# Handwritten Digits Segmentation based on Structural Approach

LUIZ E. SOARES DE OLIVEIRA[1], EDOUARD LETHELIER[1], FLÁVIO BORTOLOZZI[1], ROBERT SABOURIN[2]

[1]PUC-PR Pontifícia Universidade Católica do Paraná
PPGIA - Programa de Pós-Graduação em Informática Aplicada
LARDOC - Laboratório de Análise e Reconhecimento de Documentos
Rua Imaculada Conceição 1155, 80215-901 - Curitiba, PR - BRAZIL
{soares, edouard, fborto}@ppgia.pucpr.br

[2]ETS Ecole de Technologie Supérieure
LIVIA - Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle
1100, rue Notre Dame Ouest, Montréal, (Quebec) H3C 1K3, CANADA
sabourin@gpa.etsmtl.ca

**Abstract.** This article deals with a new segmentation approach applied to unconstrained handwritten digits. The novelty of the proposed algorithm is based on the combination of two types of structural features in order to provide the best segmentation path between connected entities. This method was developed to be applied in a segmentation-based recognition system. In this article, we first present the features used to generate our basic segmentation points, and we define our segmentation paths depending on the encountered configurations with only few heuristic rules. Then, we present a strategy based on graphs in order to manager the segmentation. Finally, we evaluate the output of our segmenter using a integrated classifier with three different combination methods.

## 1 Introduction

Off-Line handwritten numeral string recognition has been a topic of intensive research for recent years, due to its large number of potential applications. Moreover, the various peculiarities of the unconstrained handwriting stay as an open problem in handwriting recognition field.

The challenge of a segmentation technique lies in the decision of the best cut path to localize an entity to be recognized as a correct isolated character by the recognition system. The literature usually shows three different strategies to perform the segmentation : Segmentation-Recognition [12] [16] Segmentation-based Recognition [5] [11] and Segmentation-Free systems [14] [15].

In the first approach, the segmentation provides a single sequence hypotheses where each sub-sequence should contain an isolated character, which is submitted to the recognizer. This technique shows rapidly its limits when the correct segmentation does not fit as well as the pre-defined rules of the segmenter. Very often, contextual information is used during the segmentation process in order to improve the robustness of the system.

The second strategy is based on a probabilistic assumption where the final decision must express the best segmentation-recognition score of the input image. Usually, the system yields a list of hypotheses from the segmentation module and each hypothesis is then evaluated by the recognition. Finally, the list is post-processed taking into account the contextual information. Although this approach gives better reliability than the previous one, the main drawback lies in the computational effort needed to compare all the hypotheses generated. Moreover, the recognition module has to discriminate various configurations such as fragments, isolated characters and connected characters.

In the third technique, the recognizer is trained with the peculiarities of the segmentation. For example, two connected characters of class "00" correspond to a specific class. Due to the large variability of the handwriting, we can easily imagine the problem of considering all the possible configurations to be learned by the system. But in particular applications where connections are restricted to a few class-set, this approach is certainly well suited.

Our works are based on the segmentation-based recognition strategy. The aim of this article is to show how we defined a new segmentation algorithm taking into account two complementary sets of structural features. The final objective of the module is to provide the best hypotheses list of segmentation paths without any a priori knowledge of the context, such as the number of characters to be segmented. Therefore we focused our segmentation work on the limitation of heuristic rules to consider most of configurations in connected characters. We worked with binary images with a 300 dpi resolution.

In the section 2, we present the sets of features used to generate the segmentation points. Then, the section 3 shows the generation of the segmentation paths and the section 4 presents the management of the segmentation. In the sec-



0-7695-0878-2/00 $10.00 © 2000 IEEE

tion 5 we present the combination methods that we use to design our integrated classifier. In section 6, we present the evaluation of the segmentation approach by means of a integrated classifier trained on isolated characters. Finally, section 7 presents our conclusions and our future works.

## 2 Generation of the Segmentation Features

Depending on the context, a lot of features are available to find plausible segmentation points in a character image. Certainly some of the most used in the literature are the contour and profile features [7] [3] [12]. Indeed, these features are easy to be provided and they usually express directional variability of the character strokes and then possible cuts.

