A structural/statistical feature based vector for handwritten character recognition

L. Heutte a,*, T. Paquet a, J.V. Moreau b, Y. Lecourtier a, C. Olivier a

a PSI-La3i, UFR des Sciences et Techniques, Université de Rouen, F76821 Mont-Saint-Aignan Cedex, France

b Matra Systèmes & Information, Velizy-Villacoublay, France

Received 30 March 1996; revised 1 December 1997

Abstract

This paper describes the application of structural features to the statistical recognition of handwritten characters. It has been demonstrated that a complete description of the characters, based on the combination of seven different families of features, can be achieved and that the same general-purpose structural/statistical feature based vector thus defined proves efficient and robust on different categories of handwritten characters such as digits, uppercase letters and graphemes. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Handwritten character recognition; Structural features; Statistical features; Feature definition; Missing values; Statistical classification

1. Introduction

In the field of handwriting recognition, it is now agreed that a single feature extraction method and a single classification algorithm generally cannot yield a very low error rate, even if some highly reliable systems have been developed recently, especially for handwritten numerals (Smith et al., 1994; Shridhar et al., 1997). Due to the large variety of the available feature extraction methods (Trier et al., 1996), many researchers have turned towards the use of several feature extractors with more complex structures of classification, such as multistage classification schemes (Rahman and Fairhurst, 1997; Cao et al., 1995) or "parallel" combination of multiple classifiers (Xu et al., 1992; Sabourin et al., 1993). Even though the results published in these papers are of great interest, some problems still remain: "How many classifiers and what kind of classifiers should be used? For each classifier, what types of features should be chosen?". Moreover, this involves multiple tiresome learning steps for both the chosen classifiers and the combination rules.

The authors believe that there have not been sufficient efforts made towards the use of several feature extractors in a same structure of classification (e.g., in a one-shot classifier). Three major interests can however justify such an approach: (i) the use of several types of features still ensures an accurate description of the characters; (ii) the use of a single classifier preserves fast and flexible learning; (iii) the tedious tuning of combination rules is avoided.

Based on this idea, the authors investigated the possibility of combining both structural and statistical features for the recognition of handwritten char-
acters using a single statistical classifier. The key point for doing so lies in the definition of a mapping of the structural features (qualitative features) into a fixed-size feature space to feed the statistical classifier. This leads to a 124-variable feature vector, called the Structural/Statistical Feature Based Vector (SSFBV). The statistical classifier is based on a linear discrimination technique. To show that the SSFBV provides a robust description of handwritten characters, the recognition system has been tested over large-scale samples including handwritten well-segmented numerals and handprinted characters as well as graphemes which are the most current patterns generated by an handwritten cursive word segmentation (graphemes can be either well-segmented, over-segmented or under-segmented characters). This recognition scheme has proved to be efficient not only on numerals (compared to some recent published works) but also on more difficult recognition problems such as those cited above.

The paper is organized as follows. Section 2 presents the definition of the mapping retained: parametrization of the structural features and selection of the replacement rules for missing values. The seven families of features that make up the SSFBV are described in Section 3 as well as their extraction process. The various experiments conducted are presented in Section 4 including a brief description of the statistical classifier and the collection of data for learning and testing. Finally, some conclusions are drawn in Section 5.

2. From shapes to features: the structural/statistical dilemma

Handwritten character recognition systems typically involve two steps: feature extraction in which the patterns are represented by a set of features and classification in which decision rules for separating pattern classes are defined.

Features can be broadly classified into two different categories (Govindan and Shivaprasad, 1990): structural features (like strokes and bays in various directions, end points, intersections of line segments, loops and stroke relations, . . .) and statistical features (derived from the statistical distribution of points like zoning, moments, $n$-tuples, characteristic loci, . . .). Structural and statistical features appear to be complementary in that they highlight different properties of the characters. In order to cover the within-class and between-class variabilities of the patterns, the combination of the two approaches has been retained to provide an efficient and complete description of the characters.

In the same way, there are broadly two main approaches for classification (Jain, 1990): the statistical approach consisting in representing a pattern as an ordered, fixed-length list of numerical values and the structural approach describing the pattern as an unordered, variable-length list of simple shapes. The statistical approach relies on firmly established elements of statistical decision theory (Fukunaga, 1990), even though viewing a pattern in an $n$-dimensional space is rather difficult. On the contrary, the structural approach is intuitively appealing because it appears closer to human recognition strategy (Bunke and Sanfeliu, 1990). Unfortunately, this approach is usually difficult to implement in a fast, trainable and robust way over a large variety of shapes.

