HBF49 feature set: A first unified baseline for online symbol recognition

Adrien Delaye, Eric Anquetil

▶ **To cite this version:**

HAL Id: hal-00933509

https://hal.archives-ouvertes.fr/hal-00933509

Submitted on 2 Feb 2014

HBF49 feature set: a first unified baseline for online symbol recognition

Adrien Delaye[a,*], Eric Anquetil[a]

[a]*IRISA, Rennes, FRANCE*

Abstract

As the rise of pen-enabled interfaces is accompanied with an increased number of techniques for recognition of pen-based input, recent trends in symbol recognition show an escalation in systems complexity (number of features, classifiers combination) or the over-specialization of systems to specific datasets or applications. In spite of the importance of representation space in feature-based methods, few works focus on the design of feature sets adapted to a large variety of symbols, and no universal representation space was proposed as a benchmarking reference. We introduce in this work **HBF49**, a unique set of features for the representation of hand-drawn symbols to be used as a reference for evaluation of symbol recognition systems. An empirical constructive approach is adopted for designing this set of 49 simple features, able to handle a large diversity of symbols in various experimental contexts. An original effort is made for guaranteeing transparency of features design and reproducibility of experiments. We demonstrate that using off-the-shelf statistical classifiers, the **HBF49** representation performs comparably or better than state-of-the-art results reported on 8 databases of hand-drawn objects. We also obtain a good recognition performance for user-defined gestures that further attests the ability of **HBF49** to deal with a great variety of symbols.

Keywords: feature extraction, online sketch recognition, graphics recognition, symbol recognition, pen-based gesture, benchmarking

1. Introduction

Automatic recognition of electronic ink has been a very popular topic and has lead to many applications available on the market. A large amount of research work was conducted for designing highly efficient recognition methods dedicated to online handwritten characters, cursive words and sentences (see [1] for a review), Asian characters [2, 3], scientific notations [4, 5], diagramming and graphics [6], sketched symbols [7, 8, 9], pen-based gestural commands [10, 11].

Besides handwriting text recognition systems, methods for recognition of isolated hand-drawn patterns have flourished in the recent years. Interestingly, methods applied in different application fields have focused on different properties of the patterns. Image-based approaches, where the pattern is represented as a purely visual signal, have been widely adopted for symbol recognition and sketch interpretation [8, 9, 6, 12], as well as for isolated digit or character recognition [1]. The online nature of input signal however contains rich information about the dynamics of the drawing (order and number of strokes, writing direction, speed, pressure) that can largely benefit the recognition of some categories of patterns: for example, the handwriting of digits or characters involves a dynamic process that presents some stability, complementary to the visual one. Several

methods thus include a description of dynamic information in addition to the visual one [13, 14, 15, 10]. Alternately, in the case of *gesture* recognition, the dynamic information can be of primary importance. As some hand-held devices support pen-based or touch-based gestures that are expected to be realized without looking at the sensitive surface, an increased attention to the dynamics aspects over the visual one is required [16]. Accordingly, methods proposed for recognition of gestures often depend strongly on dynamic information [11, 17, 18, 19].

The most recent developments in recognition of isolated pen-based patterns have been conducted in two opposite directions. First, a tendency to more and more complex recognition systems can be noticed, in an attempt to describe simultaneously the different aspects of patterns either by increasing the number of features [10] or by combining multiple classifiers [14, 20, 15]. Conversely, the second tendency is concerned in the design of lightweight, low-resource classification systems well adapted for gesture and symbol recognition on smart phones [19, 18, 17]. Such systems are highly efficient for recognition of specific sets of simple gestures, but do not tolerate much variation in the writing style or drawing process, and cannot be easily generalized to other categories of symbols.

Although feature-based approaches for hand-drawn pattern recognition have been very popular, the design of feature set is often rather obscure or hardly explained, while more attention is paid to the recognition method. While some research works focus on specific feature extraction

---

*Corresponding author
Email address: adrien.delaye@irisa.fr (Adrien Delaye)

techniques, we believe there is a lack of a general view about feature set design. Moreover, although many benchmarking datasets have been proposed in the community (such as Ironoff database [21], NicIcon database [22], or SIGN database [23]), surprisingly no baseline representation space exist for recognition of diverse patterns, including digits, characters, symbols and gestures. It is then very difficult to compare methods and to evaluate the interest of new systems proposed in the community.

Our work is guided by the following question: *"Is it possible to design a rather universal set of features, having a reasonable size, implying no specific drawing constraints and showing good performance on diverse benchmarking datasets ?".* A versatile solution to this problem would be of great help for highlighting true progress when introducing new sets of features or recognition systems in the field.

Deep motivations to the quest for universality are the foreseen evolutions of pen-input recognition systems. Firstly, an important forthcoming challenge is the ability to deal with heterogeneous input (i.e. allow the user to mix pen gestures, handwriting, informal sketching, mathematical notation, symbol, geometrical shapes . . .). Secondly, some pen-based interfaces already admit user-defined classes of symbol or gestures [23][18][12], but more and more liberty will be expected by the user: freedom to draw single-stroke or multi-strokes gestures, choice to tolerate orientation or scale invariant symbols, ability to dynamically add or delete classes of symbols, and so forth. In this setup, the designer is not aware of the patterns the system will have to discriminate, hence the need for universal representation of patterns. These evolutions call for serious progress towards the design of feature sets that are versatile, rather than optimized for the representation of specific and homogeneous datasets.

Beside flexibility, the simplicity of description is also highly desirable. Extraction of a baseline feature set should be easy to implement, involve no hidden parameters or complex algorithms, and imply only a reasonable coding investment with limited processing resources. The space dimensionality should be as limited as possible. Again, the evolution of pen-based devices also calls for light systems with small features spaces: whereas the training of statistical recognizers require more samples as the feature space grows, in a real situation it is clearly not acceptable to ask the user to perform many training samples before he can use a customizable gesture recognition system [24]. Moreover, the credibility of the baseline representation can only benefit from a moderate dimensionality. If the baseline is not only a highly accurate method, but also a simple and light system, then it is guaranteed to be a more useful safeguard against unjustified complexity of future systems.

This study presents a solution to address these needs. We introduce a unique set of 49 features, called Heterogeneous Baseline Feature set (**HBF49**), covering diverse aspects of patterns characteristics, and presenting the following properties:

- ability to describe unconstrained pen-based input (number of strokes, writing order, direction);

- comparable/better performance with respect to state-of-the-art results on various benchmarking datasets, by using a standard Support Vector Machine (SVM) classifier;

- high performance with a simple 1-Nearest-Neighbor (1NN) classifier;

- dealing with writer-dependent (WD) or writer-independent (WI) experimental settings;

- limited in size (reasonably low number of 49 features).

The contribution of this paper is to propose this general basis for representation of handwritten patterns. Obviously, *optimality* of the feature set is not possible to define in an open-world context, since this notion is dataset-dependent. We thus adopt an empirical *constructive approach*, by integrating several families of features in order to exhaustively cover the aspects of hand-drawn objects found in a large selection of diverse datasets. To maintain a feature space of limited dimension, we only consider the simplest features sets from each family for inclusion in the description. Intuitively, maintaining the description as simple as possible should improve its robustness with respect to samples from unseen datasets. Experimental results obtained with standard classifiers show a consistently high performance for 8 datasets with large diversity of content, and universality of the representation is further attested by the good results on a set of user-defined gestures. Our constructive approach is also validated in itself, because **HBF49** description even outperforms several systems from the literature on the dataset for which they have been optimized.

HBF49 is then an efficient benchmarking tool for online symbol and gesture recognition that can be used in several ways: new feature sets can be compared to **HBF49** for evaluation, or new recognition systems can be compared with **HBF49** coupled with an SVM classifier (as in this paper). Additionally, we make the **HBF49** representation of the 8 experimental datasets available for enabling the comparison of machine learning methods (see experimental section 5).

In the next section, we review research works related to recognition of isolated online hand-drawn patterns (section 2) and highlight the great variety of existing approaches. This review is enriched by the presentation of 8 online symbol datasets (section 3) that provides an overview of the diversity of symbols properties and serves as a basis for supporting the choice of features as well as for experimental validation of our feature set. The construction of **HBF49** is presented as the main practical contribution of this paper, in section 4. Rather than adapting the features by optimal selection on datasets, we adopt a constructive approach, where the diversity of features

is looked for while maintaining simplicity of the description (section 4). In the experimental section (section 5), the feature set is shown to provide an excellent description for highly different types of symbols: handwritten digits, mathematical symbols, iconic gestures, geometrical objects, architectural objects, single stroke gestural commands, and user-defined gestural commands. For insuring reproducibility of our experiments, we exploit off-the-shelf Nearest-Neighbor and Support Vector Machine classifiers, and make extracted features datasets available under the WEKA feature format [25]. The associated discussions exemplify how this general baseline representation space can be used for asserting the interest of recently introduced symbol recognition systems.

2. Related work

This section provides a review of recognition methods for isolated online symbols. We deliberately exclude methods for recognition of text (based on Latin alphabet, but also Asian, Indian, Arabic characters and so on), because their linguistic nature makes them beyond the scope of this study. We rather limit our investigation to the analysis of isolated online symbols, digits, pictograms, and pen gestures. Because of the diversity of patterns and the broad spectrum of associated applications, existing recognition systems differ regarding several aspects: drawing constraints imposed to the user (handling of mono or multi-strokes, order or direction-sensitivity. . .), pattern representation (feature vectors, image templates, structural representations. . .), or decision-making process (distance-based classification, discriminative classification, rule-based). Notably, the consideration of the temporal dimension of the signal in the different approaches varies greatly depending on the nature of patterns and the target application, so we propose to categorize the methods according to how much importance they give to this dynamic information. Note that the large variety of methods exposed in this section reflects the importance of our work, as it highlights the need for a comparable landmark feature space that will serve as a unified reference.

2.1. Image-based methods

Image-based approaches do not consider the online information at all in representation and classification of pen-input patterns: the focus is on the *what the pattern visually looks like* rather than *how the pattern was written*. This offers the advantage of high robustness with respect to variations in the number of strokes, writing order and direction, as may be required, for example, in a writer-independent multi-strokes symbol recognition context. On the other hand, only considering the visual aspect of the patterns is a critical loss of information in style-consistent databases, such as in writer-dependent digit recognition or single stroke gestures applications, where discrimination between gestures sometimes only relies on dynamics

(e.g. when a downward and an upward vertical strokes are considered to be different symbols).

For digit and character recognition, as for handwriting recognition in general, many image-based classification systems have been designed. Obviously, offline recognition techniques qualify in this category (see for example [26, 27] for digit recognition). A popular feature set for digit recognition is the stroke direction features, represented as a histogram of local direction elements [27]. Moments of inertia, Zernike moments and other statistical combination extracted from static pattern representation can also be used [28][29].