However, they are not even fully informative to localize any kind of connection between characters, mostly when the handwriting is strongly skewed or overlapped. Therefore, we opted to consider a second set of features provided by the skeleton: the *intersection points*. These points are often located in the neighborhood of stroke connections where contour and profile features are not always available.

Let us define the contour as the image envelope. This is a bi-dimensional data where each contour point $CP_i$ is associated with the coordinates $(X_i, Y_i)$ of the image.

The profile image is obtained from a vertical projection of the first encountered transition, in both ways top-down and bottom-up. From these both sets of features, we are able to localize the first list of potential cuts which correspond to the local minima of the contour and profile (Figure 1a and 1b). We define these points as *Basic Points* $(BPs)$.

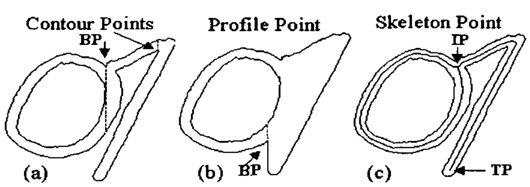


Figure 1: Segmentation points generated: (a) Contour (b) Profile (c) Skeleton Points (Intersections and Terminations)

Let us define now characteristics of the skeleton :

**Definition 1:** In a binary image, the intensity of a pixel $p$, denoted $I(p)$, is either 0 (white) or 1 (black). A pixel $p$ is a *foreground pixel* iff $I(p) = 1$. Conversely, a pixel $p$ is a *background pixel* iff $I(p) = 0$.

**Definition 2:** Using the Freeman directions [9], when the eight neighbors of a pixel $p$ are traced clockwise, the neighborhood of $p$ is denoted $T(p)$, such as:

$$T(p) = \sum_{i=1}^{8} I(p'_i) \tag{1}$$

where $p'_i$ is the $i^{th}$ neighbor of $p$. We can then define the *terminal point* $(TP)$ when $T(p) = 1$ and the *intersection point* $(IP)$ when $T(p) > 2$. The *intersection point* and *terminal point* are called *characteristic points* of the skeleton.

**Definition 3:** A skeleton path $(P_{skeleton})$ is a pixel sequence of the skeleton where each extremity corresponds to a *characteristic point.*

The skeleton image is obtained by using the thinning algorithm proposed by Jang et al [1]. The best advantage of this algorithm relies on the limited number of $IPs$ generated in the neighborhood of connected strokes of different characters. During the detection process of the $IPs$, two particular configurations are first selected as showed in Figure 2. We have noticed that particular digit connections, such as "00" for example, often generate these $IPs$ classes. All other types of $IPs$ are considered as a single $IP$ class (Class 3). The first class (Figure 2a) contains all $IPs$ with one segment in its lower part and two segments in its upper part. Depending on the Freeman directions, each class contains all the variations which respect this definition. The second class (Figure 2b) is the symmetric of Class 1.


Figure 2: Skeleton intersection points: (a) Class 1 (b) Class 2

## 3 Determination of Segmentation Paths

Once selected from the image, the $BPs$ and the $IPs$ are compared altogether in order to determine the list of segmentation hypotheses of the character image.The algorithm scans all possible relationships between $BPs$ and $IPs$ and generates a set of segmentation paths where the goal is to get the correct segmentation paths (for connected characters) in the list of hypotheses.

This association takes into account the distance between $BPs$ and $IPs$. When an $IP$ is located in the $BP$ neighborhood, this could indicate a possible stroke connection between two characters. Then our algorithm uses only the relationship between $IPs$ and $BPs$ to provide the local cut path. To determine the proximity between points, we based our comparison on the estimation of the thickness of the strokes $E_t$, obtained with the projection of the density histogram. Then, two points $BP_i$ and $IP_j$ belong to the same neighborhood if :

$$d_E(BP_i, IP_j) \le E_t \qquad (2)$$

or

$$d_E(proj_y(BP_{ik}), IP_j) \le E_t \quad \text{for} \quad k = 1, 2 \ldots n \qquad (3)$$

is verified, where $d_E$ is the Euclidian distance, $proj_y(BP_{ik})$ is the vertical projection of $BP_i$ at the step $k$ on the segment whose height is $n$. Note that the equation 3 is checked only if equation 2 is not verified. Figure 3 shows both configurations of neighborhood verification.