As a structural classifier is naturally well-suited to the use of structural features but cannot easily handle statistical features, we have chosen the frame of statistical classification to investigate the combination of both structural and statistical features in a one-shot classifier. Now, in the field of statistical classification, a pattern must be represented by a fixed-length list of $n$ numerical variables called the feature vector. Therefore, the set of features (either statistical or structural) describing a pattern must be constrained to map the feature vector. This mapping consists in a parametrization of each feature, i.e. defining numerical parameters for each feature.

The parametrization of statistical features is naturally trivial. On the contrary, structural features must be described by a set of numerical parameters which require some ‘‘continuity’’ properties, e.g. a small change in perceived shape should always result in a small displacement in parameter space (Baird, 1988). For example, suppose that $F$ represents a structural feature extracted on the pattern (such as an end-point, a hole or a concave arc . . .). Most often, the parametrization of such features is obtained using a zoning-like technique (Trier et al., 1996): a fixed $n \times m$ grid is superimposed on the pattern image and the feature is searched for in each of the $n \times m$
zones, thus giving a binary feature vector of length \( n \times m \). Unfortunately, this leads to a loss of information on the feature position as well as to a loss of continuity in the parameter space: small variations in the pattern can lead to large variations in the parameter space (Cao et al., 1995). To cope with this, the authors have decided to parametrize the structural features by continuous numerical variables, such as the \( X-Y \) position' of \( F \). Indeed, these variables are calculated relative to the center of gravity of the pattern and normalized in \([-1; +1]\) according to the width or the height of the pattern bounding box. This parametrization allows to locate accurately the structural features and to respect the principle of continuity.

In the field of handwritten character recognition, the large variety of shapes which can be found in a same class of character, pose the problem of possible multiplicity of a same structural feature in the pattern (as shown for example in Fig. 1, for the end-points).

This is a priori incompatible with the necessity of working in a fixed \( n \)-dimensional feature space, but can be overcome by over-sizing the feature space, i.e., by reserving a maximum number of occurrences of the ‘’position’’ variables for example. The choice of the maximum number of ‘’position’’ variables to be reserved for each kind of feature is done before the learning phase by simply searching and keeping the maximum number of occurrences of each feature whatever the classes of characters. However, the over-dimensioning of the feature vector requires the definition of some decision rules, when the number of detected features is not the expected one. Therefore, two replacement rules are applied, according to whether there are fewer or more features than expected:

1. Replacement rule when there are fewer features than expected. Those \( X-Y \) position variables corresponding to the missing features cannot be set to 0 because it would mean that these features are exactly on the center of gravity of the character (according to the normalization described above). However, a numerical value must be assigned to these \( X-Y \) position variables. Several simple and inexpensive techniques for handling missing values have been described in (Dixon, 1979). The only configuration which has provided good results in our own experiment consists in assigning to the missing \( X-Y \) position variables the mean that these variables take on the learning base (Heutte, 1994). In the feature space, this means to replacing those feature vectors by the center of gravity of the clouds of points.

2. Replacement rule when there are more features than expected. In this case, it is assumed that the extra features come from noise in the character, since the maximum number of occurrences for each kind of features is known after the learning phase. Therefore, in order to avoid that these extra features might not be taken into account (for example, by directly removing them), the pairs of nearest features are merged by deriving the average of their positions.

This mapping has proved to be efficient to overcome the classical dilemma of combining both statistical and structural features in the same feature space. This will thus allow to handle the large variability of handwritten patterns (digits, uppercase letters and graphemes), using the same general-purpose structural/statistical feature based vector, detailed in the following section.

3. The Structural/Statistical Feature Based Vector (SSFBV)

The SSFBV has been built in order to provide a wide range of identification clues over a large vari-
ety of handwriting recognition systems, including digit, uppercase letter and grapheme recognition. The feature extraction methodology lies in the combination of seven families of features selected from preliminary work on pseudo-character recognition (Heutte et al., 1993). The resulting SSFBV provides a complete description of the characters, mainly thanks to the combination of structural and statistical features that highlight global and local properties. The seven families of features presented below are split into two categories according to whether their parametrization leads to retain position variables (local features) or not (global features).

