Turk J Elec Engin, VOL.15, NO.1 2007, © TÜBİTAK

Offline Signature Verification Using Graph Matching

Ibrahim S. I. ABUHAIBA
Department of Electrical and Computer Engineering, Islamic University of Gaza,
P. O. Box 1276, Gaza-PALESTINE
e-mail: ihaiba@mail.iugaza.edu

Abstract

In this paper, we present a simple and effective signature verification method that depends only on the raw binary pixel intensities and avoids using complex sets of features. The method looks at the signature verification problem as a graph matching problem.

The method is tested using genuine and forgery signatures produced by five subjects. An equal error rate of 26.7% and 5.6% was achieved for skilled and random forgeries, respectively. A positive property of our algorithm is that the false acceptance rate of random forgeries vanishes at the point of equal false rejection and skilled forgery false acceptance rates. Keeping the normalization size at 32 × 64 pixels makes the verification time in the two seconds range.

Key Words: *Offline Signature Verification, Graph Matching, Thinning, Normalization.*

1. Introduction

The need to ensure that only the right people have authorization to high-security accesses has led to the development of systems for automatic personal verification. Signatures, fingerprints, palm prints, voice, and handwriting have all been used to verify the declared identity of an individual. Among all, signature has a fundamental advantage in that it is the customary way of identifying an individual in daily operations such as automated banking transaction, electronic fund transfers, document analysis, and access control.

A signature verification system must be able to detect forgeries and at the same time reduce rejection of genuine signatures. The signature verification problem can be classified into categories: offline and online. Offline signature verification [1-49] does not use dynamic information that is used extensively in online signature verification systems [20, 22, 50-54]. In this paper, we investigate the problem of offline signature verification.

The problem of offline signature verification has been faced by taking into account three different types of forgeries: random forgeries, produced without knowing either the name of the signer nor the shape of his signature; simple forgeries, produced knowing the name of the signer but without having an example of his signature; and skilled forgeries, produced by people who, looking at an original instance of the signature, attempt to imitate it as closely as possible. It is obvious that the problem of signature verification becomes more and more difficult when passing from random to simple and skilled forgeries, the latter being so difficult a task that even human beings make errors in several cases. In fact, exercises in imitating a signature often allow us to produce forgeries so similar to the originals that discrimination is practically impossible; in many cases, the distinction is complicated even more by the large variability introduced by some signers when writing their own signatures.

Many research works on signature verification have been reported. Researchers have applied many technologies, such as neural networks and parallel processing [1-8], to the problem of signature verification and they are continually introducing new ideas, concepts, and algorithms. Other approaches have been proposed and evaluated in the context of random forgeries, like 2D transforms [9], histograms of directional data [1, 6, 10] or curvature [11], horizontal and vertical projections of the writing trace of the signature [12], structural approaches [13], local measurements made on the writing trace of the signature [14] and the position of feature points located on the skeleton of the signature [15]. Following Plamondon et al [16], a handwritten signature is the result of a rapid movement. Hence, the shape of the signature remains relatively the same over time when the signature is written down on a pre-established frame (context) like a bank check. This physical constraint contributes to the relative time-invariance of the signatures, which supports using only static shape information to verify signatures. Some other solutions in the case of random forgeries are mainly based on the use of global shape descriptors as the shadow code, investigated by Sabourin et al [17]. Other approaches using global shape descriptors such as shape envelope projections on the coordinate axes, geometric moments, or even more general global features such as area, height and width, have been widely investigated [3, 4, 15, 18, 19]. In [20], offline models of signature verification are compared based on HMMs. The approach of [21] employs a three expert system that evaluates the signature three different ways and judges it as genuine, forgery, or rejection by a majority vote of the three experts. In [22], a signature verification system is presented that works with both static and dynamic features. In [23], the authors infer that shape similarity and causality of signature's generation are more important than matching the dynamics of signing. This result indicates that this dynamics is not stable enough to be used for signature verification since the subject is trying to reproduce a shape rather than a temporal pattern. This is why we use, in this paper, only static images to verify signatures. A new formalism for signature representation based on visual perception is proposed in [24]. Several learning strategies for signature verification were evaluated using a high-dimensional feature space that captures both local geometric information as well as stroke information in [25]. In [26], a serial three stage multi-expert system for facing the problem of signature verification is proposed. Many other important works in offline signature verification can be found in [27-49].

It is worth pointing out that most of the systems proposed up to now, while performing reasonably well on a single category of forgeries (random, simple or skilled), decrease in performance when working with all the categories of forgeries simultaneously, and generally this decrement is bigger than one would expect. The main reason for this behavior lies in the difficulty of defining a feature set that is adequate to work with all the classes of forgery simultaneously. Also, in current verification systems, it is difficult to find clear and strong justifications of why a specific set of features is used instead of others. Even if a justification is found, extracting such features is difficult and not robust.

In this paper, we suggest a very simple signature verification method that depends only on the raw binary pixel intensities. The method looks at the signature verification problem as a graph matching problem. The rest of the paper is organized as follows. Section 2 is a survey of related work that is based on graph matching. Signature preprocessing operations are described in Section 3. Signature verification is formulated as a graph matching problem in Section 4. We report results in Section 5. Finally, the paper is concluded in Section 6.

2. Graph Matching-Based Related Work

A large number of applications of graph matching have been described in the literature. The following is a survey of some related works. There has been many works in face recognition field. For example, in [59], the authors present a face recognition method using graph matching with a multi-layer grammatical face model. The method increases the robustness of recognition under varying lighting conditions. Furthermore, with high-level semantic understanding of the face, the authors are able to perform an intelligent recognition process driven by the status of the face, i.e. changes in expressions and poses. In [60], a system is proposed to handle the illumination problem of face recognition systems by using "Retinex and color constancy" algorithm. The Retinex and color constancy approach has been plugged with elastic bunch graph matching. The work in [61] presents a formalization of model-based facial feature recognition as an inexact graph matching problem, one graph representing a model of a face and the other an image where recognition has to be performed. A method for enhancing the performance of elastic graph matching in frontal face authentication is proposed in [62].

