methods apply a sum to the neighbor pixels. For example, the method suggested by Jain and Hong [1] defines that if a pixel is over a thinned ridge, then, it has a value 1, and 0 otherwise. Let $N_1, \ldots, N_8$ denote the 8 neighbors of a given pixel $N_0 = I(x,y)$ which value is 1, i.e. N_0 is over a ridge. Then, N_0 is a *ridge ending* if

$$\sum_{i=1}^{8} N_i = 1, \tag{1}$$

and a *ridge bifurcation* if

$$\sum_{i=1}^{8} N_i > 2. \tag{2}$$

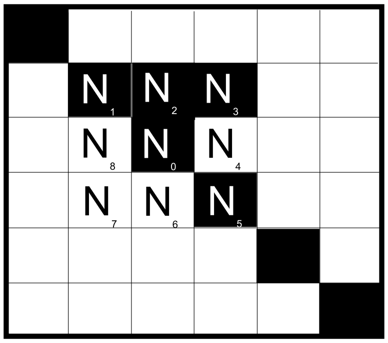

Figure 3: Neighborhood of a pixel N_0=I(x,y)

Otherwise, there is no minutiae in N_0. However, most of the thinning algorithms might leave fingerprint images with ridges portions like is shown in Figure 3. Note that applying this method to N_0 leads to an error. Indeed, $\sum_{i=1}^{8} N_i = 3$, hence, taking into account inequality 2, N_0 results in a spurious ridge bifurcation.

Therefore, it might occur that a (considerable) number of spurious minutiae appear. Usually, to overcome this problem, it is necessary to apply a post-processing method of the thinned fingerprint image or pruning method.

Unfortunately, the most usual pruning methods for image processing (e.g. [13]) are not useful for fingerprint images. Indeed, their behaviors are based on an iterative process on the image, in each iteration this process would eliminate pixels that correspond to certain patterns or masks. The problem is that the number of iterations that eliminates spikes without eliminating ridges as well has to be heuristically determined. This point is difficult, since it might work well with some types of images but very bad with other types. Ratha et al. [7] obtained good results using an adaptive morphological filtering, however, it also might leaves spikes and breaks mixed with “true” minutiae.

Since post-processing methods of the thinned fingerprint image do not always result well, and since iterative processes consume a considerable amount of time, we propose an Image Exploring Agent-Based Model for Minutiae Detection in next Section.

4 Image Exploring Agents for Minutiae Detection

Now, we describe the Image Exploring Agent-Based Model, which is able to detect minutiae over thinned fingerprint images without the need of pruning or another post-processing before detection. Indeed, we apply agents for the minutiae detection, being the agents assigned to a position in the image (its environment) with different goals. Each agent has an internal representation of its environment:

position: a pair of coordinates (x, y) into the image,

orientation: representing the direction in which it is looking.

By sensing the pixels-neighborhood of its current position, the agents can react in different ways (e.g. changing its position by moving along its orientation, changing the orientation, or creating new agents, etc.) in order to achieve its goals.

4.1 Agents' Description

The minutiae detection model over thinned fingerprint images consists of:

- **root agent** (RA)**:** a parent agent.
- Two sons of the RA called:

  ridge ending agent (REA), and

  ridge bifurcation agent (RBA)

  Only one instance of each agent (REA, RBA) for each RA is allowed.
- **Follow Ridge Agent** (FRA)**:** which has the ability of following the track of the ridges based on their local direction. It is possible to have several instances of this agent, but each one should have only REA or RBA agents as a father.

4.2 Behavior of the Agents

4.2.1 Root Agent

This is a reactive agent [14, 15] that runs through the image looking for *locations of interest*, which are possible minutiae. Each time that a location of interest is found, it reacts by sending messages to other agents. As it was mentioned above, one of the main problems in minutiae detection is to distinguish "true" minutiae from spurious ones. Before analyzing the behavior of the agent, let us introduce a characteristic property which allows us to detect minutiae. To our knowledge, it has not been applied before to this process.

Possible Minutiae Property (PM property)

Given a pixel $N_0 = I(x, y)$, and being $N_1, ..., N_8$ its neighbors, we define

$$PM(x, y) = \sum_{i=1}^{7} |N_{i+1} - N_i| + |N_1 - N_8| \quad (3)$$

This property may be used to detect minutiae in thinned fingerprint images. Indeed, if $PM(x, y) = 1$, then, there is a ridge ending in N_0, while, if $PM(x, y) = 6$, then, there is a ridge bifurcation in N_0. Otherwise, there is no minutiae in N_0.