3.1. Global features

Global properties of the patterns can be highlighted, thanks to three different families: invariant moments, projections and profiles, as presented below. Although the projection and profile families allow to describe structural properties of the patterns, the mapping of these characteristics into the feature space does not require to manage missing values.

3.1.1. Invariant moments

The invariant moments used have been proposed in (Hu, 1962). These seven moments are well-known to be invariant to position, size and orientation of the character. They are pure statistical measures of the pixel distribution around the center of gravity of the character and allow to capture the global character shape information. They are the following:

\[ \phi(1) = (\mu_{20} + \mu_{02}) \],
\[ \phi(2) = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \]
\[ \phi(3) = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2, \]
\[ \phi(4) = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2, \]
\[ \phi(5) = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - 3\mu_{03})(\mu_{21} + \mu_{03})^3, \]
\[ \phi(6) = (\mu_{20} - \mu_{02})(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2 + 4\mu_{11}(\mu_{30} + \mu_{12}) \times (\mu_{21} + \mu_{03}), \]
\[ \phi(7) = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})^3 + 3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^3 \]

where

\[ \mu_{pq} = \frac{1}{N} \left[ \sum_{i=1}^{N} (x_i - \bar{x})^p (y_i - \bar{y})^q \right]^{(p+q)/2+1} \]
\[ \cdot \sum_{i=1}^{N} (x_i - \bar{x})^p (y_i - \bar{y})^q, \]

\[ \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i, \]

and \( N \) = total number of black pixels \((x_i, y_i)\) in the image.

3.1.2. Projections

These features are derived from histograms of horizontal and vertical projections of black pixels in some particular areas of the character. They are extracted from the normalized image of the character so as to obtain normalized histograms of black pixels both on the X-axis as well as on the Y-axis. In order to locate the larger strokes of the character, relative maxima of the histograms could have been extracted. However, extracting maxima from histograms is quite difficult when the number of maxima is not known a priori. One way to tackle this problem is to derive the cumulative histogram of the original one, assuming that each important gap on the Y-axis of the cumulative histogram corresponds to a relative maximum on the Y-axis of the original histogram. Yet, when there is a gap on the Y-axis of the cumulative histogram, the corresponding abscissas are closer than in the case of a slow variation on the Y-axis. Therefore, dividing the Y-axis in enough equal parts and storing as features the corresponding abscissas allows to get information about the location of some structural properties of the character like its larger
strokes. The feature extraction principle is illustrated on a basic example in Fig. 2.

In the present case, dividing the Y-axis in 11 parts – 10 abscissas are thus retained as features from each cumulative histogram – has been found experimentally to provide the most precise description of the characters.

3.1.3. Profiles

The profile based features are partly inspired by (Shridhar and Badreldin, 1984), in which the authors use these features, extracted from left and right profiles only, for the segmentation/recognition of handwritten connected numerals. However, for handwritten character recognition, left and right profiles are not sufficient to sharply describe character variabilities. Top and bottom profiles are therefore added to our description of the character.

The four profiles are derived from normalized characters, in order to obtain normalized profiles both on the X-axis and on the Y-axis. Processing the left profile, for example, of a character consists in scanning each line of the normalized character from top to bottom and from left to right, and in storing the first found black pixel of the character. This profile is called the raw left profile of the character. Raw top, bottom and right profiles are obtained a similar way (see Fig. 3). Features derived from profiles have been selected in order to give some information about the smoothness of the character and to describe the character in terms of width and height at particular locations of the character.

The smoothness of the character is obtained from the first difference profiles (first order derivative of the raw profiles). From each first difference profile, the maximum amplitude is extracted and normalized in [0; +1] according to the height (for top and bottom profiles) or the width of the normalized character (for left and right profiles). These four variables thus correspond to some information about the smoothness of the four profiles since a near to 0 amplitude means that there is no “accident” in the...
profile, whereas a large amplitude points out a broken profile (see Fig. 3). The description of the character in terms of width (respectively height) at particular locations is derived from the difference between the left and the right raw profiles (respectively bottom and top) at three locations: 1/5, 1/2 and 4/5 of the height (respectively the width) of the normalized character bounding box. These variables are normalized in [0; +1] according to the width (respectively the height) of the bounding box. Complementary to the intersection locations (1/3 and 2/3 of the height), the width locations (1/5 and 4/5) privilege the extremities of the character.