Graph matching is used in other image similarity problems. In [63], the authors present a part-based image similarity measure derived from stochastic matching of attributed relational graphs that represent the compositional parts and part relations of image scenes. The advantage of this model is its ability to accommodate spatial attributed relations and support supervised and unsupervised learning from training data. In [64], the authors present a method for finding similarities in a pair of three-dimensional objects. The method involves obtaining boundary cubes approximations to the two objects. Having obtained the approximations, they apply exact and inexact graph-matching algorithms to quantify the similarity between two objects. In [65], the problem of computing the similarity between two images is transformed to that of approximating the distance between two extended region adjacency graphs, which are extracted from the images. Invariance to translation, rotation, and scaling could be achieved using this method.

In [66], the authors propose an approach to solve problems such as image registration, pattern detection and localization, and common pattern discovery under a unified framework based on graph matching. Experimental results show that their approach can handle rotation, scaling and translation, as well as distortion and occlusion. A method is proposed in [67] to retrieve images by content. The method is based on matching of strong graphs defined by relational signatures computed between clusters of a fuzzy partition. Such a representation combines color and spatial information between the regions and has nice geometric properties with respect to scale factor, rotation and translation. In [68], the authors describe a tracking algorithm to address the interactions among objects, and to track them individually and confidently via a static camera. It is achieved by constructing an invariant bipartite graph to model the dynamics of the tracking process, of which the nodes are classified into objects and profiles. The best match of the graph corresponds to an optimal assignment for resolving the identities of the detected objects. In [69], a new method is presented for segmentation and recognition of image objects based on structural pattern recognition. The input image is decomposed into regions through a quadtree algorithm. The decomposed image is represented by an attributed relational graph (ARG) named input graph. The objects to be recognized are also stored in an ARG named model graph. Object segmentation and recognition are accomplished by matching the input graph to the model graph.

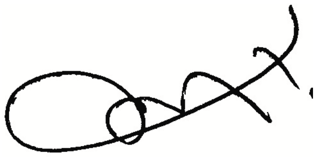
Some other diverse applications of graph matching follow. In [70], a subcircuit recognition method is developed using a nonlinear graph matching strategy. Subcircuit recognition is a problem of identifying all instances of a small subcircuit in a larger circuit. In [71], the authors propose a generic graph matching based framework that resolves the phase-ordering and fixed-ordering problems associated with scheduling on a clustered VLIW processor by simultaneously considering various scheduling alternatives of instructions. In [72], the authors present a new agent teamwork coordination strategy, called Rolegraphs, which represents and recognizes team intentions without requiring full knowledge of plans, or complete observations. The strategy relies on the role relationships formed within hierarchical teamwork structures. A graph matching approach is used to interpret these hierarchical structures, and to recognize team intentions at runtime. The work in [73] presents a graph matching model for the software architecture recovery. Modeling the recovery process as graph matching is an attempt to identify a sub-optimal transformation from a pattern graph, representing the high-level view of the system, onto a subgraph of the software system graph. In [74], the performance of an Attributed Relational Graph (ARG) matching algorithm, tailored for dealing with large graphs, is evaluated in the context of a real application: the detection of component parts in CAD images of mechanical drawings. The matching problem is a graph-subgraph isomorphism and the algorithm exploits semantic information about nodes while does not require information about the topology of the graphs to be matched.

3. Signarure Preprocessing

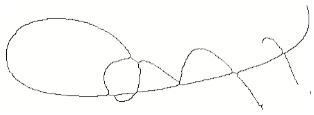
Before a signature can be compared to any other signature, it undergoes some preprocessing operations. A signature is captured as a binary image, S. Then, pepper noise is removed, if there is any, to allow more accurate calculation of center of area. The angle, θ, of least second moment of S is found. The angle is measured counterclockwise from the y-axis. The signature is rotated about the center of area of S θ degrees clockwise. This rotation eliminates skew angle of a signature which is necessary for signatures of the same subject. After rotation, the image is smoothed which is a necessary step before thinning. To reduce data, the image is thinned using Zhang-Suen algorithm [57, 58]. The thinned image undergoes a normalization step that preserves the aspect ratio of the signature. The set of pixels that constitute the final thinned-normalized image of S, denoted as the set of vertices X, represents the signature S. These steps are summarized as follows, where after every image processing operation S is replaced by the resultant image.

1. Remove pepper noise from S.
2. Find the angle of least second moment, θ, of S.
3. Rotate S θ degrees clockwise.
4. Smooth S.
5. Thin S.
6. Normalize S. The set of pixels that constitute the final normalized image, S; denoted as the set of vertices X, represents the signature S.

We illustrate signature preprocessing by an example. Figure 1(a) shows a binary image of a genuine signature. After removing pepper noise, finding the angle of least second moment, rotating the image, smoothing, and thinning, the image of Figure 1(b) is obtained. Normalizing the thinned image produces that of Figure 1(c). The set of pixels, X, of Figure 1(c) represents the genuine signature of Figure 1(a).



(a)



(b)

(c)

Figure 1. Signature preprocessing: (a) original image, (b) image after pepper noise removal, skew elimination, smoothing, and thinning, and (c) final 64 $\times$ 128 normalized image.

4. Comarison Of Two Signatures As A Graph Matching Problem

We introduce the following definitions from graph theory [55].

Definition 1. *Let V be a finite nonempty set of vertices, and let E be a set of unordered pairs of elements taken from V. The pair (V, E) is then called an undirected graph on V, where V is the set of vertices and E is its set of edges. We write $G = (V, E)$ to denote such a graph.*

Definition 2. *A graph $G = (V, E)$ is called bipartite if $V = X \cup Y$ with $X \cap Y = \emptyset$, and every edge of G is of the form $\{x, y\}$ with $x \in X$ and $y \in Y$. If every vertex in X is joined with every vertex in Y, we have a complete bipartite graph. In this case, if $|X| = m$, and $|Y| = n$, the graph is denoted $K_{m,n}$.*

Definition 3. *Let $G = (V, E)$ be a bipartite graph as defined above. A matching in G is a subset of E such that no two edges share a common vertex in X or Y. A complete matching of X into Y is a matching in G such that every $x \in X$ is the end point of an edge.*

For a bipartite graph $G = (V, E)$ with V partitioned as $X \cup Y$, a complete matching of X into Y requires $|X| \leq |Y|$. If $|X|$ is large, then the construction of such a matching cannot be accomplished just by observation or trial and error. The following theorem, [55], provides a necessary and sufficient condition for the existence of such a matching.