Likewise, several approaches for symbol recognition consider only the visual aspect of the hand-drawn pattern and neglect information about the drawing process. For the PowerPoint shapes symbols, Hse and Newton [8] extract Zernike moments as features, and use a Support Vector Machine (SVM) for classification. In the works of Kara and Stahovich [12], sketches are represented as binary image templates, and Hausdorff distance is proposed for template matching. High recognition performance is reached on a set of noisy symbols, with unconstrained number and order of strokes. A very similar method is the one of Ouyang and Davis [9] for recognition of pen digits, PowerPoint shapes symbols and sketched electrical symbols. The online input signal is converted into a set of low resolution images, and classification is performed by computing an Image Deformation Model (IDM) distance between input images and prototype images, resulting in very good performances with the three datasets. For recognition of multi-stroke geometrical shapes, Fonseca and Jorge [30] first extract geometrical features based on the enclosing rectangle and the convex hull of the strokes, then classify symbols by applying fuzzy rules inferred from training data. Apte et al. [31] classify geometric shapes by a filter-based approach that considers global geometrical features. Oltmans [32] constructs a static signature of hand-drawn symbols by combining local shape contexts.

2.2. Structural methods

These methods adopt a structural representation of the hand-drawn patterns, involving a segmentation process for identification of the individual structural primitives. Although structural methods may theoretically be strictly visual and accord no importance at all to the online information, most of them actually somehow depend on the writing order, the pen speed, or the number of strokes, because these dynamic aspects offer precious information for the segmentation process. Wether or not utilizing the dynamic information in the structural matching is up to the system designer, so these methods usually are not suited to cope with many different datasets. Indeed, for handling different datasets such as symbols, digits and gestures, it is necessary to adapt the importance of dynamic information in the recognition. Structural methods usually perform better on clean datasets, with linear or geometrical

strokes, because noisy or highly curved strokes are harder to segment properly.

For recognition of multi-strokes symbols from architectural plans, Mas et al. [33] introduced a method based on adjacency grammars, where the hand-drawn shape is considered as a set of linear segments (primitives) and geometric constraints, and described with production rules that can be inferred from training samples. Graph-based representations have also been widely used for recognition of multi-stroke sketched symbols, like in the works of Lee et al. [7] or Hammond and Paulson [34]. Calhoun et al. [35] represent symbols as a semantic network of primitives with their relationships, and utilize online information (pen speed) for segmentation of primitives. In [36], the relative positioning of strokes is considered as information of primary importance for recognition of multi-strokes symbols, thus allowing increased robustness to variation in order or number of strokes.

Several approaches for sketch recognition consider the sketch as a scene containing symbols that can be isolated and recognized by applying predefined grammar production rules constrained by spatial or temporal limitations [37, 38, 39]. Obviously, these approaches are not suited to the recognition of curved patterns, or handling of noisy patterns, because of the segmentation process that can only handle geometrical shapes. Macé and Anquetil [40] proposed a grammar-based approach for recognition of hand-drawn structured documents where temporal and spatial constraints are used for triggering reduction of grammar rules. In that case, the online pressure information (pen-ups and pen-downs) plays a critical role in the parsing process.

2.3. Trajectory-based methods

In trajectory-based methods, the actual pen trajectory is strictly reflected in the pattern representation or in the recognition process. These methods are intrinsically subject to variations of the writing process: order and number of strokes, direction of writing, pen-ups, pressure and speed.

Dynamic Time Warping consists in measuring similarity between patterns seen as sequences of points (or sequence of direction codes). It has been used for recognition of digits and characters [41][42] as well as multi-stroke gestures [22]. Hidden Markov Models can be seen as stochastic models of point sequences. They are very well suited to the modeling of handwritten words and sentences [43], but have also been used for multi-strokes symbols recognition [44] and gestures [45]. For digit recognition, Kherallah et al. [46] designed a beta-elliptic representation of the pen trajectory and velocity.

Trajectory-based methods have been used extensively for recognition of single stroke gestures, for which the online information is of primary importance. The one-dollar recognizer [18] is a popular lightweight statistical method for single stroke gesture recognition. After resampling, rotation, scale and translation normalizations, the input

gesture is compared to templates of gesture models by computing a path-distance, which is sensitive to the ordering of points. Protractor [17] performs quite similarly, but the matching of templates is based on an optimal angle distance. N-dollar recognizer [19] can additionally handle multi-stroke gestures, at the cost of an increased combinatorial in the matching, so as to handle alternative stroke orders and directions. These methods are simple to implement, require little processing resource, and are extensible to introduction of new classes of gestures because they rely on the nearest-neighbor classification paradigm. Although they perform efficiently on simple gestures, they are not suited to recognition of messy and unconstrained drawings, such as multi-stroke gestures or sketchy symbols.

Some feature-based approaches are also strongly linked to the drawing process. Trajectory features are often extracted from the spatially resampled and rescaled hand-drawn signal, forming a feature vector that can be classified by trained statistical recognizers: kNN, neural networks, SVMs... Ahmad et al. [47] extract such features for digit recognition, in association with several variants of neural networks. It is also used as a baseline approach for comparison with other feature sets in the works of Keshari and Watt [29] about character and mathematical symbol recognition, or by Niels et al. [22] for multi-stroke gestures recognition. In [48], Bahlmann proposes an efficient representation for circular variables such as the tangent slope angle extracted along the pen trajectory that is combined with linear features (coordinates of points). The technique proposed by Chakraborty and Chakraborty [49] for online alphanumeric characters also extract features from the ordered sequence of points, after a special pre-processing technique is applied for guaranteeing orientation and scale independence. Golubitsky and Watt [50] recently introduced a new trajectory-based set of features, defined from Legendre-Sobolev coefficients of the input signal seen as a temporal function, and applied it to multi-strokes symbol recognition.

2.4. Hybrid methods

As a last category, we consider methods where the representation or the recognition decision explicitly combines static, image-based information with dynamic, time-based information.

It has been stated by several authors that considering simultaneously the two aspects is beneficial for recognition of handwritten patterns. For digit recognition, Alimoglu and Alpaydin [13] experimented different strategies for combination of static and dynamic classifiers, showing a significant increase in performance for both writer-dependent and writer-independent setups. Arandjelovic and Sezgin [14] proposed to combine a temporal HMM classifier with a static classifier based on Zernike features for sketch and multi-strokes gestures recognition.

A simpler way to combine the representations is to include different types of features in feature-based pattern representation. As an example, the well-known Rubine's

algorithm [11] for single-stroke gesture recognition relies on a set of 13 features, some of which depend on the writing process (e.g. distance between the first and last point of the pen trajectory), while some others do not (e.g. length of the bounding box diagonal). Paulson et al. [51] suggested combining more geometrical features as a complement to Rubine features for increasing the importance of visual information. In the works of Laviola and Zeleznik [15], a feature set is also constructed by considering both online and offline aspects of mathematical symbols. For recognizing multi-strokes characters and gestures, Willems et al. [10] proposed an original feature set by combining stroke-level features, following an idea also evoked in [15]. A heterogeneous feature set, containing both static and dynamic information, is extracted from the global multi-strokes symbol. Then, the same set of features is extracted from each of the symbol strokes. In order to deal with varying number of strokes per sample, the stroke-level features are averaged in a unique stroke-level feature set with a constant size. A third feature set is computed from the covariance of stroke-level features. The full set, obtained by concatenation of the three sets, contains a very large number of features that model both static and dynamic information at global and local levels.

Directional features are a popular type of hybrid features for isolated character recognition [52, 53]. In the works of Bai and Huo [52] and that of Liu and Zhou [53], directional features are extracted from the online trajectory and mapped to directional images that are blurred and subsampled. The final features extracted from the images combine dynamic information (because directions are defined with respect to the writing direction of the strokes) and visual aspect (because the writing order is lost in the image representation).

2.5. Discussion

We have enumerated in this section a large number of methods for recognition of handwritten symbols, characters, digits, and gestures. In spite of a very similar problem (isolated handwritten symbol recognition), the methods focus on different properties of the patterns while neglecting some of their aspects. It is extremely difficult to compare methods from different categories, and applications in different domains seem to evolve separately: for example, methods for multi-stroke gestures recognition are not checked against the problem of digit classification, although digits could be seen as examples of pen-gestures. The use of benchmarking datasets for comparison is still not generalized, and is almost never adopted for validation across various application domains. The large variety of approaches and the lack of universal baseline make it difficult to emphasize true contributions to the field, such as the introduction of new powerful feature sets or new structural systems. Hopefully, a good baseline method can ease the evaluation of new methods and permit to judge if the development of heavy and costly systems is worth the cost.

In the rest of this paper, we introduce a feature set suited to the representation of symbols for different applications as a first unified baseline for evaluation of online symbol recognition methods. We adopt for this a hybrid feature-based approach, for two main reasons. First, hybrid feature-based methods provide a representation space where the patterns can be discriminated according to different criteria depending on the context. Unlike image-based methods or trajectory-based methods, they are suited to description of various types of patterns, from gestures where the dynamic information is essential, to hand-drawn symbols where the visual aspect is more important. Secondly, and contrarily to structural methods, the recognition procedure of feature-based methods is inherently independent of the drawing process and thus imposes no constraints to the user, hence a more easy generalization to many applications. No choice has to be made beforehand: by guaranteeing diversity in the features, it may be possible to insure that the method will properly handle different categories of patterns, with or without drawing constraints, in writer-dependent or independent settings.

We expose in section 4 the design of **HBF49**, an original set of heterogeneous features, in an attempt to address the universal symbol recognition problem. We will adopt for this a constructive approach, inspired by related works [10][11][15] and based on the observation of symbols from different datasets. Beforehand, we present a selection 8 symbol databases that cover different application domains as an overview of the problems occurring in online symbol recognition.

3. Benchmarking datasets

This section presents a selection of benchmarking datasets that are available to the community (they can either be downloaded from a website, or be obtained by contacting their authors). We selected them so as to cover many different aspects of online symbol recognition and a large panel of potential applications, but did not considered handwriting text recognition databases, such as IAM-Online database (for English handwritten text recognition) [54] or CASIA-OLHWDB (online Chinese character recognition) [55]. The eight retained datasets present a large diversity of content with very different properties (for example the relative importance of dynamic versus visual information greatly varies among the datasets). Some datasets offer writer-dependent (WD) or writer-independent (WI) experimental settings, their number of classes varies, as well as the available amount of training samples per class. Some constitute a quite easy recognition problem, with clean or style-consistent data, while others are much more challenging because of the presence of noisy data, larger variety of writing styles and higher number of classes or existence of very similar classes.