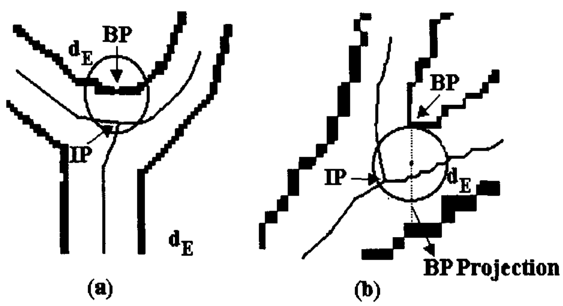


Figure 3: Distance verification between the $BP$ and $IP$ (a) Distance verified by equation 2 (b) Distance verified by equation 3

Depending on the local configurations of points, the segmentation path ($P_{seg}$) can be generated straight away from the $P_{skeleton}$ or being orthogonal to it. In the first case, the system has found particular points of class 1 and class 2, as described in Figure 2 and tries to link the sequence of $P_{skeleton}$ between these two points. The figure 4 shows an example where $P_{skeleton}$ (Figure 4b) is associated with the joining segments between $BPs$ and $IPs$, denoted ($P_{bp-ip}$) (Figure 4c) to form the final and correct segmentation path ($P_{seg}$) (Figure 4d). Thus, we can define the segmentation path: $P_{seg} = P_{skeleton} \bigcup P_{bp-ip}$

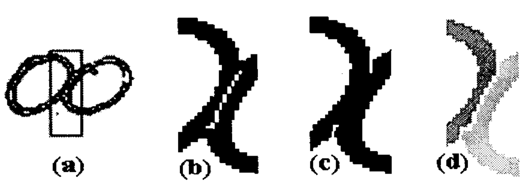


Figure 4: Particular conections: (a) Skeleton (b) $P_{skeleton}$ (c) $P_{bp-ip}$ (d) $P_{seg}$

In the second case, where no such configuration is found, but where it exists some $IPs$ and $BPs$, the segmentation path is orthogonal to $P_{skeleton}$. To determine the possible cuts around each $IP$, the algorithm performs the following tests :

1. If the considered segment of the skeleton is a stroke-end ($TP$ exists on the segment), then the cut is allowed if the segment length is significant;
2. The cut length should be minimum in the neighborhood of $IP$. To find the best position, the cut is evaluated in $2r_e$ pixels, where $r_e$ is the $IP$radius, for the minimum distance where a transition is encountered.

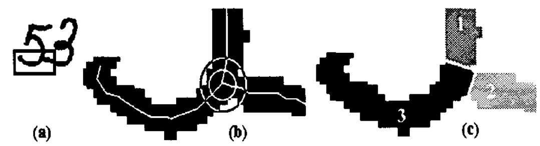


Figure 5: Orthogonal cuts: (a) Original image (b) Potencial region (c) Cuts realized

Figure 5b shows the specific regions where the search of the cut path is performed around the $IP$. For each pixel of the $P_{skeleton}$ included in the biggest circle, the orthogonal cut path is evaluated and compared to the previous minimum length already calculated. Finally, the algorithm performs the cut path for each segment around the $IP$. When the configuration does not verify the first test, the cut path is not performed as shown in Figure 5c for the segment 3.

In some cases, none $IP$ is detected in the image (see Figure 6), even if it exists a connection between two characters. To solve the lack of redundancy of $BPs$ and $IPs$, the algorithm enables cut path directly from $BPs$ in order to avoid segmentation errors.

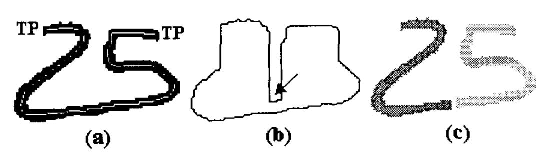


Figure 6: Lack of $IPs$: (a) Original image with the $TPs$ only (b) $BP$ from profile (c) Segmented image

Once all the segmentation paths are generated, the following task consists in determining the best character segmentation. Since our system provides over-segmented images, it seems important at this level to develop a strategy which naturally limits the number of hypotheses generated. Moreover, depending on the writing styles and the image quality, it is sometimes necessary to combine two (or more) paths in order to provide the correct segmentation.