Theorem 1. *Let $G = (V, E)$ be a bipartite graph with V partitioned as $X \cup Y$. A complete matching of X into Y exists if and only if for every subset A of X, $|A| \leq |R(A)|$, where $R(A)$ is the subset of Y consisting of vertices each of which is adjacent to at least one vertex in A.*

Let S_1 and S_2 be two offline signature images to be compared. Let X and Y be the sets of vertices (pixels) that represent S_1 and S_2, respectively. Cleary, X and Y are disjoint sets. We construct a complete bipartite graph $G = (V, E) = K_{m,n}$, from X and Y where $V = X \cup Y$, $|X| = m$, and $|Y| = n$. Since G is complete and assuming that the signatures are ordered such that $|X| \leq |Y|$, then according to Theorem 1, a complete matching of X into Y exists. Usually, there are too many such complete matchings. Our goal is to find the minimum cost complete matching of X into Y. This is some form of the well known Assignment Problem (AP) from graph theory. We use the algorithm of [56], referred to as the Hungarian Method, to solve this assignment problem; i.e., find how much the signatures S_1 and S_2 match.

The key point in using the Hungarian method to solve our AP problem is to find the cost matrix C which is $m \times n$ matrix whose rows correspond to the vertices of X and whose columns correspond to the vertices of Y. Since the vertices X and Y are originally pixels extracted from signature images S_1 and S_2, respectively, every vertex $x \in X$ or $y \in Y$ has its x and y coordinates (row and column numbers in the raster image). These coordinates are used to find the distance between x and y after aligning the centers of area of the sets X and Y. Then, an entry c_{xy} of C, $x \in X$, $y \in Y$, equals the Euclidean distance between x and y. This is the cost of matching point x from signature S_1 to point y from signature S_2. After calculating all entries of C, the formulated assignment problem is solved. The cost, $cost_{min}$, of the resultant solution equals the sum of all entries, c_{xy}, that correspond to the minimum cost solution. This cost is normalized by dividing it by $|X|$ yielding the normalized minimum cost per pixel. Then, $cost_{min}$ is further divided by a factor, f, which takes into account the columns of C, which could not be matched. The factor f, measures the percentage of vertices sharing in any complete matching X into Y.

To verify that a test signature, S, belongs to a specific subject, it is compared against a predetermined number, p, of prototype genuine signatures of the same subject as follows:

1. The p prototype signatures are preprocessed as described in Section 3 to produce the sets of vertices $Y_1, Y_2, \ldots, Y_p$.
2. S is preprocessed as described in Section 3 to produce the set of vertices X.
3. Let $d = \infty$, where d will measure the minimum distance between S and the prototype signatures of the considered subject.

```
For every set of vertices Y_i, i = 1, 2, ... , p, do
{
    Let f = min(|X|, |Y_i|)/ max(|X|, |Y_i|).
    If f ≥ α, then
    {
        Find the cost matrix C between X and Y_i as described earlier.
        Rotate the matrix C so that there are at least as many rows as columns.
        Let r = number of rows of C.
        Compute the minimum cost matching, cost_min, of rows into columns considering
        the cost matrix C.
        Let cost_min = cost_min /( f × r).
        If cost_min < d, then let d = cost_min.
    }
}
```

4. If d is less than or equal to some calculated threshold, D, then signature S is accepted, otherwise, it is rejected.

In step 3, above, sometimes, rotating C is required by the Hungarian matching algorithm, since X and Y_i will correspond to rows and columns of the cost matrix, respectively, and the algorithm finds a complete matching of rows into columns, which requires that the number of rows be less than or equal to the number of columns. Again, the factor, f, measures the percentage of vertices sharing in any complete matching of rows into columns. We continue to find a matching if f is not less than predetermined threshold, α, since such signatures most probably don't belong to the same subject. If $f \geq \alpha$, then a minimum cost matching is found. However, columns that could not be matched are punished by dividing the minimum cost, $cost_{min}$, by f. The cost is normalized by dividing by the number of rows, r. The distance threshold, D, depends on the size of the normalization box, differs from subject to subject, and is calculated as follows:

$$D = B \times D_{\max} \tag{1}$$

where B is constant, and for a specific subject, $D_{\max}$ is the maximum distance between any two of his p prototype signatures, and is calculated as follows:

1. The p prototype signatures are preprocessed as described in Section 3 to produce the sets of vertices $Y_1, Y_2, \ldots, Y_p$.

2. Let $D_{\max} = 0$.

3. For every set of vertices Y_i, $i = 1, 2, \ldots, p - 1$, do

```
For every set of vertices Y_j, j = i + 1, i + 2, ... , p, do
{
    Let f = min(|Y_i|, |Y_j|)/ max(|Y_i|, |Y_j|).
    If f ≥ α, then
    {
        Find the cost matrix C between Y_i and Y_j as described earlier.
        Rotate the matrix C so that there are at least as many rows as columns.
        Let r = number of rows of C.
        Compute the minimum cost matching, cost_min, of rows into columns considering
        the cost matrix C.
        Let cost_min = cost_min / (f × r).
        If cost_min > D_max, then let D_max = cost_min.
    }
}
```

5. Results

Datasets: A grid of 5 rows by 3 columns was prepared and printed on an A4 paper to collect signatures of subjects. Each cell of the grid is 6.3 cm width and 4.5 cm height. This size is usually sufficient to handwrite a signature freely. Subjects were asked to provide 15 of their genuine signatures on an A4 page with the described grid. Subjects were also asked not to touch the borders of the grid cells to facilitate signature segmentation. Although we explained to subjects that their signatures will be solely used in a scientific research study and this was written clearly in Arabic on the top of the page, most of the subjects refused to provide any signature. Only five subjects agreed and provided 15 genuine signatures each. Therefore, the total number of collected genuine signatures is $15 \times 5 = 75$ signatures. Figure 2 shows one sample page containing genuine signatures of subject No. 1.