3.2. Local features

Four families are concerned that allow to take into account complementary points of view of the pattern structure: intersections with straight lines, holes and concave arcs, extrema and, end points and junctions. They all require a precise positioning in the pattern bounding box and have been parametrized so as to ensure continuity in the feature space. The mapping of these features therefore requires an over-dimensioning of the feature vector as presented in Section 2 to handle position variables.

3.2.1. Intersections with straight lines

Number and position of intersections between the character and straight lines have been retained as features.

Since distortions on characters have usually higher amplitudes on the horizontal axis than on the vertical one, the number of intersections with a horizontal straight line is more stable than the one with a vertical straight line. Therefore, two horizontal straight lines and only one vertical have been chosen to describe the character in terms of intersection based features. The two horizontal straight lines are located at the first third and the second third of the height of the character, thus providing a good description of the lower and higher parts of the character. The vertical straight line goes through the center of gravity of the character, in order to obtain a more stable location of the intersections for each class of characters (see Fig. 4). Since straight line locations are fixed a priori, only the X-position of the intersections with the horizontal straight lines and the Y-position of the intersections with the vertical straight line are retained as location variables.

3.2.2. Holes and concave arcs

These features are extracted from the representation of the character by its polygonized contours. Two steps are needed for the extraction of these features: the first one is the polygonization of both the external boundaries and the internal boundaries of the character in order to describe the character by an ordered list of vertices, the second one is the extraction of the features from this particular representation. The retained variables correspond to: (1) number, perimeter and location of concave arcs and (2) number and location of holes. Therefore polygonization based features enable to get from the character some local properties (concave arcs) and some global properties (holes). Some examples are given in Fig. 5.

Concave arcs are only extracted from the ordered list of vertices corresponding to the external boundaries, and are obtained by calculating the interior angle between two successive segments. If this angle is lower than 1, then these two successive segments form a concave vertex and the process goes on until an angle superior to 1 is found. A concave arc is thus

---

Fig. 4. Intersection with straight lines based features. (a) 1st horizontal (1/3 height); (b) 2nd horizontal (2/3 height); (c) vertical (going through the center of gravity)

Fig. 5. Holes (H) and concaves arcs (CA) from polygonized contours
defined as an uninterrupted sequence of concave vertices. The perimeter of a concave arc is obtained by adding the length of the segments and its position is found by calculating the barycenter of the vertices.

Holes are directly defined by the ordered lists of vertices corresponding to the internal boundaries. The position of a hole is obtained by calculating the barycenter of its vertices.

3.2.3. Extrema

This family of features includes top, bottom, left and right extrema of the character. The retained variables correspond to number and \( X - Y \) position of the extrema.

The extrema extraction algorithm depends on the kind of extrema to be extracted. However, only the top extrema extraction is described here, since the extraction principle for the bottom, left and right extrema is similar (only the search direction changes).

The top extrema extraction algorithm consists in searching from top to bottom of the character bounding box the black pixels which have no 8-connection with upper pixels. In order to speed up and to facilitate the search, the character is first encoded in terms of its horizontal run lengths (for the extraction of top and bottom extrema) or its vertical run lengths (for left and right extrema). A top extremum is then found when a run-length has no 8-connection with an upper run-length. A filtering step allows finally to keep only the extrema coming from larger run-lengths.

Extrema based features enable to get some information about the convex envelope of the character (see Fig. 6), and are thus complementary to those extracted from polygonized boundaries (i.e., the concave arcs).

3.2.4. End points and junctions

These pure structural features (end points, \( Y \) and \( X \)-junctions) are extracted on the skeletonized representation of the character – a beardless smoothed 8-connected skeleton (see Fig. 7) – and the variables retained are the number and the \( X - Y \) position of these features. The extraction principle of these structural features consists in trying to apply on each black pixel of the skeleton a collection of \( 3 \times 3 \) templates previously defined for each kind of features.

For the extraction of \( Y \)-junctions, a collection of twelve \( 3 \times 3 \) templates, corresponding to each configuration of an \( Y \)-junction, is applied on each black pixel of the skeleton. In order to speed up the detection of these features, the number of black pixels in the \( 3 \times 3 \) neighbourhood of the tested black pixel must first contain 4 or 5 black pixels.

