The ILGDB database of realistic pen-based gestural commands

Ney Renau-Ferrer, Peiyu Li, Adrien Delaye, Eric Anquetil

The ILGDB database of realistic pen-based gestural commands

Ney Renau-Ferrer, Peiyu Li, Adrien Delaye, Eric Anquetil
INSA de Rennes
UMR IRISA, Campus de Beaulieu, F-35042 Rennes
Université Européenne de Bretagne, France

Abstract

In this paper, we introduce the Intuidoc-Loustic Gestures DataBase (ILGDB), a new publicly available database of realistic pen-based gestures for evaluation of recognition systems in pen-enabled interfaces. ILGDB was collected in a real world context and in an immersive environment. As it contains a large number of unconstrained user-defined gestures, ILGDB offers a unique diversity of content that is likely to serve as a precious tool for benchmarking of gesture recognition systems. We report first baseline experimental results on the task of Writer-Dependent gesture recognition.

1 Introduction

Pen-based and tactile interaction is a type of human-computer interaction which is increasingly difficult to escape. With the popularization of smartphones and tabletPCs, users get more and more familiar with interfaces admitting gestural commands, i.e. where they can perform pen-based or touch-based strokes for triggering specific commands. If many works were conducted for recognition of pen-based gestures, few benchmarking tools are available for gesture recognition. For many years, benchmarks for text recognition have been proposed (e.g. ironoff [4]...), but it is only recently that benchmarks were introduced for the evaluation of pen gestures recognition systems: SIGN dataset of single-stroke gestural commands [1], NicIcon database of multi-strokes pen-based gestures [3]. Existing gestural commands databases however suffer from two strong limitations. First, they only contain a limited vocabulary of gestures: 14 classes in NicIcon, 17 classes in SIGN. This is an important flaw because future pen-based interfaces will allow users to customize their own sets of commands, and recognition systems will have to handle any kind of pen gestures (*open-world* context) [1]. The second limitation is the bias induced by an unrealistic collect process. Participants to the NicIcon collect were for example asked to fill forms with an electronic pen [3], with no feedback permitted, and no indication given about correctness of the gestures. Additionnaly to the shift from electronic pen to pen-enabled interface, such a copying task is far from a real case usage, and important effects are neglected: user memorizing and getting accostumed to gestures and writing device over time, adaptation to the system...Existing databases therefore do not reflect the reality of what a user may produce in a real usage, and thus do not permit a proper evaluation of gesture recognition systems.

In this paper we present **ILGDB**, a database of about 7,000 pen-based gestures that have been collected from 40 users in real-life situation, immersed in a simulated application where gestural commands trigger simulated effects. The two characteristics of this database are a high diversity of contents (including numerous user-defined gestures) and an original collecting process that guarantees a realistic content. This database was built for two main reasons: firstly, to evaluate users capability to memorize gestural commands according to different types of vocabulary (user-defined gestures, predefined gestures, semi-customized gestures)[2]; secondly, to gather realistic data for constituting a reference benchmark for gesture recognition systems.

In the next section, we describe the simulated application that was used for collecting gestures, then we present the collection procedure in more details. The database contents is more precisely presented in section 4. In the last section, we introduce baseline classification results obtained on this database on the task of writer-dependent gesture recognition.

2 Simulated application: Picture Editor

In order not to introduce any bias in the experiment, we didn't want users to be aware that they were supposed to learn gestural commands, and they were not aware of the fact that gestures were being collected.

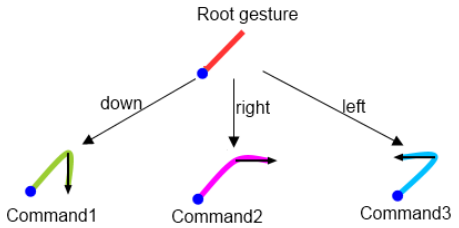


Figure 1. SCGC for a gesture family

We have developed a pen-based picture editor software where users can perform various actions by using graphic gestures. Our goal was to deceive the user by letting him think he was testing our interface and evaluating the relevance of gestural commands for manipulating pictures. The application developed is a simple image organizer, viewer and editor allowing users to perform actions by simply drawing gestures on the images. A help menu is available for assisting the user in retrieving available gestures. Also, since the procedure is fully supervised under a predefined scenario, we can detect whenever the user draws an incorrect gesture. We have implemented 21 commands in the application, organized in 7 families of actions. Examples include: *Apply a black and white effect* (from the *effect* family) or *Add a classic frame* (*frame* family).

3 Collection procedure

Acquisition process for one user involves 5 phases of using the application on a TabletPC: an initialization phase, and 2 use phases punctuated with 2 test phases. During initialization, users have to draw the 21 gestures 3 times each in order to train themselves. In each use phase, 34 questions asked users to perform commands by drawing gestures on appropriate images (for example *"Apply black and white effect on image 4"*). During use phases, users can access a help window presenting all the available gestures. Some commands are asked more than the others: 6 of them are asked 3 times, 4 of them are asked twice, 8 of them only once. During test phases, users are asked to perform a gesture on a blank area (for example *"Perform the gesture corresponding to: send by email"*). Each test phase includes 21 questions (one for each command), without help menu. Table 1 presents the number of samples collected during a session, where classes are gathered into columns depending on their frequency (e.g. first column shows that for 6 classes 11 samples are collected).

Users were divided into 3 distinct groups. Users of the group 1 had to define gestures of their choice for each of the 21 commands. Users from group 2 only had to draw a *root gesture* for each of the 7 command families, then full gestures were automatically generated based on this root gesture. Full gestures were generated by extending the end of the stroke following three directions: down, right, left (as shown in Figure 1). This type of gestural commands are called Semi-Customizable Gestural Commands (SCGC) in the sequel. Users from group 3 were imposed a set of 21 predefined gestures, generated once for all the users by the same principle of family root extension.