The same grid was used to collect forgeries of signatures of the collected genuine signatures. Each of the same five subjects was trained on signatures of the others. Then, he/she was asked to provide from 2 to 4 forgery signatures for each of the other genuine signatures. 15 forgeries were collected for each genuine signature. So, the total number of forgery signatures is $15 \times 5 = 75$ signatures. Figure 3 shows one sample page of 15 forgeries of the genuine signatures shown in Figure 2.

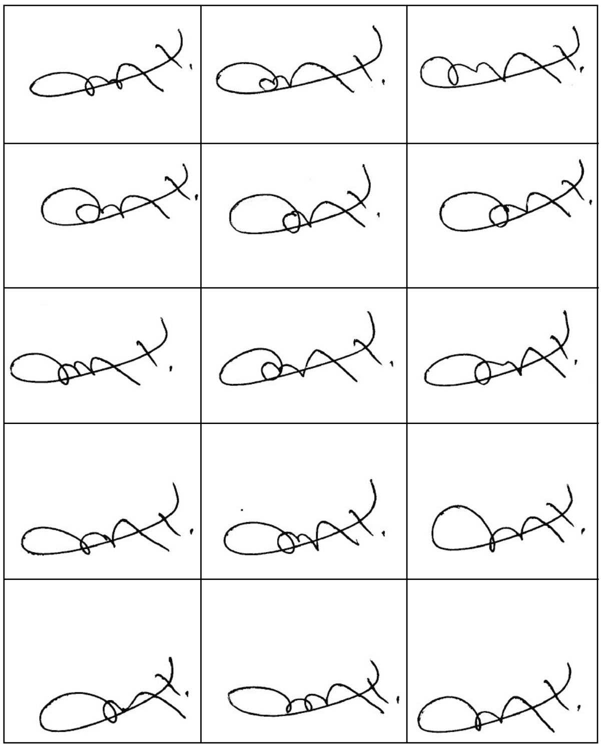


Figure 2. Fifteen genuine signatures of subject No. 1.

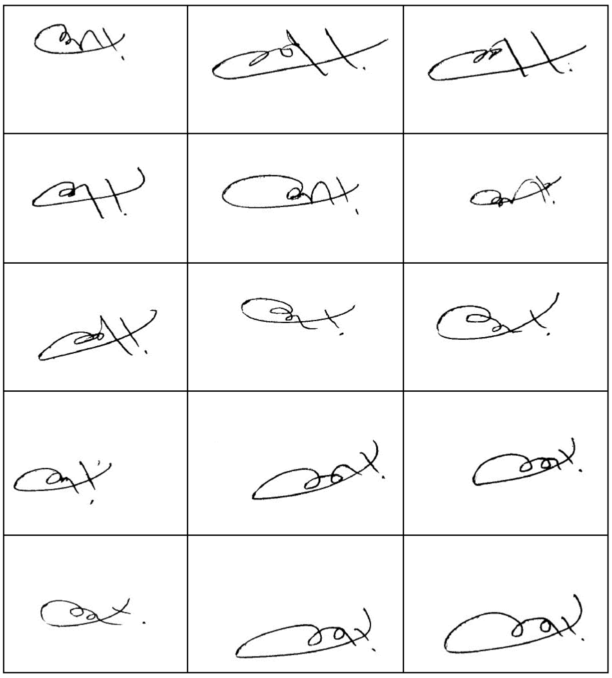


Figure 3. Fifteen forgeries of signature of subject No. 1. These signatures were produced by the other four subjects.

After signature collection, they were scanned as binary images using an hp ScanJet 3400C scanner. The resolution was set to 300 dpi. The algorithm was run on a Pentium III PC, 850 MHz with 384 MB RAM. Three types of tests were conducted:

1. Genuine test, where genuine signatures were verified by considering the first three genuine signatures of every subject as prototype signatures, i.e., $p = 3$. The rest 12 signatures of every subject are tested against his three prototype signatures. Therefore, the total number of tested signatures equals $12 \times 5 = 60$. This test is used to compute the false rejection rate, *FRR*.

2. Random forgery test, where for every subject, all genuine signatures of all other subjects are considered random forgeries of signature of the subject under consideration. This gives $5 \times (5–1) \times 15 = 300$ signatures to be tested. A false acceptance rate, *FAR_random*, is calculated specifically for this test.

3. Skilled forgery test, where for every subject, 15 skilled forgeries are tested yielding a total of $15 \times 5 = 75$ skilled test forgeries. Another false acceptance rate, *FAR_skilled*, is computed for this test.

The algorithm was run many times using the following sizes of normalization boxes: 8×16, 16×32, 24×48, 32×64, 40×80, 48×96, 56×112, and 64×128. For every specific normalization box, the first three signatures of every subject were preprocessed just once and used in all tests. The ratio, α, of the number of rows to number of columns of cost matrix, C, was set to 0.90. The factor, B, used to calculate $D_{\max}$ was allowed to vary from 0 to 3 in increments of 0.05.

Figure 4(a-h) displays how the false rejection rate, *FRR*, false acceptance rate for random forgeries, *FAR_random*, and false acceptance rate for skilled forgeries, *FAR_skilled*, change against B, for different sizes of normalization box. As B gets larger, *FRR* gets smaller and the two *FARs* get larger. It is natural to notice that the *FAR_random* curve is lower than the *FAR_skilled* curve, since in random forgeries the signer has no previous knowledge and/or training on the signature he is forging. He just provides any signature that can be easily detected as the curve shows. However, in skilled forgeries, the forger does not sign until he has known the model of the signature to be forged, and he has been trained on that signature. Curves of equal error rate, *EER*, defined as the rate at which *FRR* = *FAR*, against size of normalization box are displayed in Figure 5. Notice that as the size increases, *EER* decreases until it reaches 26.7% and 5.6% for skilled and random forgeries, respectively, at size 32×64 pixels, $B = 1.1655$. After this size, *EER* fluctuates. Returning to Figure 4, we notice, except for part (a), that *FAR _random* vanishes at the point of equal false rejection and skilled forgery false acceptance rates, which is a positive property of our algorithm.