Extraction of \( X \)-junctions is rather different because two levels are used to detect an \( X \)-junction. The first one is based on the application of two particular templates, in order to detect what is called a real \( X \)-junction. Since the skeletonization process tends to split a real \( X \)-junction into two \( Y \)-junctions (thus generating a false stroke), a second level is applied to merge the pairs of nearest \( Y \)-junctions into \( X \)-junctions (see Fig. 7).

A black pixel is considered to be an end point if there is only one black pixel in its \( 3 \times 3 \) neighborhood.

Extrema based features enable to get some information about the convex envelope of the character (see Fig. 6), and are thus complementary to those extracted from polygonized boundaries (i.e., the concave arcs).

3.2.4. End points and junctions

These pure structural features (end points, \( Y \) and \( X \)-junctions) are extracted on the skeletonized representation of the character – a beardless smoothed 8-connected skeleton (see Fig. 7) – and the variables retained are the number and the \( X - Y \) position of these features. The extraction principle of these structural features consists in trying to apply on each black pixel of the skeleton a collection of \( 3 \times 3 \) templates previously defined for each kind of features.

For the extraction of \( Y \)-junctions, a collection of twelve \( 3 \times 3 \) templates, corresponding to each configuration of an \( Y \)-junction, is applied on each black pixel of the skeleton. In order to speed up the detection of these features, the number of black pixels in the \( 3 \times 3 \) neighbourhood of the tested black pixel must first contain 4 or 5 black pixels.

Extraction of \( X \)-junctions is rather different because two levels are used to detect an \( X \)-junction. The first one is based on the application of two particular templates, in order to detect what is called a real \( X \)-junction. Since the skeletonization process tends to split a real \( X \)-junction into two \( Y \)-junctions (thus generating a false stroke), a second level is applied to merge the pairs of nearest \( Y \)-junctions into \( X \)-junctions (see Fig. 7).

A black pixel is considered to be an end point if there is only one black pixel in its \( 3 \times 3 \) neighborhood.
Overall, the SSFBV is made of 124 variables, distributed in the seven families of features as follows: 7 invariant moments; 20 for the horizontal and vertical projections; 10 for the top, bottom, left and right profiles; 15 for the intersections with horizontal and vertical straight lines; 21 for the holes and concave arcs; 30 for the top, bottom, left and right extrema; 21 for the end points and junctions. In addition, variables are normalized either in \([-1;+1]\) if they are X–Y position variables or invariant moments or in \([0;+1]\) if they are “number of feature” variables, projection and profile based variables or perimeter of concave arcs.

4. Experiments

The efficiency and robustness of the SSFBV have been tested over a wide range of handwritten character recognition problems, using the same recognition scheme based on a linear discrimination technique. Learning and recognition algorithms are briefly presented, followed by the description of the databases used for the experiments; experimental results are then presented and analyzed.

4.1. Learning and recognition

The present character recognition system is divided into two different stages: the learning stage and the decision stage. Input of both these two stages is the character image from which the 124-variable feature vector described above is derived.

The learning stage is processed as follows. Let \(\omega_i\) be one of the \(K\) character classes. Let \(C(\omega_i)\) be a set of input vectors derived from a sample of characters belonging to \(\omega_i\). For each pair of clusters \((C(\omega_i), C(\omega_j))\), a linear discrimination technique is performed to derive a between-cluster hyperplane. The \(K\)-class problem is thus converted into \([K(K-1)]/2\) 2-class sub-problems of discriminating each possible pair of classes (Devijver and Kittler, 1982). For any pair of classes \(\omega_i\) and \(\omega_j\), a hyperplane is calculated. Each hyperplane partitions the feature space into two areas: one area clusters the points located on the same side as \(\omega_i\) in comparison with the hyperplane, the other area clusters the points located on the same side as \(\omega_j\). The output of the learning stage is finally a set of hyperplanes corresponding to the separation of each pair of classes.

The decision algorithm is applied to each character as follows. The feature vector associated to the character is first derived. Then its position is examined with respect to each hyperplane separating a pair of classes. If one class is better than the others, a confidence level is associated to this class with respect to the distance between the chosen class and the second best class. All the classes with a non-zero confidence level with respect to the chosen class are also kept in the list of possible solutions. The output of the recognition stage is thus an ordered list of character classes associated to their decreasing confidence level.