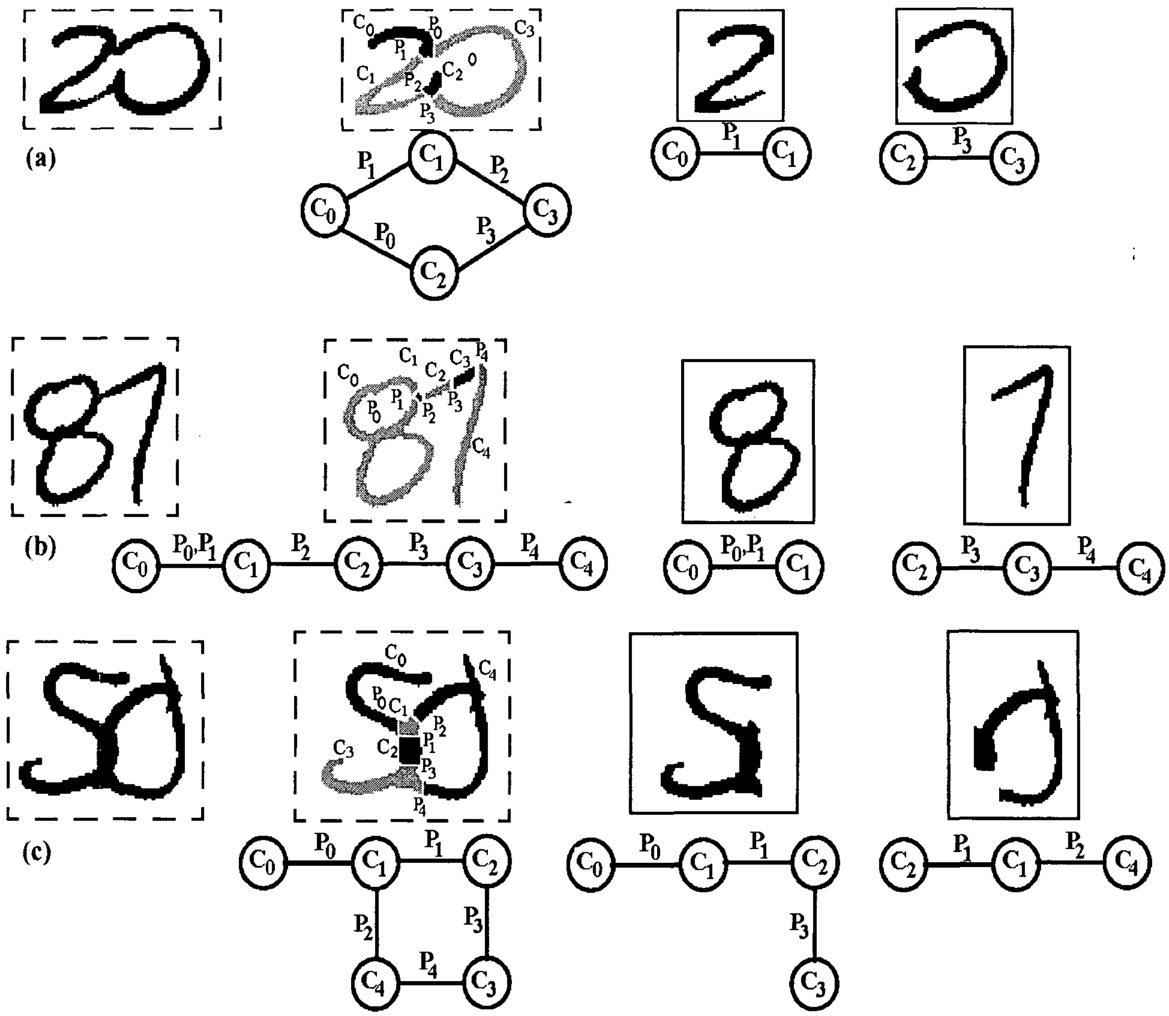


Figure 7: Segmentation hypotheses, initial graphs and linear sub-graphs

## 4 Management of the Segmentation

The approach that we developed is based on works founded in the literature where the stroke sequences are generated through a segmentation graph [8].But in our case, the strategy applied to generate all the segmentation hypotheses takes into account the initial graph, and not the linear graph which is obtained after some filter procedures.