This selection is believed to exhaustively cover the contexts where online symbols are likely to be used, and thus to provide a reliable overview of the related recognition

Table 1: Properties of the datasets. Nature of patterns is described with following flags: N for multi-strokes patterns, S for single-strokes, R for rotated patterns, V for vectorized. WI and WD columns indicate the possibility of running Writer-Independent and Writer-Dependent experiments.

dataset	classes	samples	nature	WI	WD
Ironoff-digits	10	4,086	N	y	n
LaViola	48	11,602	N	y	y
CVCsymb	25	4,278	N	y	y
ImiSketchS	10	1,020	N,R,V	y	n
HHReco	13	7,791	N,R	y	y
NicIcon	14	26,163	N	y	y
Sign	17	33,579	S	y	y
ILG	588	4,656	S	n	y

difficulties. We will use evidence from observation of these datasets as a guide for supporting the design of **HBF49** feature set in the next section 4. Moreover, we will make use of these datasets for experimentally validating the universality of **HBF49** in section 5.

Table 1 presents the datasets properties, and figure 1 presents samples extracted from each dataset for visually attesting the diversity of their content.

The **Ironoff-digits** (see figure 1(a)) is a subset of IRONOFF database, a dual online/offline database for isolated character recognition [21]. **Ironoff-digits** is limited to the 10 classes of digits in their online representation. It contains 4,086 samples and offers a challenging writer-independent context, with many different writing styles and only one sample per writer and per class. Results with this datasets were reported in [47][56][57], and it served as a benchmarking dataset for Lipitk library [58].

The **LaViola** dataset [15] (figure 1(b)) is contains a total of 11,602 samples of handwritten digits, characters and mathematical symbols, written by 11 persons. Characteristics of this dataset are the higher number of classes (48) and the relatively few training samples (10 per class and per writer). Writer-dependent and writer-independent experiments are possible, and results on this dataset have been reported in [15][41][29].

CVCsymb is a dual online/offline database of architectural and electrical symbols [33](figure 1(c)), containing 50 different symbols divided in two sets of 25 (each writer was asked to draw samples from one of the two sets of classes only). We retain in our experiment only a half of the online dataset, consisting of 25 classes of symbols collected from 12 writers, totalizing 4,278 samples. The symbols are composed of linear sections and arcs, freely drawn, with a varying number of strokes (from 2 to 20 strokes, with an average of 6) written with varying order and direction [59]. Writer-independent and writer-dependent experiments are possible, but no global recognition experiments have been reported [33].

ImiSketchS is a recently collected database of hand-drawn architectural symbols [60] (figure 1(d)). We retain from it the set of 1,020 samples from the 10 classes of *furniture* symbols. A particularity of this database is the vectorized form of the patterns resulting from the collecting process. Samples are handwritten offline, and analyzed with an architectural plan analyzer presented in [61, 62]. Segments are detected from the image data, and extracted symbols are represented as sets of segments, with no online information (information on the order and number of strokes is lost). The drawing conditions are constraints-free, and the symbols may be oriented in any direction, depending on their original context. Writer-independent experiments can be conducted.

The **HHReco** [8] dataset is a benchmark of 13 sketched geometric shapes (regular polygons, circle, arch, cylinder...) (figure 1(e)) referred to as *Powerpoint symbols*, and composed of a mix of linear and curved sections. 19 users provided a total of 7,410 examples (about 30 per class and per user), with no constraints in the writing process (orientation, number and order of strokes). Both writer-dependent and writer-independent contexts were tested and results were reported with different methods [8][9].

NicIcon database [22] contains 14 classes of iconic multi-strokes gestures (figure 1(f)). The gestures represent chosen pictograms expressing emergency situations (e.g. fire, accident). While these expressive symbols are familiar and easy to recognize for the human eye, no standard writing process is known to the writers. Consequently, the database is quite noisy, with a highly varying number and order of strokes, retraces, sloppy drawings. This rich dataset contains a total of 26,163 gestures from 34 writers (in average, 55 samples per writer per class), with both offline and online data, and includes pressure measures and pen-ups movements measures. For harmonization with other datasets, we simply consider online pen-down strokes, and remove pressure information. Results for WI and WD experiments with online data have been reported in [10, 14, 20, 63].

Sign is a database of 17 classes of single stroke gestures [23](figure 1(g)). 33,759 samples were collected from 20 users (about 50 samples per class per writer), and results were reported for incremental recognition in WD experimental settings only [24, 23]. Some classes of gestures in **Sign** can only be distinguished based on dynamic information, because they are identical up to a change in the writing direction.

The last dataset is **ILG**, a collection of single stroke pen-based gestures (figure 1(h)) [64]. This dataset contains many user-defined gestures: in the collection process, users were asked to design a set of 21 gestures of their choice for triggering commands in a simulated image editor application. The collection process was carefully designed so as to build an immersive environment and to guarantee a realistic usage scenario (see the details in [64]). We retain the datasets *test1* and *test2*, where each of the 28 writers provided a total of $28 \times 21 = 588$ potentially different gestures, unequally distributed over the classes. Since
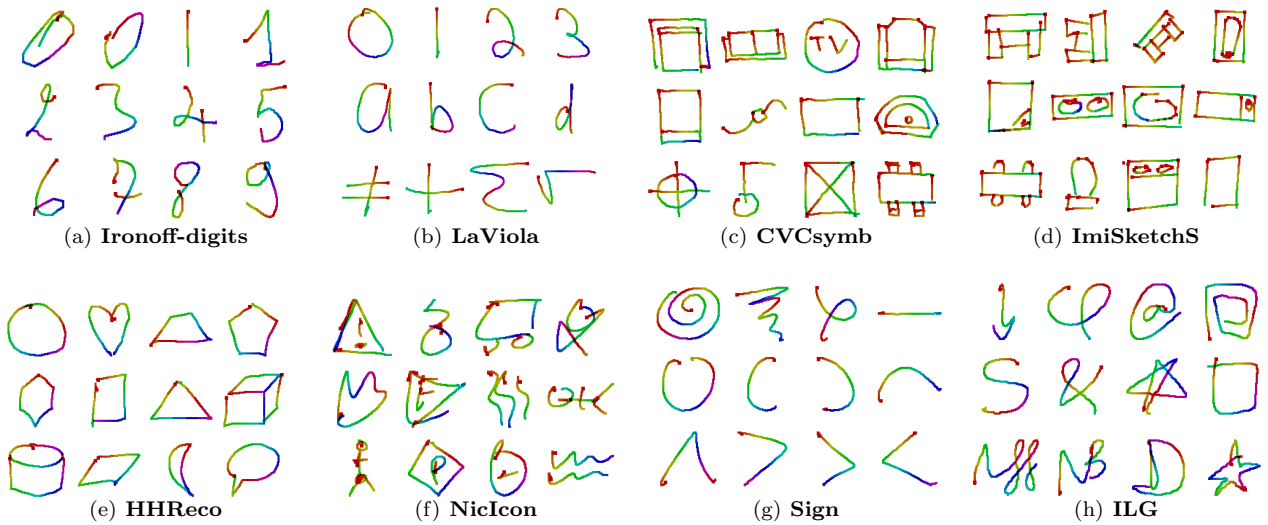
| (a) **Ironoff-digits** | (b) **LaViola** | (c) **CVCsymb** | (d) **ImiSketchS** |

| (e) **HHReco** | (f) **NicIcon** | (g) **Sign** | (h) **ILG** |

Figure 1: Samples from the 8 datasets.

ILG contains a very large number of different gestures, it offers a good benchmarking tool for experimenting how universal a feature set representation is. Simultaneously, because it only offers few training samples for each class, it also constitutes a realistic challenge regarding the representation space complexity: in order to train statistical models for recognition of user-specified classes of gestures, the feature space should remain limited in size. Obviously, this dataset can only be used for writer-dependent experiments.

The selection of datasets presented here is believed to cover the range of problems found in isolated symbol recognition in general: style variations (number and order of strokes, direction of writing), high number of classes, rotations... The diversity of contents also emphasizes the needs for combining different categories of features in a universal description. For example, dynamics information have an obvious importance for the *gestures* datasets (**NicIcon**, **Sign** and **ILG**), while more visual description seems required for **CVCsymb** or **HHReco** datasets.

4. Heterogeneous baseline feature set

We introduce here **HBF49** (for Heterogeneous Baseline Features): a set of 49 features for unified description of heterogeneous datasets, where diverse features are considered in a constructive approach, with the objective of accurately representing symbols from multiple contexts (such as in the 8 datasets presented in section 3). First, we discuss the constructive approach adopted and justify the choices for inclusion or exclusion of several types of features. We then introduce a series of useful notations and definitions, and describe preprocessing methods applied on the handwritten symbols before the feature extraction itself. The categories of features included in **HBF49** are

then presented, with a formal definition given for each feature.

4.1. Constructive approach justification

In our view, approaches for designing a set of features can be categorized as *theoretical-based*, *constructive* or *selective*.

Theoretically-based feature sets consist of a number of features that is theoretically funded, by their definition. As an example, the number of Zernike features is determined by the order of Zernike moments considered (it may depend on a tolerated reconstruction error of the image)[8]. Theoretical-based approaches are of great interest and often permit to make progress by discovering new representations; however it is unlikely that any family of theoretically-based features can handle symbol representation universally, considering the diversity of datasets.

Selective approaches aim at selecting the best subset of features (or combination of features), e.g. by optimization of the recognition score on a given dataset. Selection is usually applied on a large set of features (itself either theoretically defined or defined by construction, as in [10]). By definition, these approaches do not qualify for the universality problem directly, because they are tuned with respect to a dataset. To be widely used, we also believe that a baseline representation should be simple and require no such data-dependent optimization scheme.

Constructive approaches are commonly used for solving specific problems. They are empirical, based on the designer intuition, and often inspired from data observation. The construction process consists in gathering features describing different aspects of the data samples to constitute an accurate global description. This is naturally the approach we adopt, inspired by the work of Willems et al. [10], Rubine [11] and LaViola and Zeleznik [15].

7

Contrarily to references [10][11][15], we try to exhaustively describe aspects of online symbols from a variety of contexts. Rubine's work was focused on single stroke gesture recognition [11], hence his feature set is not suited to describe multi-stroke gestures or complex symbols because it contains no visual features and no global description [10, 51]. The feature set of Willems et al. [10] is efficient for multi-strokes gestures representation, by combining global and local stroke-level features, but at the cost of a high dimensionality (758 features are defined in [10]). Laviola and Zeleznik combine dynamic and visual features for mathematical symbols recognition [15], by including both global and local level description in a set of 47 features. However, it is biased towards dynamics information which is exaggeratedly important in their Writer-Dependent context, with a dataset mostly composed on single-stroke gestures.

The methodology we adopt consists in combining different families of features, so as to exhaustively cover the aspects of online symbols encountered in the literature. We integrate subsets of features that are more or less sensitive to different properties of the patterns: sensitivity to the writing process (with dynamic features), to the visual aspect of the pattern (with visual features), invariance to the pattern orientation...