The use of a linear discrimination based classifier has two advantages over other statistical approaches in practice. The first one is that, unlike the non-parametric approaches (like k-NN or Parzen Window based classifiers for example), this classifier does not require large memory capacity and computation time during the decision stage. The second one is that this classifier is less sensitive to correlation between features than a Bayesian classifier such as the maximum-likelihood classifier for example (Heutte, 1994). This last point is important in practice since the efficiency of the SSFBV relies mainly on the complementarity of the seven families of features which of course cannot guaranty a complete decorrelation in the parameter space.

4.2. Data collection for learning and testing

The experiments have been conducted on three different categories of handwritten characters: digits, uppercase letters and graphemes. These data have been collected without any constraint on the writing style or writing instrument.

The numeral samples have been extracted from the NIST Special Database 3 which contains a total number of 223,125 digits representing 2100 writers. For the experiment reported here, an arbitrarily selected subset of 106,707 digits, evenly distributed in the 10 natural classes from ‘0’ to ‘9’, has been partitioned into the two following non-overlapping sets: (1) training set, 53,383 digits and (2) test set,
53,324 digits. Examples of the digit samples are given in Fig. 8a.

The uppercase letter samples have been collected from the dead letter envelopes by the U.S. Postal Services at different locations. Among a total number of 15,960 uppercase letter samples, distributed in the 26 natural classes from ‘A’ to ‘Z’, 12,765 samples have been used as training data (i.e., 80% of the full database) without overlapping with 3195 samples used as testing data (i.e., the remaining 20%). Fig. 8b shows examples of the uppercase letter samples.

![Examples of digitized data](image)

Fig. 8. Examples of digitized data (for display purposes, normalized to a fixed size). (a) Digits (10 classes); (b) uppercase letters (26 classes); (c) graphemes (72 classes).
The grapheme samples have been generated by a handwritten cursive word segmentation performed on address word images provided by the U.S. Postal Services (Plessis et al., 1993). Since graphemes can either correspond to only one character (succeeded segmentation) or correspond to a part of a character (over-segmentation) or to more than one character (under-segmentation), 72 classes of graphemes have been retained which cover both the variety of graphemes output by the segmentation and variances in writing styles. These classes include: 52 well-segmented lowercase letters (two allographs per letter), 9 uppercase letters whose shape is not included in the 52 lowercase letters, 8 graphemes corresponding to the most encountered over-segmented characters and 3 graphemes corresponding to the two most frequently never-segmented characters (double ‘t’, double ‘o’, …). Examples of grapheme samples are shown in Fig. 8c. The full database contains 26,382 graphemes and has been partitioned into the two following non-overlapping subsets: (1) training set, 21,108 graphemes (i.e., 80% of the total number of samples) and (2) test set, 5,274 graphemes (i.e., the remaining 20%).

Finally, it must be noted that the choice of samples to be in the training set or in the test set was arbitrary for the three databases, and that no ambiguous, ill-segmented or misclassified character has been removed from either the training sets or the test sets.

4.3. Experimental results

The overall experimental results obtained from the testing sets are listed in Table 1, for the three kinds of handwritten characters. TopM (M = 1,2,3,4) stands for the percentage of presence (recognition) or absence (substitution) of the correct solution among the M best rated. For convenience, only the ‘recognition vs. substitution’ index has been reported in Tables 2–4. However, the particularity of the classifier used leads us to define an ‘‘indecision rate’’ according to the following relation:

\[ \text{recognition(\%)} + \text{substitution(\%)} + \text{indecision(\%)} = 100\% \] (1)

Indeed, as mentioned in (Devijver and Kittler, 1982), one drawback of the decision algorithm presented in Section 4.1 is that some decisions may be undefined or ambiguous between two or more classes (i.e., the same confidence level is attributed to each of the classes involved). Therefore, TopM indecision rate represents the percentage of characters for which M + 1 classes at least (including or not the correct solution) have been proposed with equal confidence levels.