In this way, the algorithm is able to solve a lot of segmentation configurations which are not solved by the initial approach. Figure 7 shows the segmentation hypotheses, where $C_i$ is the sub-component and $P_i$ is the segmentation path, as well as the initial graphs and the linear sub-graphs.

Basically, the algorithm works in two levels: In the first level, the initial graph is decomposed in linear sub-graphs that represent the segmentation hypotheses without a common sub-component for two entities (Figure 7b). Each linear sub-graph can be composed by a sequence of the $n$ sub-components, where $1 \leq n \leq 6$. The maximum value to $n$ was determined after we observe the behavior of the segmentation algorithm in our database.

In the second level, the linear sub-graphs generated at the first level are combined with each other in order to enable the duplication of the some sub-components. This partial duplication allows the algorithm to solve the superposition cases (Figure 7c).

## 5 Classifiers Combination

In the handwriting recognition context, the effect of a large variability in character representation is a difficult task to overcome. Moreover, the impact of the segmentation algorithm on the connected characters will accentuate this variability.Several combination approaches have been proposed in the litterature [6] [10] in order to improve efficiency and accuracy of the recognition module. We chose three of the most common methods encountered to evaluate the effects.

### 5.1 Cascade

This method is based on the complementarity between recognizers. Only the rejected images from the first classifier are submitted to the second classifier, and so on until the last one. Then, each classifier can be dedicated to a specific variability of the writing. Note that the ordering choice depends on the decreasing order of rejection rate for each classifier to achieve the best recognition rate.

### 5.2 Average

This approach provides an output vector which is the averaged sum of scores from the output-set of all the classifiers. This is a simple method which is often limited by its smoothing effects.

### 5.3 Product

It is a drastic method to combine classifier outputs. Even if it is well adapted for the fusion of decorelated classifiers, this approach is efficient when all classifiers provide a significant probability gradient between the maximum output value (theorically 1.0) and the others

## 6 Evaluation of the Segmentation

This kind of evaluation is often subjective. Proceeded manually, it depends on the operator's knowledge of the system while automatically, it depends on the training database of the classifiers. Since our system relies on a segmentation-based recognition approach, we chose the automatic evaluation of the segmentation module and we used the manual evaluation to estimate the possible highest segmentation rate.

The recognition module contains three Multi-Layer Perceptrons (MLPs), trained on three types of features : Concavity Measures ($k_1$) [9], Edge Maps ($k_2$) [2] and a mix Concavity - Contour ($k_3$). The training database contains 9,500 images (10 classes) of naturally isolated handwritten digits extracted from our laboratory database of 2,000 brazilian checks.The evaluation was runned with another set of the same database splited into two sub-bases : 900 images of connected digits (pairs and triples for the most of them) and 3,500 images of naturally isolated digits. The latter base was used to compare the effects of the segmentation module on the classification performances.With the manual verification of the segmentation, 98.5% of connected characters were correctly segmented. Most of the segmentation errors are caused by the lack of BPs in the neighborhood of the connected strokes (see Figure 8c).The automatic evaluation, expressed in Tables 1 and 2, shows the recognition results for both subsets (isolated and connected databases respectively) and for each strategy of combination. For these both experimentations, each classification mode proceeded the same rejection threshold.