To solve the apparent trade-off between the desired description simplicity (low dimensionality) and the universality of representation, we intuitively pick what we consider to be the simplest set of features for each family and consider for inclusion only small groups of features with a maximum size of 10 each. This choice disqualifies many theoretical-based features sets, such as high-order Zernike moments [8], or directional features extracted from a regular grid [53]. Moreover, since we want the description to be as general as possible, we do not perform any selection of features based on training data, and deliberately omit features automatically designed from data such as the ones proposed by Lemieux et al. [65]. We believe that this inclination towards simplicity of description will also be strongly beneficial for the robustness of the representation, which is highly desirable for universal symbol recognition.

The good balance between the families of features included in this **HBF49** is essential for universality. In comparison with [15], we then strengthen the visual aspect of the description and integrate more orientation-invariant features. Most of features included in **HBF49** are inspired from [10][11][15], from which we picked the ones we considered as important, simple to implement (i.e. involving no complex extraction algorithm) and general (that are meaningful in any dataset). Additional families of visual features are incorporated: zoning features (as a coarse global quantization of the image, sensitive to rotation), histogram of directions (quantization of the trajectory direction, invariant to rotation), and Hu moments (global moment-based image description, invariant to rotation). These three sets offer complementary description of the patterns, while being independent on the writing process and very limited in size (9, 8 and 7 features respectively).

The resulting feature set **HBF49** is not claimed to be the best possible set of features for universal symbol recognition. However, we are confident that it provides an accurate and robust description of symbols in a great variety of context thanks to a complete coverage of the patterns properties. If our constructive approach cannot guarantee optimality of performance, we are however convinced that only a rather large modification of the feature set can bring a significant improvement of the general performance.

Since we present **HBF49** as a baseline method, we give here all the details that guarantee an easy reproduction of the feature extraction. Whenever reproducing a feature presented in one of the reference works [10][11][15], we mention it with its associated index for completeness.

4.2. Definitions and notations

Definition A pattern $\mathcal{S}$ is a (possibly resampled) sequence of points captured from the trajectory of the pen position on the input device. $\mathcal{S} = \{s_1, .., s_i, .., s_n\}$, where n is the number of points in $\mathcal{S}$. Each point $s_i = (x_i, y_i)$ is located in the bidimensional space.

For homogeneous representation, we do not consider time or pressure information that may be available in some cases. However, some points can always be detected as pen-up points: they denote points of the trajectory where the contact between pen and surface was interrupted. In particular, the last point s_n from $\mathcal{S}$ is necessarily a pen-up point.

Definition A stroke S is a subsequence of a pattern $\mathcal{S}$ that is preceded and ended by pen-up points. It describes the trajectory of a unitary element of writing. A pattern has at least one stroke. We denote S_k the sequence of points $\{s_i..s_j\}$ belonging to the k-th stroke of $\mathcal{S}$, hence $\mathcal{S} = \cup_{k=1}^{K} S_k$, where K is the number of strokes in $\mathcal{S}$.

We use the following notation for characterizing properties of the pattern $\mathcal{S}$ (see figure 2(a) for visualization on an example):

- u_x, u_y are the unit vectors codirectional with the x and y axes (we assume y is pointing down as in image coordinate system),

- n is the total number of points in $\mathcal{S}$,

- s_1 and s_n are the first and last points in the writing order of $\mathcal{S}$,

- $||s_a s_b||$ denotes the Euclidean distance between points $s_a = (x_a, y_a)$ and $s_b = (x_b, y_b)$,

- $L_{i,j}$ is the path length from s_i to s_j, where path length is accumulated as follows:

$$L_{i,j} = \sum_{l=i}^{j-1} \begin{cases} 0 & \text{if } s_l \text{ is a pen-up point} \\ ||s_l s_{l+1}|| & otherwise \end{cases}$$

8

- $L = L_{1,n}$ is the total length of $\mathcal{S}$,

- s_m is the middle-path point, i.e. the point such that $L_{1,m} = L_{m,n}$,

- x_{max} is the abscissa of the rightmost point of $\mathcal{S}$, x_{min}, y_{max} and y_{min} are the other extreme coordinates,

- B is the bounding box of $\mathcal{S}$, it is the rectangle parallel to the axis, defined by x_{min}, x_{max}, y_{min}, y_{max},

- $w = x_{max} - x_{min}$ is the width of B, and $h = y_{max} - y_{min}$ is the height of B, (if w or h are null, their value is set to 1),

- c_x and c_y are the coordinates of the center of B,

- $\mu = \frac{1}{n} \sum_{i=1}^{n} s_i = (\mu_x, \mu_y)$ is the center of gravity of the pattern.

4.3. Preprocessing

Prior to the feature extraction itself, we expose the preprocessing operations applied on the input patterns. We chose to normalize patterns in scale and translation, and to apply a trajectory resampling strategy. These operations are simple to perform and guarantee a better stability of extracted features, for any type of input pattern. Normalization of the pattern orientation is not desirable in general, since sensitivity to orientation can be required in some datasets where patterns from different classes are identical up to a rotation.

Linear scaling and translation. An input pattern $\mathcal{S}$ is first rescaled so that its maximal dimension is equal to a normalized size. This rescaling is linear; hence produces no deformation of the pattern proportions or shape. We chose the normalized dimension as 128 in all our experiments. The scaled pattern $\mathcal{S}'$ is translated so that the top-left corner of its bounding box matches with the origin of axes.

Trajectory resampling. The sampled points of the input pattern are unequally distributed along the trajectory, due to variations in writing speed. Although this brings some dynamic information about the drawing process, an increased robustness of feature extraction is obtained by first resampling the points. We chose to spatially resample the points, so that the points on the resampled trajectory are equidistant. The imposed distance between two points in the resampled trajectory is fixed to 8, this value being related to the box dimension of 128.

4.4. Dynamic features

A first category of features included in **HBF49** are the dynamic features. They model the writing process, focusing on *how* the pattern was accomplished by the writer, by implicitly or explicitly describing the order of points in $\mathcal{S}$, the writing direction, the number and order of strokes, and so on. We deliberately chose to exclude features based

on pressure (pressure variations, off-strokes measures) or temporal information (speed, acceleration), because this information is not always available (see [66] for a few examples). We retain in **HBF49** a total of 14 dynamic features, as described below.

Starting and ending points position. Positions of the first and last points constitute important features for distinguishing patterns in many situations. For example, single stroke gestures, digits or letters often present stable beginning and ending points. We compute the coordinates relatively to a virtual square box of side $l = max(h, w)$, centered on $c = (c_x, c_y)$, so as to avoid noisy measures in the case of objects with very small width or height. The two features for the starting point are computed as follows:

$$f_1 = \frac{x_1 - c_x}{l} + \frac{1}{2}, \quad f_2 = \frac{y_1 - c_y}{l} + \frac{1}{2}, \qquad (1)$$

and similarly for the ending point:

$$f_3 = \frac{x_n - c_x}{l} + \frac{1}{2}, \quad f_4 = \frac{y_n - c_y}{l} + \frac{1}{2}. \qquad (2)$$

Similar features are also included in [66] (indexes 35-38) and [15] (*side ratios*).

First point to last point vector. Related to the first set of features, the vector $v = \overrightarrow{s_1 s_n}$ conveys additional information about the pattern dynamics. The vector length $||v|| = ||s_1 s_n||$, as well as the cosine and the sine of its angle with respect to horizontal line are measured (see figure 2(b)):

$$f_5 = ||v||, \quad f_6 = \frac{\overrightarrow{v_x}.\overrightarrow{u_x}}{||v||}, \quad f_7 = \frac{\overrightarrow{v_y}.\overrightarrow{u_y}}{||v||}, \qquad (3)$$

with $v = (v_x, v_y)$. In order to cope with instability of f_6 and f_7 angle measurement when f_5 is small, we force their values to zero when the distance f_5 is below a minimal distance empirically set to $d_{min} = max(w, h)/4$.

Closure permits to highlight differences between closed pattern (such as loops, circles, "0" or "o" characters, polygons) and patterns that are elongated ("l", "1", "∫"). It is defined as:

$$f_8 = \frac{||v||}{L}. \qquad (4)$$

References for features f_5 to f_8 are [11](4,5,6), [66] (59 and 6) and [15].

Angle of initial vector. The initial vector is defined by the first points of the trajectory. In our implementation, and after spatial resampling of the pattern, we actually consider the vector between the first and the third point: $w = \overrightarrow{s_1 s_3}$, as soon as the first stroke has 3 points or more. The initial angle is described by the cosine and sine measures (see figure 2(b)):

$$f_9 = \frac{\overrightarrow{w_x}.\overrightarrow{u_x}}{||w||}, \quad f_{10} = \frac{\overrightarrow{w_y}.\overrightarrow{u_y}}{||w||}. \qquad (5)$$
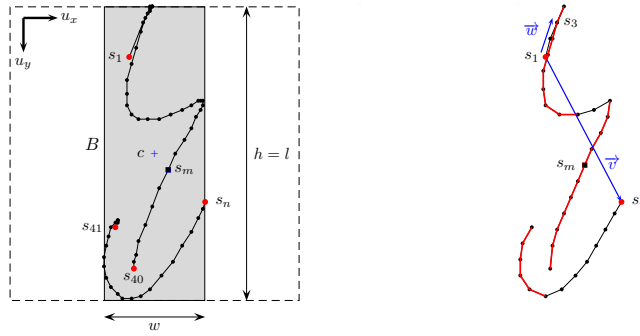
(a) Definitions for a pattern $\mathcal{S}$ with $K = 2$ strokes and 55 points. The gray area represents the bounding box B.

(b) Representation of online features, extracted from spatially resampled and rescaled pattern, with $l = 128$. First-to-last and initial vectors are in blue, downstrokes in red.

Figure 2: Representation of definitions and notations, and definitions of several online features.

Since we consider temporally close points from a spatially resampled signal, it is guaranteed that the distance $||w||$ cannot be too small, so stability of these features is insured. Similar features are used in [11] (1,2) and [66] (55,56).

Inflexions. Two *inflexion* features are measured from the positioning of the middle-path point s_m with respect to the middle point of segment $s_1 s_n$, so as to distinguish between cups oriented in different directions (e.g. "u"-shaped and "n"-shaped patterns) (see figure 2(b)):

$$f_{11} = \frac{1}{w}\left(x_m - \frac{x_1 + x_n}{2}\right), \quad f_{12} = \frac{1}{h}\left(y_m - \frac{y_1 + y_n}{2}\right). \tag{6}$$

These simple features are preferred to explicit cups features that involve more complex cup detection algorithm as in [66](32-34).

Proportion of downstrokes trajectory. As established by Anquetil and Lorette [67], *downstrokes* (portions of drawing trajectories oriented towards the bottom of the writing surface, i.e. oriented towards increasing values in dimension y) are especially important in the perception of handwriting. Since the number of downstrokes might be variable and not pertinent for multi-strokes gestures, we chose to extract the proportion of trajectory length covered by the downstrokes. This single feature is computed as:

$$f_{13} = \sum_{k=1}^{p} L_{k_i,k_j} \tag{7}$$

with p the number of downstrokes. A trajectory $\mathcal{T} = \{s_{k_i}..s_{k_j}\}$ is a *downstroke* if the following conditions are satisfied:

- $\mathcal{T}$ is included in a single stroke,

- $D = \sum_{k=k_i}^{k_j-1} max(0, y_{k+1} - y_k)$, the cumulative downward distance is superior than a threshold T_1,

- $\mathcal{T}$ contains no subsequence $s_p..s_q$ such that the local cumulative upward distance $U = \sum_{k=p}^{q-1} max(0, y_p - y_{p+1})$ exceeds a threshold T_2.