From Table 1, we can see on the digit and uppercase letter testing sets that the system can achieve a Top1 recognition rate of 97.84% and 97.27%, respectively and a recognition rate of more than 98.50% for both testing sets, if we consider the
correct solution among the 2 best rated. On the other hand, but not surprisingly, the Top1 performance of the system on the grapheme testing set is significantly lower: 65.54% recognition for 9.42% substitution. Although this result seems to be poor compared to those obtained on digits and uppercase letters, it must be related to the large number of classes involved (i.e., 72 classes). Indeed, the larger the number of classes, the larger the number of undefined decision regions. This explains that 25% of the decisions are undefined at the Top1 level. Besides, near 86% of the total number of graphemes are well-recognized if we consider the correct solution among the two best rated: when replaced in the context of handwritten cursive word recognition, this last result enables to generate more frequently the correct word hypothesis (Heutte, 1994).

The overall “recognition rate vs. substitution rate” can give a broad idea of the SSFBV performance. However, this is not the only index for evaluating the efficiency and robustness of classification: performance can be indexed by the confidence level which is used as a threshold for rejection. Such performances are listed in Tables 2–4 (with a graphic representation of the results on the right) for the three kinds of handwritten characters. Recog., Subst., Reject. and Reliab. are abbreviations of recognition, substitution, rejection and reliability rates, respectively. The reliability rate is defined by

\[ \text{Reliab.} = \frac{\text{Recog.}}{100\% - \text{Reject.}} = \frac{\text{Recog.}}{\text{Recog.} + \text{Subst.}}. \tag{2} \]

Since, as mentioned above, there is a trade-off between the recognition performance and the number of classes involved in the recognition process, the highest performance is obviously achieved on the digit and uppercase letter testing sets: with a rejection rate of 8.9% of the total number of digits (respectively 16.2% of the total number of uppercase letters), the system can achieve a substitution rate of 0.25% (resp. 0.40%) while still preserving a recognition rate as high as 90.8% (resp. 83.4%). On the other hand, the lowest substitution rate achieved on
the grapheme testing set is only 2.6%, for a recognition rate of 52.8%. However, if we take into account the wide range of shapes to be recognized in the grapheme case, this last result appears to be good.

Finally, results given in Tables 2–4 demonstrate that the same SSFBV allows to achieve high performance on the three kinds of patterns, i.e., more than 99.5% of reliability on digits and uppercase letters and 95.3% on graphemes. These last results led to the two following comments. The first one is that, even if the performance achieved on digits is broadly comparable with other recent work using the same or similar databases (Xu et al., 1992; Sabourin et al., 1993; Cao et al., 1995; Rahman and Fairhurst, 1997), the results reported in this study have been obtained by means of a simple and single structure of classification. This tends to prove that the combination of multiple classifiers may not be the only solution to yield a low error rate. The second comment is that the results for the uppercase letter set show only a small deterioration, compared to those obtained on digits, even though the class set has increased from 10 to 26 classes. This is emphasized by the performance achieved on the grapheme set (72 classes), which is comparable or superior to the one obtained on segmented characters in handwritten cursive word recognition problems (Srihari, 1993; Chen et al., 1994; Kimura et al., 1997). This last remark suggests that the combination of the seven families of features may be more discriminatory for larger class problems than other reported approaches, while keeping little advantage for smaller class sets.

5. Conclusions

We have presented in this paper a new feature vector for handwritten character recognition, which combines the strengths of both statistical and structural feature extractors. Thanks to a combination of seven complementary families of features (ranging from pure structural to pure statistical and including both local and global features), a complete description of the characters can be achieved, thus providing a wide range of identification clues. In order to feed a statistical classifier based on a linear discrimination technique, a parametrization of the structural features has been proposed and allows to get rid of the most common-used transformation of these features into binary values.

Even if this parametrization results in an over-dimensioning of the feature vector and needs to define some replacement rules for missing data, good performance can be achieved. The highest obtained are the following ones:

(i) 0.25% substitution for 90.8% recognition on 53,324 handwritten well-segmented digits extracted from the NIST Database,
(ii) 0.40% substitution for 83.4% recognition on 3195 uppercase letters collected from US dead letter envelopes,
(iii) 2.59% substitution for 52.8% recognition on 5274 graphemes generated by an handwritten cursive word segmentation performed on US address word images.

Finally, we have thus demonstrated that, by means of a single structure of classification, high performance of recognition can be achieved on different kinds of handwritten patterns using the same general-purpose feature vector, provided that this feature vector combines the strengths of both statistical and structural feature extractors.

References

Heutte, L., Moreau, J.V., Lecourtier, Y, 1993. A new feature extraction strategy for handwritten pseudo-character recogni-