From the agent's point of view, if N_0 is a ridge ending, the agent is placed on a location of interest called *possible ridge ending* (PRE), while if it is a ridge bifurcation, the agent is placed on one called *possible ridge bifurcation* (PRB). Otherwise, the RA agent moves to the following place. Each time that RA is placed on a PRE, it sends a message of *possible ridge ending message* ($PREM$) to the REA indicating that in it current position might exist a ridge ending and the REA needs to check it. Analogously, when the RA is placed on a PRB, a message of a *possible ridge bifurcation message* ($PRBM$) is sent to the RBA.

Observation: the application of the *Possible Minutiae Property* to the example shown in Figure 3 results in $PM(x, y) = 4$, that is, N_0 is not a *spurious ridge bifurcation* nor a *location of interest.*

4.2.2 Follow Ridge Agent

When this agent begins working, its location and orientation are assigned by its father (a REA or a RBA). Then, the agent starts running along the ridge in the orientation it knows, but also sensing the environment to detect a change of the ridge orientation. It has to take into account that a change of orientation can not result into stepping back along the ridge. The agent keeps running along the ridge until the ridge ends or another condition to stop it, e.g. to run along the ridge N steps - being N large enough - or another condition heuristically determined, becomes true.

Then, when this agent stops, it sends a message to its father (a REA or a RBA) so the father would be able to determine if the possible minutiae is true or not. The message includes the number of steps made and if there were orthogonal orientation changes, the points where that changes took place.

4.2.3 Ridge Ending Agent

The goal of this agent is to determine whether a Possible Ridge Ending (PRE) where the Root Agent (RA) was placed is a true ridge ending or not (Figure 4).

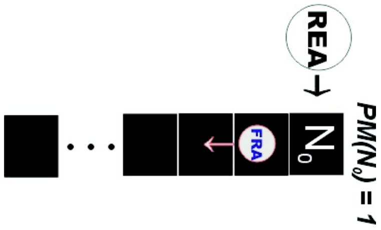

Figure 4: Ridge Ending Agent

Each time a Possible Ridge Ending Message ($PREM$) arrives, the state of this agent can be either busy or idle.

On Busy State the message is posted on a FiFo queue (First in - First Out), so the agent could manage it latter.

On Idle State the agent changes its state to busy and starts its work.

To determinate if PRE is a true minutiae, the Ridge Ending Agent (REA) senses its environment to find out the orientation of the ridge. Then, it sets the position and orientation (internal state) values of the Follow Ridge Agent (FRA) agent that will follow the ridge. Now, the FRA starts its work and the REA waits until a message arrives from the former.

Finally, the REA evaluates the PRE considering the information received in the message:

- if the number of steps given by the FRA is lesser than the constant MIN_RIDGE_LENGTH[2], which represents the minimum ridges length, then the PRE is a spurious ridge ending (isolated ridge).
- otherwise, if the FRA has changed its orientation orthogonally and in that point $PM(x, y) = 6$, the PRE also is a spurious ridge ending (spikes or bridges).
- in other cases, the PRE is a true ridge ending.

After the evaluation, the REA takes another PRE from the queue (if there is one).

4.2.4 Ridge Bifurcation Agent

This agent works similarly to the previous one. But now, the RBA has to determine if a PRB where the RA was placed is a true ridge bifurcation or not (Figure 5).

Each time a $PRBM$ arrives, the state of the agent can be either busy or idle.

On Busy State just like before, the message is posted on a FiFo queue, so the agent could manage it latter.

[2] The value of this constant has been heuristically determined. In our work, MIN_RIDGE_LENGTH is equal to 15

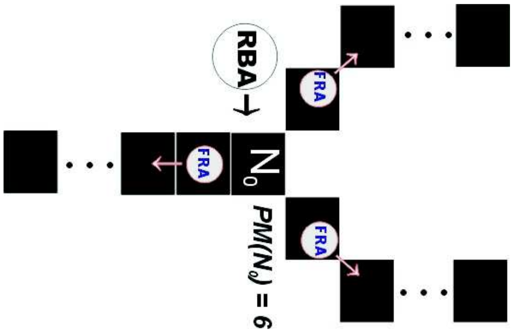

Figure 5: Ridge Bifurcation Agent

On Idle State the agent changes its state to busy and starts its work.