Downstrokes are represented in red in the example of figure 2(b). Considering the mapping of patterns to a normalized box of maximal dimension 128, we found that satisfying values for the thresholds are $T_1 = 2$ and $T_2 = 5$.

Number of strokes. The last feature depending on dynamic information simply accounts for the number of strokes K in the input pattern.

$$f_{14} = K \tag{8}$$

The number of strokes (or number of pen-up points) is of great importance for many cases. Actually, it always is (except for single stroke datasets) a good indication of the pattern graphic complexity. While the number of strokes per sample participates in the individual writing style and can greatly vary from a writer to another, in average people obviously tend to use more strokes for more complex drawings. Some symbols also require to be drawn in several strokes (for example the "+" or "=" mathematical symbols). This feature has been used in [66](44) and in [15].

Contrarily to the works of Willems et al. [10], we decide not to include any local or stroke-level online features, because we want to maintain a small number of features.

4.5. Visual features

The second type of features are *visual* features, in the sense that they do not depend on the writing process, but focus on the appearance of the writing result. Since

visual features are not affected by changes in stroke order or writing direction, they insure an increased robustness for patterns that *look the same*, no matter how they are drawn. This information is of particular importance for recognition of complex or structured symbols, with no standard writing process, and for writer-independent contexts. Moreover, some patterns can only be recognized with visual descriptors, when their identification is based on their Gestalt (e.g. recognizing a scratch-out or a helix gesture may require visual features). Over-tracing, or unexpected connection between strokes are also easier to handle with this type of features [9]. In essence, the role of visual features in **HBF49** is to cover the description of aspects exploited by the *image-based* methods identified in the literature review (see section 2).

In addition to the dynamic/visual complementarity, it is useful to keep in **HBF49** features that have different sensitivity to the orientation of patterns. Whereas in some datasets orientation should be considered as a discriminating factor between classes, in other cases orientation-free representation is better suited. Across the families of visual features enumerated in this section, we will explicitly mention the ones that are independent to the orientation of the pattern.

Bounding box diagonal angle. We measure the box diagonal angle with respect to the horizontal (see also [11](4)):

$$f_{15} = \arctan \frac{h}{w}.$$ (9)

Trajectory length. The trajectory length is an orientation-independent feature also included in [66](1), [11](8) and [15]:

$$f_{16} = L.$$ (10)

A related feature is the ratio between the half-perimeter of the bounding box and the trajectory length:

$$f_{17} = \frac{w + h}{L}.$$ (11)

This feature can characterize graphical complexity of gestures, e.g. differentiate between a spiral and a simple circle having the same bounding box.

Deviation. This orientation-independent feature evaluates the average distance from points of $\mathcal{S}$ to the center of gravity μ (see [66](68)):

$$f_{18} = \frac{1}{n} \sum_{i=1}^{n} ||s_i \mu||.$$ (12)

Average direction. A first directional feature is computed by averaging the directions of segments defined in the trajectory of $\mathcal{S}$ (see also [66](12)):

$$f_{19} = \frac{1}{n-1} \sum_{i=1}^{n-1} \arctan \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right).$$ (13)

Note that this measure does not depend on the writing direction.

Curvature and perpendicularity. These two orientation-independent features compute the summation of local curvature and perpendicularity measure between subsequent segments of $\mathcal{S}$ (see [66](8,13), [11](9)). We denote θ_i the angle defined by consecutive segments within the same stroke:

$$\theta_i = \arccos \left\{ \frac{\overrightarrow{s_{i-1}s_i} . \overrightarrow{s_i s_{i+1}}}{||\overrightarrow{s_{i-1}s_i}|| ||\overrightarrow{s_i s_{i+1}}||} \right\}.$$ (14)

The curvature and perpendicularity are defined as:

$$f_{20} = \sum_{i=2}^{n-1} \theta_i, \quad f_{21} = \sum_{i=2}^{n-1} \sin^2(\theta_i).$$ (15)

The curvature of a straight line is null, while curved shapes have high curvatures. Perpendicularity feature can detect abrupt changes of direction in the trajectory, e.g. right-angle corners of polygons.

k-perpendicularity, k-angle. Two additional orientation-independent angular features are defined from a parameter k that determines another measure of local angles θ_i^k:

$$\theta_i^k = \arccos \left\{ \frac{\overrightarrow{s_{i-k}s_i} . \overrightarrow{s_i s_{i+k}}}{||\overrightarrow{s_{i-k}s_i}|| ||\overrightarrow{s_i s_{i+k}}||} \right\},$$ (16)

where s_{i-k}, s_i and s_{i+k} have to belong to the same stroke. From the angles θ_i^k, we compute the k-perpendicularity and the maximum k-angle:

$$f_{22} = \sum_{i=k+1}^{n-k} \sin^2(\theta_i^k), \quad f_{23} = \max_{i=1+k}^{n-k} \theta_i^k.$$ (17)

Given the chosen pattern dimension normalization and spatial resampling distance, fixing k to 2 yields a good definition for these features. Similar measures are proposed in [66](64,21).

Absolute angle histogram. Additionally to average direction (feature f_{19}), we measure an angle histogram that provides four features accounting for the number of segments oriented in eight directional bins. For each segment the orientation is given by:

$$\alpha_i = \angle(\overrightarrow{s_i s_{i+1}}, \vec{u_x}).$$ (18)

From the eight histogram bins (h_1 to h_8) defined by eight directions, each angle α_i lies between two consecutive bins h_1 (of direction $\vec{v_1}$) and h_2 (of direction $\vec{v_2}$). We propose a fuzzy quantization of angles to the histogram bins, i.e. an angle contributes to two bins with distinct degrees (contrarily to direct quantization proposed by Laviola, where only the best bin is considered [15]). Suppose that α_i is closer to $\vec{v_1}$ than $\vec{v_2}$, then it contributes to h_1 and h_2 with weights w_1 and w_2:

$$w_1 = 1 - \frac{\angle(\overrightarrow{u_{\alpha_i}}, \vec{v_1})}{\pi/4}, \quad w_2 = 1 - w_1,$$ (19)

where u_{α_i} is the unit vector oriented by α_i. Finally, the four features f_{24} to f_{27} are computed as the sum of contributions from all angles α_i to opposite directional bins, in order to guarantee independence with respect to the direction of writing:

$$f_{24} = \frac{h_1 + h_5}{n_a}, \quad .. \quad f_{27} = \frac{h_4 + h_8}{n_a}, \qquad (20)$$

with n_a the number of segments in $\mathcal{S}$ ($n_a = n - K$, with K the number of strokes). These features provide a richer description of the pattern trajectory dominant orientations.

Relative angle histogram. Another directional histogram accounts for local changes of direction, as a complement to the average curvature and (k-)perpendicularity features. It is independent from the symbol orientation. First, we find that relative local angles benefit from smoothing by linear combination of θ_i and θ_i^k (see definitions in equations 14 and 16):

$$\psi_i^k = \gamma\theta_i + (1 - \gamma)\theta_i^k, \qquad (21)$$

where we empirically set $\gamma = 0.25$ and $k = 2$. The contributions of ψ_i^k angles are cumulated in four histogram bins uniformly distributed in $[0, \pi]$. Like for histogram of absolute orientations, contributions to the histogram of relative orientations are weighted by the inverse of their angular distance with the central direction of the two neighboring bins. Four features f_{28} to f_{31} are obtained from the histogram divided by n_a.

2D histogram. We define a regular 3x3 2D partition of the bounding box B, providing a global description of the points repartition by 9 zoning features [68]. Like for other histograms, we compute a fuzzy weighted contribution from each point to its 4 neighboring cells, where the weights depend on the distance from the point to the cell centers [69]. Figure 3(a) represents the partition defined over the bounding box of a sample pattern, with the cell centers $c_{i,j}$. With this definition, a point s (in red) contributes to four neighboring cells with degrees $\mu_{12}, \mu_{13}, \mu_{22}$ and μ_{23}. Features f_{32} to f_{40} describe the accumulated contribution of the points of $\mathcal{S}$ to the nine cells, divided by n:

$$f_{32} = \frac{1}{n}\sum_{i=1}^{n}\mu_{11}(s_i) \quad .. \quad f_{40} = \frac{1}{n}\sum_{i=1}^{n}\mu_{33}(s) \qquad (22)$$

These features were not included in the large NicIcon feature set [66], but we believe that they provide a precious information about the global visual aspect of $\mathcal{S}$. Laviola and Zeleznik exploited similar features [15].

Hu moments. Hu moments are a set of seven features computed from inertia moments and presenting invariance to deformations such as rotation, scale, and translation [70]. First, inertia central moments are computed:

$$m_{pq} = \sum_{i=1}^{n}(x_i - \mu_x)^p(y_i - \mu_y)^q, \quad \text{for } 0 \leq p, q \leq 3 \qquad (23)$$
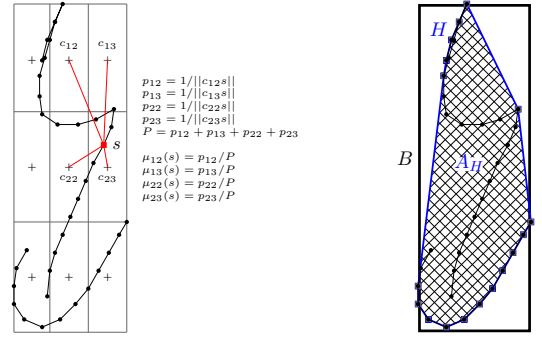
(a) Fuzzy 2D Histogram computation   (b) Convex hull

Figure 3: Representation of offline features definitions.

where $\mu = (\mu_x, \mu_y)$ is the center of gravity of $\mathcal{S}$. The moments are then normalized, for guaranteeing scale independence:

$$\nu_{pq} = \frac{m_{pq}}{m_{00}^{\gamma}}, \quad \text{with } \gamma = 1 + \frac{p + q}{2}. \qquad (24)$$

The seven Hu moments, orientation-independent by construction, are finally computed as described in [70]:

$$
\begin{aligned}
f_{41} &= \nu_{02} + \nu_{20} \\
f_{42} &= (\nu_{20} - \nu_{02})^2 + 4\nu_{11}^2 \\
f_{43} &= (\nu_{30} - 3\nu_{12})^2 + (3\nu_{21} - \nu_{03})^2 \\
f_{44} &= (\nu_{30} + \nu_{12})^2 + (\nu_{21} + \nu_{03})^2 \\
f_{45} &= (\nu_{30} - 3\nu_{12})^2(\nu_{30} + \nu_{12}) \\
&\quad [(\nu_{30} + \nu_{12})^2 - 3(\nu_{21} + \nu_{03})^2] \\
&\quad + (3\nu_{21} - \nu_{03})(\nu_{21} + \nu_{03}) \\
&\quad [3(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] \\
f_{46} &= (\nu_{20} - \nu02)[(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] \\
&\quad + 4\nu_{11}(\nu_{30} + \nu_{12})(\nu_{21} + \nu03) \\
f_{47} &= (3\nu_{21} - \nu_{03})(\nu_{30} + \nu_{12}) \\
&\quad [(\nu_{30} + \nu_{12})^2 - 3(\nu_{21} + \nu_{03})^2] \\
&\quad - (\nu_{30} - 3\nu_{12})(\nu_{21} + \nu_{03}) \\
&\quad [3(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2].
\end{aligned}
$$

Whereas the set of Hu moments is not popular for symbol representation, we integrate it in **HBF49** because it is a good visual descriptor with low dimensionality and powerful invariance properties. Its definition in the domain of geometrical moments also makes it complementary to visual features such as histograms of angles or zoning.