Samples of rejected genuine signatures and accepted skilled forgeries are shown in Figure 6, for subject No. 1, 32×64 normalization box, $B = 1.1655$, at the point of equal false rejection rate and false acceptance rate of skilled forgeries. For this subject, 3 out of 12 genuine signatures were rejected and only one forgery out of 15 was accepted. All random forgeries were rejected for this subject.

In [24], two types of classifiers, a nearest neighbor and a threshold classifier, are used for offline signature verification. These classifiers show a total error rate below 2% and 1%, respectively, in the context of random forgeries. These rates are better that ours which is 5.6%. For skilled forgeries, the false acceptance rate of our algorithm is near those of other researchers. However, the main advantage or our approach is its simplicity compared to that of [24] and other approaches.

Figure 7 shows the average processing time required to verify one signature. It is noticed that the time increases exponentially as the size of the normalization box increases, which is a well known behavior of the Hungarian method for complete graph matching. Keeping the size at 32×64 pixels makes the time in the two seconds range. Also, using more powerful computers should decrease processing time.

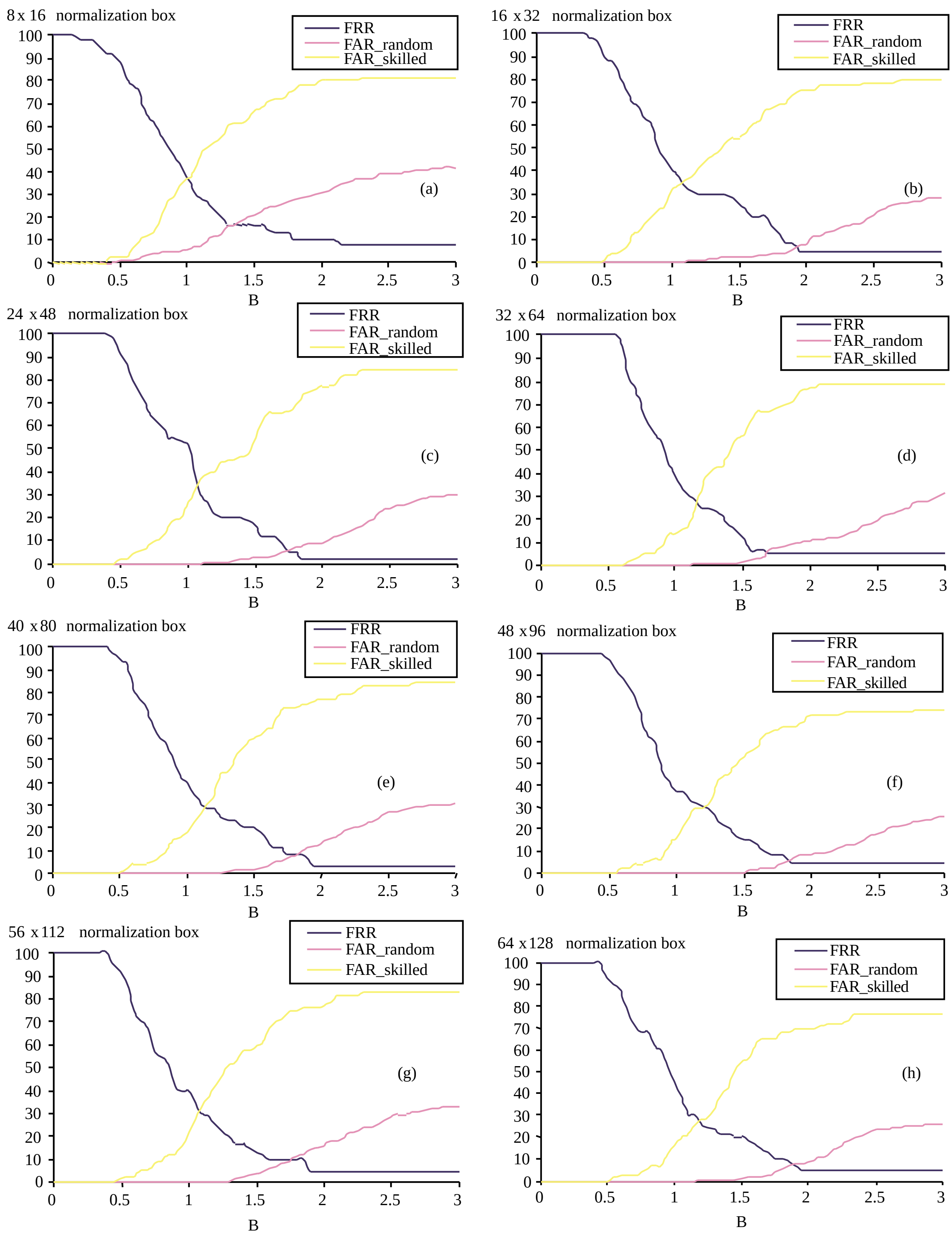


Figure 4. False rejection and acceptance rates against B: with different sizes of normalization box: (a) 8 × 16, (b) 16 × 32, (c) 24 × 48, (d) 32 × 64, (e) 40 × 80, (f) 48 × 96, (g) 56 × 112 and (h) 64 × 128.

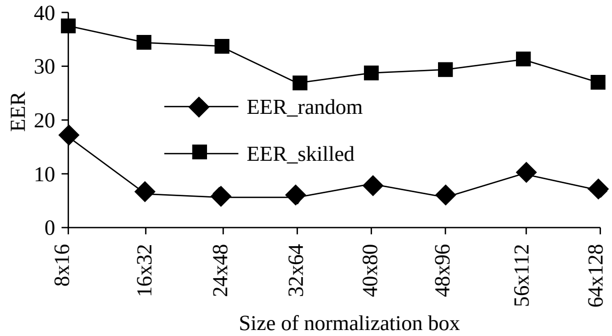


Figure 5. Equal error rate against size of normalization box.