The reliability rate (*Rel*) is defined by:

$$Rel = \frac{\text{Rec. rate}}{\text{Rec. rate} \quad + \quad \text{Err. rate}} \times 100 \qquad (4)$$

Table 1: Classifiers Performance - Isolated Database

| Classifier | *Rec %* | *Err %* | *Rej %* | *Rel %* |
|---|---|---|---|---|
| $k_1$ | 90.74 | 0.25 | 8.99 | 99.71 |
| $k_2$ | 90.05 | 2.98 | 6.95 | 96.78 |
| $k_3$ | 90.16 | 0.22 | 9.61 | 99.75 |
| Cascade | 98.52 | 0.09 | 1.38 | 99.90 |
| Average | 92.97 | 0.09 | 6.92 | 99.89 |
| Product | 91.68 | 0.09 | 8.21 | 99.89 |

Table 2: Classifiers Performance - Connected Database

| Classifier | *Rec %* | *Err %* | *Rej %* | *Rel %* |
|---|---|---|---|---|
| $k_1$ | 84.83 | 6.98 | 8.18 | 92.39 |
| $k_2$ | 83.18 | 13.29 | 3.15 | 86.21 |
| $k_3$ | 85.84 | 7.41 | 6.74 | 92.05 |
| Cascade | 95.24 | 2.14 | 2.61 | 97.80 |
| Average | 70.60 | 10.15 | 20.90 | 87.43 |
| Product | 68.74 | 8.40 | 24.20 | 89.11 |

Considering the subset of isolated digits, the three combination modes improve the performances of the classifiers ($k_1$, $k_2$ and $k_3$). The best result is performed by the Cascade with 98.52% of correct classification and 1.38% of rejected patterns. This is also the best approach for the subset of connected digits with 95.24% of correct classification and 2.61% of rejection. But with this subset, the classification rates achieved by the Average and Product modes are lower than the rates provided by the single recognizers. This difference shows the influence of the segmentation module when the classifiers are not trained with segmented digits.

The performance analysis of the connected characters subset lead us to define two categories of errors: segmentation errors and classification errors. We define a segmentation error as the best score configuration where one (or more) character is mis-segmented and provokes a misclassification (see Figure 8a). A classification error is the best score configuration where the segmentation is correct but one (or more) character is mis-classified (see Figure 8b). In both cases, we hope soon improve the reliability of the classification module by adding segmented characters in the training database.

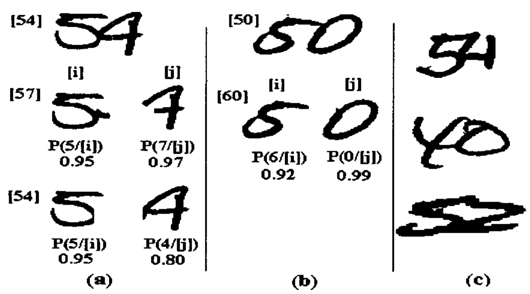


Figure 8: Cases of errors (a) Segmentation error (b) Recognition error (c) Cases not solved

This analysis shows the encouraging performances of our segmentation algorithm. With a limited number of rules, this technique is able to perform a correct segmentation in most cases of connected characters, even if the digits are strongly overlapped or skewed.

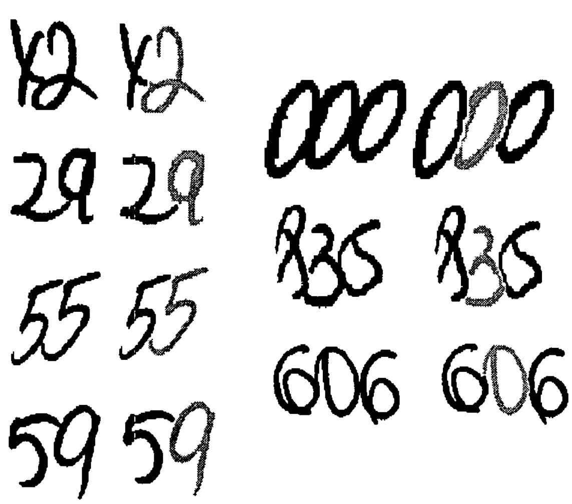

Figure 9: Correct segmentation in complex cases

To obtain the same results, current structural approaches need to use a lot of heuristics to consider the large variability of handwriting [5]. Moreover, any particular preprocessing technique is necessary as we find in other approaches such as windowing [4] [11] or histogram projection [13]. If we compare with algorithms only based on contour and profile features [3], our method is more accurate in case of strong connections. Figure 9 shows examples of particular configurations where segmentation is not obvious with classical methods.