To determine if the Possible Ridge Bifurcation (PRB) is a true minutiae, the agent needs to sense its environment to find out the three ridges that form the bifurcation. Once it finds the ridges, it uses three FRA agents to follow each one of them. Before that, the Ridge Bifurcation Agent (RBA) sets the internal states of each FRA with the same position, being the orientation of each one the respective orientation of each ridge. Then, the three FRA start their work and the RBA waits until the three messages arrive.

Finally, as with the REA, if the number of steps of each one of the three FRA is greater than MIN_RIDGE_LENGTH, the PRB is a true ridge bifurcation, otherwise, it is a spurious one. After the determination of the PRB, the RBA takes another PRB from the queue (if there is one).

Next, a thinned fingerprint image with all the locations of interest detected by the $RootAgent$ is shown in Figure 6, followed by the same fingerprint image, but now only with the true minutiae detected using our method (Figure 7).

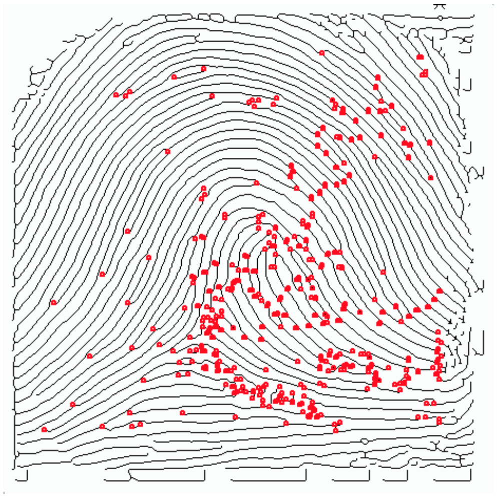

Figure 6: Fingerprint with all the locations of interest detected

5 Preliminary Results

The performance of this proposal has been evaluated using the Goodness Index (G.I.). This G.I. is a usual value for quality determination of a minutiae detection algorithm.
It compares the set of minutiae obtained using an automated minutiae detection process against the set of minutiae obtained by an human expert. It is desirable to have a G.I. value near 1, which means that the number of real minutiae missed and the number of spurious minutiae detected are low (for more details see [7]).

Table 1 shows some preliminary results obtained using ten fingerprints randomly chosen from the N.I.S.T. 9 database [16]. It can be observed that the minimum value is 0.25 while the maximum value is 0.49 and the mean is 0.37. The total process time for the minutiae detection using a PC Pentium II/400 MHz with 64 MB RAM is around 0.3 seconds.

It is not easy to make a direct comparison

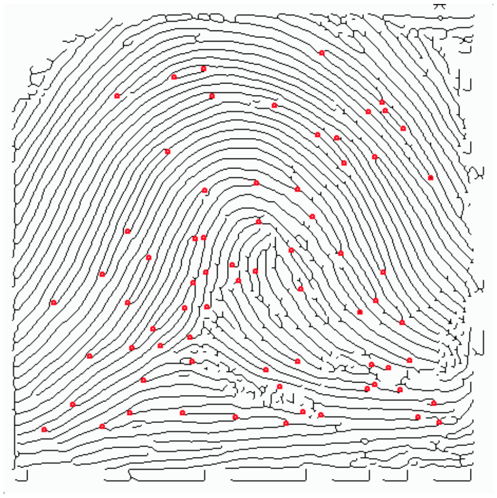

Figure 7: Final result with all the true minutiae detected

with others proposal like Ratha et al.'s method and Jain et al.'s method ([7, 8]). Indeed, two main reasons make this comparison rather difficult. First, they use fingerprint databases whose access is not free. Secondly, Ratha et al.'s work is tested on a SPARC 20 MODEL 30 machine and we have no access to this kind of equipment. Also, Jain et al. do not provide processing time information in their article. Nevertheless, a comparison is still intended, which is summarized in Table 2. In this comparison, the G.I. and Processing Time values are mean ones obtained of each proposal.

6 Concluding Remarks

Before proposing some possible improvements to our work, let us summarize the general minutiae detection process proposed. It begins with the Root Agent exploring the thinned fingerprint image, looking for locations of interest applying the Possible Minutiae Property. Each time that the Root Agent finds one, it sends either a Possible Ridge Ending Message or a Possible Ridge Bifurcation Message to the corresponding receiver. The receiver, the Ridge Ending Agent or the Ridge Bifurcation Agent, processes the message and communicates with the Follow Ridge Agent to determine the veracity of the possible minutiae.