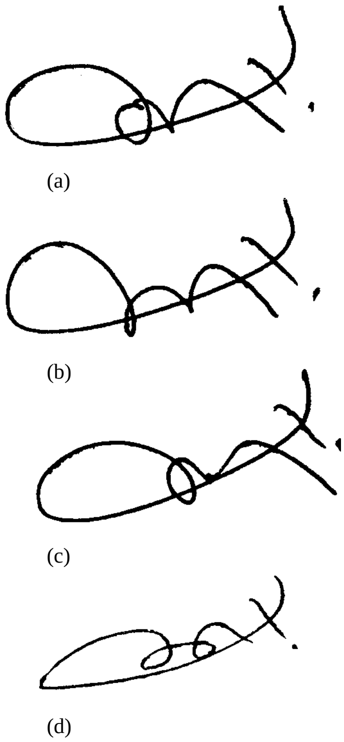


Figure 6. Examples of false rejection and false acceptance for subject No. 1: (a), (b), and (c) rejected genuine signatures, and (d) accepted forgery.

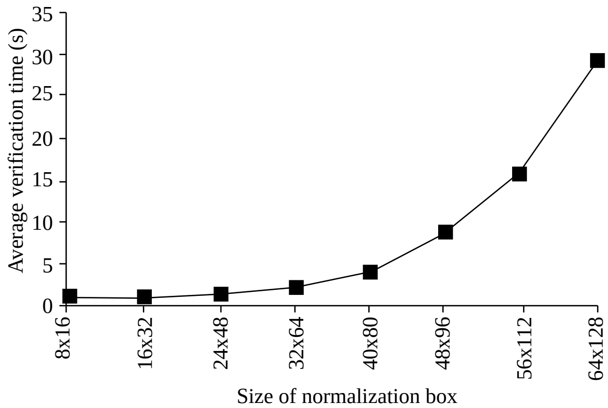


Figure 7. Average verification time versus size of normalization box.

6. Conclusion

In this paper, we investigated the problem of offline signature verification. From the results, it is obvious that the problem of signature verification becomes more difficult when passing from random to skilled forgeries, the latter being so difficult a task that even human beings make errors in several cases.

We presented a simple and effective signature verification method that depends only on the raw binary pixel intensities and avoids using complex sets of features that are usually used in the literature. Our method looks at the signature verification problem as a graph matching problem, for which we used the Hungarian method to solve.

The method was tested using genuine and forgery signatures produced by five subjects. Three types of tests were conducted: (1) genuine test, where genuine signatures were verified, (2) random forgery test, where for every subject, all genuine signatures of all other subjects are considered random forgeries of signature of the subject under consideration, and (3) skilled forgery test, where for every subject, skilled forgeries are tested. An equal error rate of 26.7% and 5.6% for skilled and random forgeries, respectively, was achieved at size 32 × 64 pixels. A positive property of our algorithm is that the false acceptance rate of random forgeries vanishes at the point of equal false rejection and skilled forgery false acceptance rates.

Concerning processing time, It is noticed that it increases exponentially as the size of the normalization box increases. Keeping the size at 32×64 pixels makes the time in the two seconds range. Also, using more powerful computers should decrease processing time.

Acknowledgement

The author would like to thank the referees for their valuable comments which helped to improve paper presentation.

References

[1] J.P. Drouhard, R. Sabourin, M. Godbout, "Evaluation of a Training Method and of Various Rejection Criteria for a Neural Network Classifier Used for Off-Line Signature Verification," IEEE Int'l Conf. Neural Networks, Orlando, Fla., June 26-July 2, pp. 294-4,299, 1994.

[2] R. Sabourin, J.P. Drouhard, "Off-Line Signature Verification Using Directional PDF and Neural Networks," Proc. of the 11^{th} ICPR, pp. 321-325, The Hague, The Netherlands, 1992.

[3] H. Cardot, M. Revenu, B. Victorri, M.J. Revillet, "A static signature verification system based on a cooperative neural network architecture," International Journal of Pattern Recognition and Artificial Intelligence, 8(3), pp. 679–692, 1994.

[4] R. Bajaj, S. Chaudhury, "Signature verification using multiple neural classifiers," Pattern Recognition, 30(1), pp. 1–7, 1997.

[5] K. Huang, H. Yan, "Off-line signature verification based on geometric feature extraction and neural network classification," Pattern Recognition, 30(1), pp. 9–17, 1997.

[6] J.P. Drouhard, R. Sabourin, M. Godbout, "A neural network approach to off-line signature verification using directional PDF," Pattern Recognition, 29(3), pp. 415–424, 1996.

[7] L.L. Lee, "Neural approaches for human signature verification," Third International Conference on Signal Processing Proceedings, pp. 1346–1349, 1996.

[8] M.C. Fairhurst, P. Brittan, "An Evaluation of Parallel Strategies for Feature Vector Construction in Automatic Signature Verification Systems," Int. J. Pattern Recognition and Artificial Intelligence, vol. 8, no. 3, pp. 661-678, 1994.

[9] W.F. Nemcek, W.C. Lin, "Experimental Investigation of Automatic Signature Verification," IEEE Trans. Systems, Man and Cybernetics, pp. 121-126, 1974.

[10] T.S. Wilkinson, J.W. Goodman, "Slope Histogram Detection of Forged Handwritten Signatures," Proc. SPIE, pp. 293-304, Boston, 1990.

[11] E.R. Brocklehurst, “Computer Methods of Signature Verification,” J. Forensic Science Society, pp. 445-457, 1985.

[12] M. Ammar, Y. Yoshida, T. Fukumura, “Off-Line Preprocessing and Verification of Signatures,” Int’l J. Pattern Recognition and Artificial Intelligence, vol. 2, no. 4., pp. 589-602, 1988.

[13] R. Sabourin, R. Plamondon, L. Beaumier, “Structural Interpretation of Handwritten Signature Images,” Int. J. on Pattern Recognition and Artificial Intelligence, vol.8, no. 3, pp. 709-748, 1994.

[14] R.N. Nagel, A. Rosenfeld, “Computer Detection of Freehand Forgeries,” IEEE Trans. Computers, vol. 26, no. 9, pp. 895-905, 1977.

[15] Y. Qi, B. R. Hunt, “Signature verification using global and grid features,” Pattern Recognition, 27(12), pp. 1621–1629, 1994.