## 7 Conclusion and Perspectives

We have presented a new segmentation-based recognition approach applied to unconstrained handwritten digits. The segmentation technique uses two sets of structural features to provide the possible segmentation paths, with any contextual information of the input image. Due to the small number of rules used in the cut path generation, the system is able to provide the correct segmentation in most cases of connected characters. The first results show that we can easily improve the recognition rate by considering the segmentation outputs in the training step of the recognizer. The cascade strategy will be used to yield this new database.

The next studies we plan to implement in this system deal with the combination of segmentation information in the recognition module in order to perform a real segmentation-based recognition system. Moreover, we will focus on a new recognition system based on product combination where various types of features will be used to improve the rejection rate.

## References


[1] B.K.Jang and R.T.Chim. Reconstructable parallel thinning. In C.Y.Suen and P.S.P Wang, editors, *Thinning Methodologies for pattern recognition*, pages 181–217. World Scientific, 1994.

[2] Y. C. Chim, A.A. Kassim, and Y.Ibrahim. Dual classifier system for handprinted alphanumeric character recognition. *Pattern Analysis and Application*, 1:155–162, 1998.

[3] E.Lethelier, M.Leroux, and M.Gilloux. An automatic reading system for handwritten numeral amounts on french checks. In *3$^0$ ICDAR*, pages 92–97, Montreal - Canada, 1995.

[4] G.Martin, M.Rashid, and J.Pittman. Integrated segmentation and recognition through exhaustive scans or learned saccadic jumps. In I.Guyon and P.S.P Wang, editors, *Advances in pattern recognition systems using network technologies*, pages 187–203. World Scientific, 1993.

[5] H.Nishida and S.Mori. A model-based split-and-merge method for character string recognition. In P.S.P Wang, editor, *Document Image Analysis*, pages 209–226. World Scientific, 1994.

[6] J.Kittler, M.Hatef, R.Duin, and J.Matas. On combining classifiers. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[7] E. Lecolinet and J.P. Crettez. A grapheme-based segmentation technique for cursive script recognition. In *$1^0$ ICDAR*, pages 740–749, St.Malo - France, 1991.

[8] M. Leroux, E. Lethelier, M. Gilloux, and B. Lemarie. Automatic reading of handwritten amounts on french checks. In S.Impedovo et al, editor, *International Journal of Pattern Recognition and Artificial Intelligence*, pages 619–637. World Scientific, 1997.

[9] L.Heutte, P.Pereira, O.Bougeois, J.Moreau, B.Plessis, P.Courtellemont, and Y.Lecourtier. Multi-bank check recognition system:consideration on the numeral amount recognition system. In *International Journal of Pattern Recognition and Artificial Intelligence*, pages 595–618. 1997.

[10] L.Lam and C.Y.Suen. Optimal combinations of pattern classifiers. *Pattern Recognition Letters*, 16(9):945–954, 1995.

[11] O. Matan and C.J.C.Burges. Recognizing overlapping hand-printed characters by centered-objects integrated segmentation and recognition. In *IJCNN*, pages 504–511, 1991.

[12] M.Shridhar and A.Badreldin. Recognition of isolated and simply connected handwritten numerals. *Pattern Recognition*, 19(1):1–12, 1986.

[13] R.Fenrich. Segmentation of automatically located handwritten words. In *$2^{nd}$ International Workshop on Frontiers in Handwriting Recognition*, pages 33–44, Chateau de Bonas - France, 1991.

[14] S.Choi and I. Oh. A segmentation-free recognition of two touching numerals using neural networks. In *$5^0$ ICDAR*, pages 253–256, Bangalore - India, 1999.

[15] X. Wang, V. Govindaraju, and S. Srihari. Holistic recognition of touching digits. In *$6^0$ IWFHR*, pages 295–303, 1998.

[16] Z.Shi and V. Govindaraju. Segmentation and recognition of connected handwritten numeral strings. In *Progress in Handwriting Recognition*, pages 515–518. World Scientific, 1996.