Convex hull features. The last set of features captures more precisely the geometry of the pattern by considering its convex hull. The convex hull H of $\mathcal{S}$ is computed

12

with the Graham algorithm (see [71]). Figure 3(b) represents a pattern with its convex hull. If $H = v_1..v_p$, with $v_i = (v_{i,x}, v_{i,y})$ denote its ordered vertices, its area A_H is easily computed as:

$$A_H = \frac{1}{2} \left| \sum_{i=1}^{p-1} (v_{i,x} * v_{i+1,y} - v_{i+1,x} * v_{i,y}) \right|. \qquad (25)$$

The two related features are the convex hull area normalized by the bounding box area, and the compactness (as defined in [66](2,3)), which is orientation-independent:

$$f_{48} = \frac{A_H}{w * h}, \quad f_{49} = \frac{L^2}{A_H}. \qquad (26)$$

5. Experiments

We conducted experimental evaluation of the **HBF49** pattern representation over the datasets presented in section 3. In this section, we first explain our experimental protocol in details for easy reproducibility. The results are then presented and compared with other results reported in the literature. The first objective of these experiments is to verify the answer to our initial question: we want to show that a simple set of features, with limited size, in association with a standard classification method, can reach satisfying results for recognition of many different sets of symbols. Consequently, we conform to a strictly identical protocol in all experiments, by utilizing the **HBF49** features, with no tuning, optimization, or selection of features whatsoever. The second objective of this section is to exemplify the interest of **HBF49** as a baseline method for evaluating symbol recognition systems by running comparative experiments.

5.1. Experimental protocol

A simple common protocol is employed in the experiments. All the datasets were extracted in the same conditions, with no parameter adaptation in any of the preprocessing or feature extraction steps. The resulting sets of features are made publicly available on a dedicated web page [1], where the Weka *Arff* format [25] is used to store the extracted features. Scripting tools are also provided for an easy reproduction of our experiments.

5.1.1. Classifiers

As standard benchmarking classification systems, we chose 1-Nearest-Neighbor (NN) and Support Vector Machine (SVM) classifiers. We make use of the LIBSVM [72] implementation library for the SVM classifiers. Weka and LIBSVM are freely available tools that permit to fully reproduce all the experiments presented. The 1NN classifier has no parameters; it simply uses the Euclidian distance

in the feature space to classify a test sample with the label of its closest prototype (where all training samples are kept as prototypes). For the SVM classifier, we chose to use a Gaussian kernel with fixed parameters that appeared to provide good results: gamma parameter is set to 10^{-2} and the slack variable C is set to 10^2. We decide not to optimize these parameters, since this operation is dataset-dependent.

5.1.2. Data partitioning

Since information about the writer identity is not always available (see section 3), we adopt different experimental settings on the datasets. For **HHReco**, **Sign**, **CVCsymb** and **LaViola**, a Writer-Independent (WI) Cross-Validation (CV) scheme is adopted, where each CV fold contains the data from one writer. The global WI recognition rate is then averaged from k CV experiments, where k is the number of writers. On **LaViola**, a predefined Train/Evaluation (T/E) partition is available on the data of each writer, and we use it for Writer-Dependent (WD) performance evaluation. On **HHReco**, **Sign** and **CVC-symb**, we run a 10-CV on the data from each writer and then average the performance over all the writers.

For datasets **ImiSketchS** and **Ironoff-digits**, we proceed respectively to a 5-CV (as suggested by the dataset partitioning) and a 10-CV experiments for WI performance, while WD tests are not feasible. Since no data partitioning is suggested by the authors of the database, we use a Stratified Random Cross-Validation strategy for generating all the CV folds.

For **NicIcon**, a partition of the samples is determined for both WD and WI settings. Since no optimization needs to be performed, we use both training and test sets for training the classifiers and the evaluation set for measuring the recognition performance (T+V/E).

The **ILG** dataset is exploited only in WD setting: we use the 3 first samples of each class as the training data (they correspond to the samples asked to the user at the initialization of the application in the collect process) and remaining samples for test. For each of the 28 experiments (one per writer), 21 classes are in competition, but these classes vary from a writer to another.

5.2. Baseline results

We first present the results obtained with the **HBF49** set of features on the 8 datasets, following the protocols exposed in the previous paragraph. Table 2 summarizes the recognition rates obtained for each experiment.

The obtained recognition performance first validates the ability of **HBF49** to properly handle objects of different nature and datasets of diverse difficulty in both WD and WI settings. Having a recognition rate above 85% for all the experiments with the simple NN classifier demonstrates a consistent behavior of the feature set over highly different application contexts. This illustrates the universality of **HBF49** and offers solid guarantees of its ability to deal with other datasets in the future. Note that

---

[1] http://www.irisa.fr/intuidoc/HBF49.html

Table 2: recognition rates obtained with **HBF49** feature set, on the 8 datasets, in both WD and WI modes. For each dataset, the number of classes is recalled (see table 1).

Dataset	#cl	exp	NN	SVM
Ironoff-digits	10	WI 10-CV	97.01	98.07
LaViola	48	WI-CV	91.40	93.64
		WD T/E	92.09	93.41
CVCsymb	25	WI-CV	89.50	96.58
		WD 10-CV	98.34	99.46
ImiSketchS	10	WI 5-CV	86.83	94.39
HHReco	13	WI-CV	86.26	92.55
		WD 10-CV	99.73	99.74
NicIcon	14	WI T+V/E	94.55	97.44
		WD T+V/E	98.87	99.30
Sign	17	WI-CV	99.09	99.56
		WD 10-CV	99.53	99.65
ILG	21	WD	93.38	93.54

whereas **HBF49** contains several rotation-sensitive features and no particular normalization is applied on these datasets, the recognition rates are still high for **HHReco** and **ImiSketchS**, where symbols from the same class can have a different orientation. Obviously, introduction of more orientation-independent features, or a specific normalization method can significantly improve these results, but the baseline method performs reasonably well even with this type of difficulty.

In accordance to what can be expected, the SVM classifier offers enhanced performance for all the experiments. With this classifier, **HBF49** representation is particularly efficient with the datasets **NicIcon** (multi-stroke gestures), **Ironoff-digits** (digits), **Sign** (single stroke gestures), and **CVCsymb** (multi-strokes symbols), with recognition rates above 96% and as high as 99.5%. WD settings generally lead to higher results in comparison with WI experiments, showing that the **HBF49** representation efficiently captures individual writing styles. For the **LaViola** dataset, where WI surprisingly performs better, we suspect that the high number of classes make it harder for the SVM classifier to generalize from a small training database (as in WD mode) than from a larger database (as in WI mode). In WD experiment on datasets **CVCsymb**, **HHReco**, **NicIcon** and **Sign**, recognition rates are extremely accurate (from 99.30% to 99.74%).

The datasets **Sign** and **ImiSketchS** were introduced quite recently, and our experiments establish a new baseline performance for future comparisons. Performance on **ImiSketchS** can be significantly improved by applying a rotation normalization as a pre-processing (for example by rotation the patterns based on their minimal bounding rectangle). The **Sign** database does not present difficulty for recognition methods, but it would be interesting to try reducing the number of features. A more compact representation is indeed desirable for building systems that can evolve to match the user needs and preferences [23]. For **CVCsymb**, to our knowledge no other results were reported on this classification task.

Table 3: Comparisons on **Ironoff-digits**. 1NN = Nearest Neighbor; FIS = Fuzzy Inference System; MLP = MultiLayer Perceptron; TDNN = Time-Delay Neural Network

	System	protocol	%
[57]	480 *star features*, 1NN	WI 3-CV	93.5
[56]	43 features, FIS	WI 0.5/0.5	95.6
	43 features, SVM	WI 0.5/0.5	95.5
[58]	Dynamic Time Warping, 1NN	n.a.	97.94
[47]	350 time-based features, MLP	n.a.	98.2
	350 time-based features, TDNN		98.4
	350 time-based features, SVM		**98.83**
	49 features **HBF49**, NN	WI 10-CV	97.01
	49 features **HBF49**, SVM	WI 10-CV	98.07

On the peculiar **ILG** dataset, our WD experiment leads to a recognition rate of 93.54% (with SVM). In this case where patterns only consist of single stroke gestures, a trajectory-based method such as Dynamic Time Warping may give better results. However, this experiment demonstrates that the representation offered by **HBF49** is quite efficient even for unpredictable patterns such as these user-defined gestures. Of course, the reduced number of training samples is also a strong limit to the classifier performance, and it is not surprising that the NN classifier, that involves no training step, performs comparably well. This is the first recognition performance reported on this dataset.

5.3. Comparative results

In this section, we confront our baseline method with other systems based on results reported in the literature on the four datasets **Ironoff-digits**, **LaViola**, **HHReco** and **NicIcon**. The comparison not only involves the overall performance of the systems, but also the systems complexity (number of features, combination of classifiers...). The first objective is to attest the goodness of baseline results reported in table 2, in comparison with state-of-the-art results. The second objective is to demonstrate the interest of the **HBF49** baseline representation for evaluating the interest of sometimes complex symbol recognition methods. Since it is often difficult to reproduce experiments from the literature so as to directly compare the performance of feature sets, this study merely compares our method as a complete recognition system (**HBF49** representation combined with a SVM or NN classifier) to results reported by other authors. This is one of the foreseen usages of the **HBF49** baseline, and it permits to evaluate the interest of any type of systems, feature-based or not.

5.3.1. Ironoff-digits

The table 3 presents results reported with the **Ironoff-digits** dataset, in a Writer-Independent context.