[16] R. Plamondon, G. Lorette, “Automatic signature verification and writer identification—the state of the art,” Pattern Recognition, 22, pp. 107–131, 1989.

[17] R. Sabourin, M. Cheriet, G. Genest, “An Extended-Shadow-Code Based Approach for Off-Line Signature Verification,” Second IAPR Conf. Document Analysis and Recognition, pp. 1-5,Tsukuba, Japan, Oct. 1993.

[18] M. Ammar, “Progress in verification of skillfully simulated handwritten signatures,” International Journal of Pattern Recognition and Artificial Intelligence, 5(1–2):337–351, 1991.

[19] N.A. Murshed, R. Sabourin, F. Bortolozzi, “A cognitive approach to off-line signature verification,” International Journal of Pattern Recognition and Artificial Intelligence, 11(5), pp. 801–825, 1997.

[20] G. Rigoli, A. Kosmala, “A Systematic Comparison Between on-line and off-line Methods for Signature Verification with Hidden Markov Models,” 14th International Conference on Pattern Recognition - vol. II, pp. 1755—1757, Australia, 1998.

[21] G. Dimauro, S. Impedovo, G. Pirlo, A. Salizo, “A multi-expert signature verification system for bank check processing,” International Journal of Pattern Recognition and Artificial Intelligence, 11(5), pp. 827–844, 1997.

[22] Q. Wu, S. Lee, I. Jou, “On-line signature verification based on split-and-merge matching mechanism,” Pattern Recognition Letters, 18(7), pp. 665-673, 1997.

[23] M. Munich, P. Perona, “Visual signature verification using affine arc-length,” Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn., 1999.

[24] R. Sabourin, G. Genest, F.J. Prêteux, “Off-Line Signature Verification by Local Granulometric Size Distributions,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 9, pp. 976-988, 1997.

[25] S.N. Srihari, A. Xu, M.K. Kalera, “Learning strategies and classification methods for off-line signature verification,” Proc. of the 7th Int. Workshop on Frontiers in handwriting recognition (IWHR) , pp. 161–166, 2004.

[26] C. Sansone, M. Vento, “Signature Verification: Increasing Performance by a Multi-Stage System,” Pattern Analysis & Applications, No. 3, pp. 169–181, 2000.

[27] C. Allgrove, M. C. Fairhurst, “Majority Voting for Improved Signature Verification,” IEE Colloquium on Visual Biometrics, (Ref No. 2000/018), pp. 911-914, 2000.

[28] G. Dimauro, S. Impedovo, G. Pirlo, “Component-Oriented Algorithms for Signature Verification,” Int. J. Pattern Recognition and Artificial Intelligence, vol. 8, no. 3, pp. 771-793, 1994.

[29] N.A. Murshed, F. Bortolozzi, R. Sabourin, “Off-Line Signature Verification, Without a Priori Knowledge of Class w2. A New Approach, ”Proc. Third IAPR Conf. Document Analysis and Recognition, pp. 191-196, Aug. 14-16, Montréal, Canada, pp. 191-196, 1995.

[30] F. Nouboud, R. Plamondon, “Global Parameters and Curves for Off-Line Signature Verification,” Proc. Int’l Workshop on Frontiers in Handwriting Recognition, Taiwan, pp. 145-155, 1994.

[31] J. Mathyer, “The Expert Examination of Signatures,” J. Criminal Law, Criminology, and Police Science, vol. 5, no. 3, May-June, pp. 122-133, 1961.

[32] R. Sabourin, R. Plamondon, G. Lorette, “Off-Line Identification with Handwritten Signature Images: Survey and Perspectives,” Structured Document Image Analysis. New York: Springer-Verlag, pp. 219-234, 1992.

[33] R. Sabourin, G. Genest, “An Extended-Shadow-Code Based Approach for Off-Line Signature Verification: Part 2—Evaluation of Several Multi-Classifier Combination Strategies, ”Third IAPR Conf. Document Analysis and Recognition, pp. 197-201, Montréal, Canada, Aug. 14-16, 1995.

[34] M. Ammar, Y. Yoshida, T. Fukumura, “A new effective approach for off-line verification of signatures by using pressure features”. Proceedings of the 8th International Conference on Pattern Recognition, pages 566–569, 1986.

[35] J.K. Guo, D. Doermann, A. Rosenfeld, “Local correspondence for detecting random forgeries,” Proceedings of the International Conference on Document Analysis and Recognition, pp. 319–323, 1997.

[36] M.K. Kalera, B. Zhang, S.N. Srihari, “Off-line signature verification and identification using distance statistics,” Proceedings of the International Graphonomics Society Conference, Scottsdale, AZ, pp. 228–232, November 2003.

[37] S. Lee, J.C. Pan, “Off-line tracing and representation of signatures,” IEEE Transactions on Systems, Man and Cybernetics, Vol. 22, pp. 755–771, 1992.

[38] M.E. Munich, P. Perona, “Visual identification by signature tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(2), pp. 200–217, 2003.

[39] R. Plamondon, G. Lorette, “On-line and off-line handwriting recognition: A comprehensive survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), pp. 63–84, 2000.

[40] R. Sabourin, R. Plamondon, “Preprocessing of handwritten signatures from image gradient analysis,” Proceedings of the 8th International Conference on Pattern Recognition, pp. 576–579, 1986.

[41] R. Plamondon, F. Leclerc, ”Automatic signature verification: the state of the art 1989–1993,” International Journal of Pattern Recognition and Artificial Intelligence, 8(3), pp. 643–660, 1994.

[42] L.L. Lee, M.G. Lizarraga, N.R. Gomes, A.L. Koerich, “A prototype for Brazilian bank check recognition,” International Journal of Pattern Recognition and Artificial Intelligence, 11(4), pp. 549–569, 1997.

[43] Y. Qi, B.R. Hunt, “A multiresolution approach to computer verification of handwritten signatures,” IEEE Transactions on Image Processing, 4(6), pp. 870–874, 1995.