Fingerprint	GI	Processing Time (sec)
1	0.31	0.2
2	0.42	0.2
3	0.27	0.4
4	0.43	0.2
5	0.49	0.3
6	0.36	0.2
7	0.40	0.2
8	0.47	0.4
9	0.25	0.3
10	0.33	0.2

Table 1: Preliminary results of our minutiae detection process

Method	GI	Proc. Time
Ratha et al.'s method	0.24	1.8 sec.
Jain et al.'s method†	0.24	not provided
Jain et al.'s method‡	0.39	not provided
Our method	0.37	0.3 sec.

Table 2: Methods' comparison
‡ and † mean *with* and *without enhancement technique* respectively.

Using this technique, it seems that a good speed process for minutiae detection with acceptable quality might be obtained. Moreover, in order to improve the time spent on minutiae detection without losing quality, it would be possible to have several *RootAgents* running at the same time, each one with different image parts to work on.

While in order to obtain a better Goodness Index value, we are working on a variable value - based on the local length of the ridges - rather than a constant one for the minimum ridge length.

Finally, we are developing a complete Automated Fingerprint Identification System that extends our minutiae detection process.

Acknowledgements

Adriana Zapico acknowledges partial support of Universidad Nacional de Río Cuarto and Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

References

[1] A. K.Jain and L. Hong. *On-line fingerprint verification.* Technical Report MSU-CPS-99-19, Department of Computer Science, Michigan State University, East Lansing, Michigan, March 1999.

[2] S. T. Arriaga. *El Estudio científico de la Dactiloscopía.* Editorial Limusa, 1987.

[3] H.C. Lee and R.E. Gaensslen, editors. *Advances in Fingerprint Technology.* New York: Elsevier, 1991.

[4] Federal Bureau of Invertigation (F.B.I.), http://www.fbi.gov/hq/cjisd/iafis.htm.

[5] N. K. Ratha, A. Senior and R. M. Bolle. *Automated biometrics. Lecture Notes in Computer Science*, 2013:445–455, 2001.

[6] Federal Bureau of Investigation. *The Science of Fingerprints: Classification and Uses.* U. S. Government Printing Office. Washington D. C., 1984.

[7] N. K.Ratha, S. Chen and A. K. Jain. *Adaptive flow orientation based feature extraction in fingerprint images.* Technical Report MSU-CPS-99-17,Department of Computer Science, Michigan State University, East Lansing, Michigan, March 1999.

[8] L. Hong and A. K. Jain. *Fingerprint image enhancement: Algorithm and performance evaluation.* Technical Report MSU-CPS-97-35, Department of Computer Science, Michigan State University, East Lansing, Michigan, October 1997.

[9] A.M. Bazen, M. van Otterlo, S.H. Gerez and M. Poel. *A reinforcement learning agent for minutiae extraction from fingerprints.* In B. Krose, M. de Rijke, G. Schreiber and M. van Someren, editors, *Proceedings of the Belgium-Netherlands Artificial Intelligence Conference (BNAIC'01')*, pages 329–336, 2001.

[10] P.G.M. van der Meulen, H. Schipper, A.M.Bazen and S.H.Gerez. *Pmdgp:a distributed object-oriented genetic programming environment.* In The Netherlands Heijen, editor, *Proc. ASCI Conference*, 2001.

[11] V. K. Sagar and K. J. Beng. *Fingerprint feature extraction using fuzzy logic and neural networks.* Technical report.

[12] W.F. Leung, S.H. Leung, W.H. Lau and A. Luk. *Fingerprint recognition using neural network.* In *IEEE workshop Neural Network for Signal Processing*, pages 226–235, 1991.

[13] J. C. Russ. *The Image Processing HandBook.* CRC Press, 1994.

[14] M. Costa and B. Feij'o. *An architecture for concurrent reactive agents in real-time animation.* In *In Proceedings of SIBGRAPI'96, IX Brazilian Symposium of Computer Graphics and Image Processing*, pages 281–288, 1996.

[15] S. Nolfi. *Power and limits of reactive agents.* (in press) Power and Limits of Reactive Agents. Neurocomputing.

[16] National Institute for Standards and Technology (N.I.S.T.), http://www.nist.gov.