It shows that the baseline **HBF49** performance is comparable or better than most approaches. The best found reference, from the works of Ahmad et al. [47], with a recognition rate of 98.83%, is significantly higher than

Table 4: Comparisons on **LaViola** dataset. MSFT = Microsoft Classifier; T = predefined training set; E = predefined Evaluation set

	System	protocol	%
[15]	47 hybrid features, Adaboost	T/E WD	93.0
	86 features (47f + MSFT), Adaboost		**94.6**
	86 features, Adaboost + MSFT pruning		**94.6**
[41]	Freeman chaincodes, 1NN	T/E WD	92.1
	Reduced Freeman chaincodes, 1NN		92.0
[29]	Chebyshev coef 11th order, SVM	T/E WD	92.9
	180 features (sampling 30 points), SVM		94.5
	66 features (sampling 11 points), SVM		93.5
	49 features **HBF49**, NN	T/E WD	92.09
	49 features **HBF49**, SVM	T/E WD	93.41

Table 5: Comparisons on **HHReco** dataset. IDM = Image Deformation Model.

	System	protocol	%
[8]	Zernike moments 10th order, SVM	WI-CV	96.7
[9]	image templates, IDM distance	WI-CV	**98.2**
	image templates, IDM (no rotation)		95.2
[32]	visual parts	WI-CV	94.4
	49 features **HBF49**, NN	WI-CV	86.26
	49 features **HBF49**, SVM	WI-CV	92.55
[8]	Zernike moments 10th order, SVM	WD 10-CV	97.3
	49 features **HBF49**, NN	WD 10-CV	99.73
	49 features **HBF49**, SVM	WD 10-CV	**99.74**

Table 6: Comparisons on WI/WD tests with **NicIcon** dataset. T = Training set; V = Validation set; E = Evaluation set. 1NN = Nearest Neighbor; HMM = Hidden Markov Models; MCS = Multiple Classifier System.

	System	protocol	%
[14]	HMM	WI T/E	58.8
	Zernike moments features, SVM		70.3
	Combination HMM+SVM		83.1
[69]	Blurred Shape Models (offline)	WI	90.62
[63]	DTW	WI T/E	93.6
	28 global features, SVM		73
	30 trajectory-based features, SVM		90.30
	60 trajectory-based features, SVM		88.62
	545 selected features (mfsI), SVM		96.43
	MCS (4 SVM, 4 MLP, DTW)		**97.83**
[20]	257 visual features, 1NN	subset	96
	49 features **HBF49**, NN	WI T+V/E	94.55
	49 features **HBF49**, SVM	WI T+V/E	97.44
[69]	Blurred Shape Models (offline)	WD T/E	94.38
[63]	DTW	WD T/E	98.41
	660 selected features (mfsD), SVM		99.25
	MCS (4 SVM, 4 MLP, DTW)		**99.51**
	49 features **HBF49**, NN	WD T+V/E	98.87
	49 features **HBF49**, SVM	WD T+V/E	99.30

HBF49, but with 7 times more features. The **HBF49** performance is slightly higher than DTW result reported by [58], and significantly outperforms the two methods of Ragot and Anquetil [56]. The interest of high-dimensional *star feature* set introduced by Dinesh and Sridhar [57] is questionable when its performance is compared to our baseline result with NN classifier.

5.3.2. LaViola

Results about **LaViola** dataset, only available for the Writer-Dependent mode, are presented in table 4.

In all the reported experiments, the same data partitioning is adopted (following the partition suggested by the authors), insuring a perfect comparability of the performances. **HBF49** performs comparably with several of other methods, better than trajectory-based methods from Simistra et al. [41], and better than the Chebyshev features introduced in [29]. The baseline result with SVM classifier is only outperformed by a large set of 180 trajectory-based features [29], and by Laviola's systems with 86 features [15], that includes the Microsoft gesture Recognizer as a weak classifier in the Adaboost system.

5.3.3. HHReco

For the **HHReco** dataset, most of the methods only focus on the WI mode (see table 5).

With this experimental settings, the **HBF49** performance is significantly lower than competitors, because no specific treatment is performed for insuring orientation-independence. Actually, we noticed that in **HHReco**, the rotation variations are a mark of individual user style: for example one writer may draw his triangles pointing down while all others draw triangles that point up. This make the rotation variations very hard to handle in a WI context, since sometimes the rotation of samples from the testing set is completely different from all cases seen in the training set. The method of Ouyang and Davis [9] alleviates this problem by augmenting the training database with rotated versions of the training samples, and reaches a high performance of 98.2%. Without this treatment, their performance drops to 95.2%. In the works of Hse and Newton [8], only rotation-invariant features are utilized (Zernike moments), leading to a performance of 96.7%.

The results obtained on WD mode confirm that **HBF49** is perfectly able to handle the symbols from this dataset: a 99.74% accuracy rate is obtained, which is significantly higher than the performance obtained with rotation-invariant features from Hse's work.

5.3.4. NicIcon

Many methods have been proposed for recognizing symbols from the **NicIcon** dataset (see table 6).

In WI mode, the **HBF49** representation provides the second highest recognition rate found in the literature, only outperformed by a Multiple Classifier System (MCS) that combines 9 classifiers: 4 SVM classifiers with different features sets (respectively 28, 30, 60, and 550 features), 4 MLP classifiers with the same feature sets, and a DTW classifier. Results from [14] show that HMM is not a good representation for multi-strokes gestures of **NicIcon**. Strictly visual approaches, like the Zernike-based method

from [14] or the method of Almazan et al. [69] perform quite poorly as well. In the works of Tumen et al. [20], different types of features are also combined (Zernike moments, Image Deformation Model features, Shape Context features, Extended Trace Transform features) but no dynamic features are included, and optimal selection of 257 features leads to a recognition rate of about 96% (on a reduced dataset).

The rich set of 545 heterogeneous features proposed by the authors of the database performs well [63]. However, the significantly higher performance of **HBF49** permits to be doubtful about the interest of multiplying stroke-level features for representing multi-stroke gestures. In **HBF49**, the introduction of several global visual features (like zoning features and Hu moments features) seem to provide an important information that is not compensated by hundreds of stroke-level features. Likewise, the interest of combining classifiers such as HMM and SVM with Zernike features for mixing visual and dynamic information is clearly not established by the results from Arandjelovic et al. [14].

For the WD mode, fewer results are found in the literature. The strictly visual approach presented in [69] performs significantly lower than other methods, illustrating the importance of dynamic information for WD symbol recognition. This is confirmed by the high recognition rate reached by a DTW-based classifier [63]. Similarly to results in WI mode, the **HBF49** approach performs better than an optimized, high-dimensional feature set from [63]. Only the heavy MCS system presents a better performance, of 99.51%.

5.3.5. Synthesis

We have compared in this section the performance of **HBF49** (in combination with off-the-shelf SVM and NN classifiers) with state-of-the-art results reported in the literature over 4 datasets, including a total of 35 symbol recognition results from 14 references (all published between 2002 and 2011). Among the 35 comparisons, in 22 cases our method with SVM classifier performs better than the results reported from the literature. In one experimental setup (**HHReco** in WD), our method outperforms the best result reported in the literature, and in most cases our result is very close to the best reported ones, while our system is often much more simpler than competitors (in number of features or classifier complexity).

This not only proves that **HBF49** is an efficient baseline for evaluating new systems, but also demonstrates the interest of our constructive approach for universal feature set design. Indeed, we would like to emphasize that our universal feature set competes with systems that were designed, tuned and optimized for the recognition of specific datasets. For example, on the **NicIcon** dataset, **HBF49** outperforms most systems from [63] that have been optimized specifically on this dataset (in particular by employing feature selection). This proves that a careful constructive approach can yield a better description than selective

approaches.

6. Conclusion

In this paper, we have introduced a new set of simple features, chosen from different categories of features so as to constitute a reproducible baseline experiment for evaluation of future handwritten symbol or gesture recognition methods. **HBF49** was constructed based on empirical choices and careful data observation, with the constraint of maintaining a low dimensionality. Despite its simplicity and compactness, it performs very well on many benchmarking datasets, showing its ability to deal with many application contexts and its robustness with respect to patterns of diverse nature. Performance comparable or better than state-of-the-art is obtained with standard classifiers in several configurations with publicly available datasets. First landmarks are also established on several recently introduced datasets. It is expected that **HBF49** can be utilized by researchers as a benchmarking experiment for better highlighting their contributions to the field of hand-drawn pattern recognition. Transparency of our approach, details of the features extraction design, and presentation of our simple experimental protocol are aimed towards an easy exploitation of our work for this matter.

Additionally to baseline experiments, we believe that **HBF49**, designed as a generic representation of symbols, without consideration of drawing constraints or domain specificities, can serve as a basis for designing *universal* systems. As future user-centered interaction systems will offer more flexibility and give more liberty to the final user (ability to choose his personal gestures and customize the system), the recognition methods will have to be extensible and to deal with new unpredictable symbols. With its experimentally validated flexibility and its reasonably reduced size, **HBF49** is a good initial guess for designing symbol recognition systems of our near future.

7. References

[1] C. Tappert, C. Suen, T. Wakahara, The state of the art in online handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 787–808.

[2] C. Liu, S. Jaeger, M. Nakagawa, Online recognition of Chinese characters: the state-of-the-art, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 198–213.

[3] S. Jaeger, C. Liu, M. Nakagawa, The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition, International Journal on Document Analysis and Recognition 6 (2003) 75–88.

[4] R. Zanibbi, D. Blostein, J. Cordy, Recognizing mathematical expressions using tree transformation, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 1455–1467.

[5] T. Ouyang, R. Davis, Learning from neighboring strokes: Combining appearance and context for multi-domain sketch recognition, in: Proceedings of the 23rd Annual Conference Neural Information Processing Systems, 2009, Curran Associates, Inc, 2009, pp. 1401–1409.

[6] L. Wenyin, On-line graphics recognition: State-of-the-art, in: Proceedings of the 5th IAPR International Workshop on Graphics Recognition, 2003, pp. 291–304.

[7] W. Lee, L. Burak Kara, T. Stahovich, An efficient graph-based recognizer for hand-drawn symbols, Computers & Graphics 31 (2007) 554–567.

[8] H. Hse, A. Newton, Sketched symbol recognition using Zernike moments, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 367 – 370.

[9] T. Ouyang, R. Davis, A visual approach to sketched symbol recognition, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2009, pp. 1463–1468.

[10] D. Willems, R. Niels, M. van Gerven, L. Vuurpijl, Iconic and multi-stroke gesture recognition, Pattern Recognition 42 (2009) 3303–3312.

[11] D. Rubine, Specifying gestures by example, in: Proceedings of the 18th annual conference on Computer graphics and interactive techniques, 1991, ACM New York, NY, USA, pp. 329–337.

[12] L. Kara, T. Stahovich, An image-based, trainable symbol recognizer for hand-drawn sketches, Computers & Graphics 29 (2005) 501–517.

[13] F. Alimoglu, E. Alpaydin, Combining multiple representations and classifiers for pen-based handwritten digit recognition, in: Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1997, volume 2, IEEE, pp. 637–640.