[44] P.S. Deng, H.–Y. M. Liao, C.W. Ho, H.–R. Tyan, “Wavelet-based off-line signature verification,” Computer Vision and Image Understanding, 76, pp. 173–190, 1999.

[45] B. Fang, C.H. Leung, Y.Y. Tang, K.W. Tse, P.C.K. Kwok, Y.K. Wong, “Off-line signature verification by the tracking of feature and stroke positions,” Pattern Recognition, 36, pp. 91–101, 2003.

[46] T.J. Hastie, E. Kishon, M. Clark, J. Fan, “A model for signature verification,” in Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, 1, pp. 191–196, 1991.

[47] J.S. Lew, “Kinematic theory of signature verification measurements,” Mathematical Biosciences, 48, pp. 25–51, 1980.

[48] T. Matsuura, H. Sakai, “On stochastic system representation of handwriting process and its application to signature verification,” Third International Conference on Signal Processing Proceedings, pp. 1330–1333, 1996.

[49] R. Sabourin, J.–P. Drouhard, E.S. Wah, “Shape matrices as a mixed shape factor for offline signature verification,” Proceedings of the Fourth International Conference on Document Analysis and Recognition, pp. 661–665, 1997.

[50] R. Plamondon, “The Design of An On-Line Signature Verification System: From Theory to Practice,” Int. J. Pattern Recognition and Artificial Intelligence, vol. 8, no. 3, pp. 795-811, 1994.

[51] R. Plamondon, S.N. Srihari, “On-Line and Offline Handwriting Recognition: A Comprehensive Survey,” IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, 2000.

[52] V.S. Nalwa, “Automatic On-Line Signature Verification,” Proceedings of the IEEE, Vol. 85, No. 2, pp. 215-239, 1997.

[53] L. Lee, T. Berger, E. Aviczer, “Reliable on-line Human Signature Verification Systems,” IEEE Trans. Pattern Analysis and Machine Intelligence, 18(6), pp. 643-647, 1996.

[54] B. Kashi, J. Hu, W.L. Nelson, W. Turin, “Hidden Markov Model approach to online handwritten signature verification” International Journal on Document Analysis and Recognition, 1(2), pp. 319-330, 1998.

[55] R.P. Grimaldi, Discrete and Combinatorial Mathematics, Addison-Wesley Longman Inc., forth edition, 1999.

[56] N. Christofides, Graph Theory: An Algorithmic Approach, Academic Press Inc., 1977.

[57] T. Zhang, C. Suen, “A fast parallel algorithm for thinning digital patterns,” Communications of the ACM, vol. 27, pp. 236-239, Mar., 1984.

[58] G.X. Ritter, J.N. Wilson, Handbook of Computer Vision Algorithms in Image Algebra, CRC Press, 2^{nd} edition, 2001.

[59] Z. Xu, J. Luo, ”Face Recognition by Expression-Driven Sketch Graph Matching”, 18th International Conference on Pattern Recognition (ICPR’06), pp. 1119-1122, 2006.

[60] N. Kela, A. Rattani, P. Gupta, ”Illumination Invariant Elastic Bunch Graph Matching for Efficient Face Recognition”, 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’06), p. 42, 2006.

[61] R. Cesar, E. Bengoetxea, I. Bloch, ”Inexact Graph Matching Using Stochastic Optimization Techniques for Facial Feature Recognition”, 16th International Conference on Pattern Recognition (ICPR’02) - Volume 2, p. 20465, 2002.

[62] A. Tefas, C. Kotropoulos, I. Pitas, "Using Support Vector Machines to Enhance the Performance of Elastic Graph Matching for Frontal Face Authentication ", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 7, pp. 735-746, 2001.

[63] D.-Q. Zhang, S.-F Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning", Proceedings of the 12th annual ACM international conference on Multimedia, New York, NY, USA, pp. 877–884, 2004.

[64] C. Maple, Y. Wang, "A Three-dimensional Object Similarity Test Using Graph Matching Techniques", Eighth International Conference on Information Visualisation (IV'04), pp. 363-369, 2004.

[65] R. Baeza-Yates, G. Valiente, " An Image Similarity Measure Based on Graph Matching", Seventh International Symposium on String Processing Information Retrieval (SPIRE'00), p. 28, 2000.

[66] H. Jiang, Chong-Wah Ngo, "Graph Based Image Matching", 17th International Conference on Pattern Recognition (ICPR'04) - Volume 3, pp. 658-661, 2004.

[67] S. Tabbone, L. Wendling, "Retrieving Images by Content from Strong Relational Graph Matching", 17th International Conference on Pattern Recognition (ICPR'04) - Volume 2, pp. 951-954, 2004.

[68] Hwann-Tzong Chen, Horng-Horng Lin, Tyng-Luh Liu, "Multi-Object Tracking Using Dynamical Graph Matching", 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01) - Volume 2, p. 210, 2001.

[69] L. Augusto C., R.M. Cesar-Jr., " Quadtree-Based Inexact Graph Matching for Image Analysis", XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05), pp. 205-212, 2005.

[70] N. Rubanov, "An efficient subcircuit recognition using the nonlinear graph matching", Proceedings of the 18th annual symposium on Integrated circuits and system design, Florianolpolis, Brazil, pp. 44-49, 2005.

[71] R. Nagpal, Y.N. Srikant, "A Graph Matching Based Integrated Scheduling Framework for Clustered VLIW Processors", 2004 International Conference on Parallel Processing Workshops (ICPPW'04), pp. 530-537, 2004.

[72] S. Soon, A. Pearce, M. Noble, "Adaptive Teamwork Coordination Using Graph Matching over Hierarchical Intentional Structures", Third International Joint Conference on Autonomous Agents and Multi-agent Systems - Volume 1 (AAMAS'04), pp. 294-301, 2004.

[73] K. Sartipi, K. Kontogiannis, "On Modeling Software Architecture Recovery as Graph Matching", 19th IEEE International Conference on Software Maintenance (ICSM'03), p. 224, 2003.

[74] L.P. Cordella, P. Foggia, C. Sansone, M. Vento, " Fast Graph Matching for Detecting CAD Image Components", 15th International Conference on Pattern Recognition (ICPR'00) - Volume 2, p. 6034, 2000.