[14] R. Arandjelović, T. Sezgin, Sketch recognition by fusion of temporal and image-based features, Pattern Recognition 44 (2011) 1225–1234.

[15] J. LaViola, R. Zeleznik, A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 1917–1926.

[16] A. Bragdon, E. Nelson, Y. Li, K. Hinckley, Experimental analysis of touch-screen gesture designs in mobile environments, in: Proceedings of the annual conference on Human factors in computing systems, 2011, ACM, pp. 403–412.

[17] Y. Li, Protractor: a fast and accurate gesture recognizer, in: Proceedings of the 28th international conference on Human factors in computing systems, 2010, ACM, pp. 2169–2172.

[18] J. Wobbrock, A. Wilson, Y. Li, Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes, in: Proceedings of the 20th annual ACM symposium on User interface software and technology, 2007, ACM, pp. 159–168.

[19] L. Anthony, J. Wobbrock, A lightweight multistroke recognizer for user interface prototypes, in: Proceedings of Graphics Interface, 2010, Canadian Information Processing Society, pp. 245–252.

[20] R. Tumen, M. Acer, T. Sezgin, Feature extraction and classifier combination for image-based sketch recognition, in: Joint Session of the Seventh Sketch-Based Interfaces and Modeling Workshop, 2010, Eurographics Association, pp. 63–70.

[21] C. Viard-Gaudin, P. Lallican, S. Knerr, P. Binter, The IRESTE On/Off (ironoff) dual handwriting database, in: Proceedings of the 5th International Conference on Document Analysis and Recognition, 1999, IEEE, pp. 455–458.

[22] R. Niels, D. Willems, L. Vuurpijl, The NicIcon database of handwritten icons, in: Proceedings of the 11th International Conference on the Frontiers of Handwriting Recognition, 2008, pp. 296–301.

[23] A. Almaksour, E. Anquetil, S. Quiniou, M. Cheriet, Personalizable pen-based interface using lifelong learning, in: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition, 2010, IEEE, pp. 188–193.

[24] A. Almaksour, E. Anquetil, S. Quiniou, M. Cheriet, Evolving fuzzy classifiers: Application to incremental learning of handwritten gesture recognition systems, in: Proceedings of the 20th International Conference on Pattern Recognition, 2010, pp. 4056–4059.

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, The WEKA data mining software: an update, ACM SIGKDD Explorations Newsletter 11 (2009) 10–18.

[26] A. Jain, D. Zongker, Representation and recognition of handwritten digits using deformable templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 1386–1390.

[27] C. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: benchmarking of state-of-the-art techniques, Pattern Recognition 36 (2003) 2271–2285.

[28] M. Cheriet, N. Kharma, C.-L. Liu, C. Suen, Character Recognition Systems: A Guide for Students and Practitioners, John Wiley & Sons, 2007.

[29] B. Keshari, S. Watt, Online mathematical symbol recognition using SVMs with features from functional approximation, in: Proceedings of Mathematical User-Interfaces Workshop, 2008.

[30] M. Fonseca, J. Jorge, Using fuzzy logic to recognize geometric shapes interactively, in: Proceedings of the 9th IEEE International Conference on Fuzzy Systems, 2000, volume 1, IEEE, pp. 291–296.

[31] A. Apte, V. Vo, T. Kimura, Recognizing multistroke geometric shapes: an experimental evaluation, in: Proceedings of the 6th annual ACM symposium on User interface software and technology, 1993, ACM, pp. 121–128.

[32] M. Oltmans, Envisioning sketch recognition: a local feature based approach to recognizing informal sketches, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[33] J. Mas, B. Lamiroy, G. Sanchez, J. Llados, Automatic adjacency grammar generation from user drawn sketches, in: Proceedings of the 18th International Conference on Pattern Recognition, 2006, pp. 1026–1029.

[34] T. Hammond, B. Paulson, Recognizing sketched multistroke primitives, ACM Transactions on Interactive Intelligent Systems 1 (2011) 4:1–4:34.

[35] C. Calhoun, T. Stahovich, T. Kurtoglu, L. Kara, Recognizing multi-stroke symbols, in: AAAI Spring Symposium on Sketch Understanding, 2002, pp. 15–23.

[36] A. Delaye, E. Anquetil, Fuzzy relative positioning templates for symbol recognition, in: Proceedings of the 11th International Conference on Document Analysis and Recognition, 2011, pp. 1220–1224.

[37] A. Hall, C. Pomm, P. Widmayer, A combinatorial approach to multi-domain sketch recognition, in: Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling, 2007, ACM New York, NY, USA, pp. 7–14.

[38] T. Hammond, R. Davis, Ladder, a sketching language for user interface developers, Computers & Graphics 29 (2005) 518–532.

[39] L. Gennari, L. Kara, T. Stahovich, K. Shimada, Combining geometry and domain knowledge to interpret hand-drawn diagrams, Computers & Graphics 29 (2005) 547–562.

[40] S. Macé, E. Anquetil, Eager interpretation of on-line hand-drawn structured documents: The DALI methodology, Pattern Recognition 42 (2009) 3202–3214.

[41] F. Simistira, V. Katsouros, G. Carayannis, A template matching distance for recognition of on-line mathematical symbols, in: Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, 2008.

[42] S. Connell, A. Jain, Template-based online character recognition, Pattern Recognition 34 (2001) 1–14.

[43] J. Hu, M. Brown, W. Turin, HMM based online handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 1039–1045.

[44] T. Sezgin, R. Davis, HMM-based efficient sketch recognition, in: Proceedings of the 10th international conference on Intelligent user interfaces, 2005, ACM, pp. 281–283.

[45] D. Anderson, C. Bailey, M. Skubic, Hidden Markov Model symbol recognition for sketch-based interfaces, in: AAAI Fall Symposium, 2004, pp. 15–21.

[46] M. Kherallah, L. Haddad, A. Alimi, A. Mitiche, On-line handwritten digit recognition based on trajectory and velocity modeling, Pattern Recognition Letters 29 (2008) 580–594.

[47] A. Ahmad, M. Khalia, C. Viard-Gaudin, E. Poisson, Online handwriting recognition using Support Vector Machine, in: Proceedings of the IEEE Region 10 Conference TENCON, 2004, IEEE, pp. 311–314.

[48] C. Bahlmann, Directional features in online handwriting recognition, Pattern Recognition 39 (2006) 115–125.

[49] B. Chakraborty, G. Chakraborty, A new feature extraction technique for on-line recognition of handwritten alphanumeric characters, Information Sciences 148 (2002) 55–70.

[50] O. Golubitsky, S. Watt, Online recognition of multi-stroke symbols with orthogonal series, in: Proceedings of the 10th International Conference on Document Analysis and Recognition, 2009, IEEE, pp. 1265–1269.

[51] B. Paulson, P. Rajan, P. Davalos, R. Gutierrez-Osuna, T. Hammond, What!?! no Rubine features?: using geometric-based features to produce normalized confidence values for sketch recognition, in: HCC Workshop: Sketch Tools for Diagramming, 2008, pp. 57–63.

[52] Z.-L. Bai, Q. Huo, A study on the use of 8-directional features for online handwritten Chinese character recognition, in: Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005, volume 1, pp. 262–266.

[53] C.-L. Liu, X.-D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, in: Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, 2006.

[54] M. Liwicki, H. Bunke, IAM-OnDB-an on-line English sentence database acquired from handwritten text on a whiteboard, in: Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005, IEEE, pp. 956–961.

[55] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, CASIA online and offline chinese handwriting databases, in: Proceedings of the 11th International Conference on Document Analysis and Recognition, 2011, pp. 37 –41.

[56] N. Ragot, E. Anquetil, A generic hybrid classifier based on hierarchical fuzzy modeling: Experiments on on-line handwritten character recognition, in: Proceedings of the 7th International Conference on Document Analysis and Recognition, 2003, pp. 963–967.

[57] M. Dinesh, M. Sridhar, A feature based on encoding the relative position of a point in the character for online handwritten character recognition, in: Proceedings of the 9th International Conference on Document Analysis and Recognition, 2007, IEEE, pp. 1014–1017.

[58] M. Dinesh, N. Sharma, Lipi alphanumeric character recognizer, http://lipitk.sourceforge.net/lipi-reco-char-alphanum_1_0.htm, 2002.

[59] J. M. Romeu, A Syntactic Pattern Recognition Approach based on a Distribution Tolerant Adjacency Grammar and a Spatial Indexed Parser. Application to Sketched Document Recognition, Ph.D. thesis, Centre de Visió per Computador, 2010.

[60] Imisketchsdb, http://www.irisa.fr/intuidoc/IMIsketchSDB.html, 2012.

[61] A. Ghorbel, E. Anquetil, A. Lemaitre, Optimization analysis based on a breadth-first exploration for a structural approach of sketch interpretation, in: Proceedings of the 10th IAPR International Workshop on Document Analysis Systems, 2012, pp. 240–244.

[62] A. Ghorbel, A. Lemaitre, E. Anquetil, Competitive hybrid exploration for off-line sketches structure recognition, in: Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition, 2012. To appear.

[63] L. Vuurpijl, D. Willems, R. Niels, M. van Gerven, Design issues for pen-centric interactive maps, in: R. Babuka, F. C. Groen (Eds.), Interactive Collaborative Information Systems, volume 281 of *Studies in Computational Intelligence*, Springer-Verlag New York Inc, 2010, pp. 273–297.

[64] N. Renau-Ferrer, P.-Y. Li, A. Delaye, E. Anquetil, The IL-GDB database of realistic pen-based gestural commands, in: Proceedings of the 21st International Conference on Pattern Recognition, 2012. To appear.

[65] A. Lemieux, C. Gagné, M. Parizeau, Genetic engineering of handwriting representations, in: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition, 2002, IEEE, pp. 145–150.

[66] D. Willems, R. Niels, Definitions for Features used in Online Pen Gesture Recognition, Technical Report, NICI, Radboud University Nijmegen, 2009.

[67] E. Anquetil, G. Lorette, Perceptual model of handwriting drawing. Application to the handwriting segmentation problem, in: Proceedings of the 4th International Conference on Document Analysis and Recognition, 1997, volume 1, pp. 112–117.

[68] S. Impedovo, G. Pirlo, R. Modugno, A. Ferrante, Zoning Methods for Hand-Written Character Recognition: An Overview, in: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition, 2010, IEEE, pp. 329–334.

[69] J. Almazán, A. Fornés, E. Valveny, A non-rigid feature extraction method for shape recognition, in: Proceedings of the 11th International Conference on Document Analysis and Recognition, 2011, IEEE, pp. 987–991.

[70] M. Hu, Visual pattern recognition by moment invariants, Information Theory, IRE Transactions on 8 (1962) 179–187.

[71] R. Graham, An efficient algorithm for determining the convex hull of a finite planar set, Information processing letters 1 (1972) 132–133.

[72] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27.