Table 1. Repetitions of gestures by phase.

# of classes	6	4	8	3
Initialization	3	3	3	3
Use 1	3	2	1	0
Test 1	1	1	1	1
Use 2	3	2	1	0
Test 2	1	1	1	1
Total	11	9	7	5

Table 2. Overview.

Dataset	#Writers	#Classes	#Samples
Group 1	21	21 x 21 = 441	3,653
Group 2	7	7 x 21 = 147	1,003
Group 3	12	21	2,310
Total	40	609	6,976

4 Data

A total of 40 persons participated in the data collection: 21 users in *Group 1*, 7 in *Group 2* and 12 in *Group 3*. The database is composed of 6,976 gestures in total. Table 2 summarizes the repartition of classes, writers and samples. This table emphasizes the originality of **ILGDB**, which includes a very large number of classes. Since users of *Group 1* have defined their own gestures, the number of classes is potentially as high as 441. Obviously, among the 441 classes, some actually have the same shape: for example, a heart-shaped gesture was frequently chosen by users for triggering a specific command. Conversely, some identical gestures might also have been chosen for different actions, making apparently identical gestures having different labels. The same properties can be observed for the data from *group 2*, where SCGC gestures are generated from user-defined roots. These characteristics qualify gestures collected with *group 1* and *group 2* only for running Writer-Dependent experimentations, while

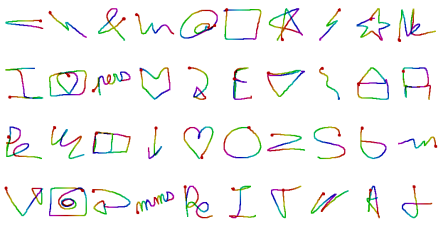


Figure 2. Customized gestures (*group 1*).

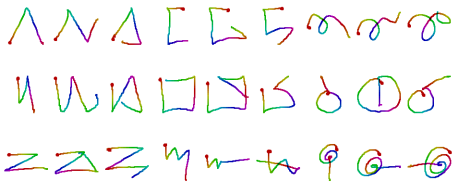


Figure 3. SCGC gestures (*group 2*).

group 3 can also be used in a Writer-Independent context. However, the large diversity of gestures makes this database a very useful benchmark for attesting polyvalence of gesture recognition methods. Rather than optimization on a small tuned dataset of 21 gestures, we offer the possibility to evaluate recognition systems in a realistic *open-world* context.

4.1 Diversity of the content

The only constraint imposed to the users for designing their own gestures was to perform a single-stroke gesture. As a result, the gestures from *group 1* dataset can be considered as a good approximation of the possibilities offered by single stroke gestures. The diversity of the content from this dataset can be appreciated in figure 2, that shows 50 examples from different writers and different classes. Diversity in the gesture complexity is obvious. Some gestures are very simple, for example basic geometric shapes, characters, waves; some are more elaborate, like arrows or stars; some are really complex, like handwritten sequence of letters, or retraced polygons. In figure 3, the samples are organized by sequences of three gestures from the same family, resulting from a common root chosen by the user. It presents 27 examples of SCGC from *group 2*, where a large diversity can be observed as well. Finally, the static dictionary of 21 classes utilized in *group 3* is presented in 4. It can be noticed that the set of predefined gestures only involves simple gestures, with few changes in writing direction and no retracing.

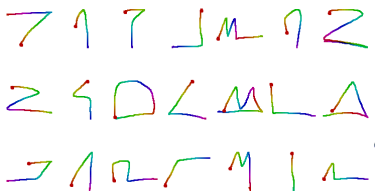


Figure 4. Predefined gestures (*group 3*).

5 Recognition experiments

ILGDB is made available to the community to facilitate realistic evaluation of gesture recognition methods. In this section, we present first baseline experiments that can be useful for future comparisons. These experiments also provide a quantified measure of the database difficulty. In the next paragraph we present the set of features we designed for accurate description of the gestures.

5.1 Feature extraction (HBF49)

The feature set design is critical, as it should be able to represent any type of single stroke gestures. A reasonably low-dimension representation is required to permit learning of classifiers from few training samples. Indeed, only 3 training samples are available for each class. This constraint is also realistic: good customizable gesture systems should only need few training samples from the user. We exploit the HBF49 representation[1] combining dynamic features (positions of first and last points, initial angle...) and static features (box aspect ratio, length, curvature, 2D histogram, Hu moments...). This set of 49 features altogether guarantees a good polyvalence of the representation.

5.2 Recognition results

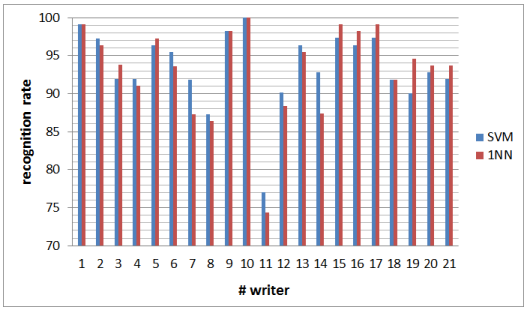
All the experiments are conducted in a Writer-Dependent (WD) mode. From the data of each writer, the samples realized at the initialization phase are considered as training samples. Two different types of classifiers are used: a simple Nearest-Neighbor (1NN) classifier (with all training samples kept as class prototypes) and a Support Vector Machine (SVM), with a Gaussian kernel. Table 3 reports the recognition rates averaged over the writers from each group.

It first appears that the 1NN and SVM behave quite comparably over the database, with slightly better performances obtained with SVM. On one hand, the over-

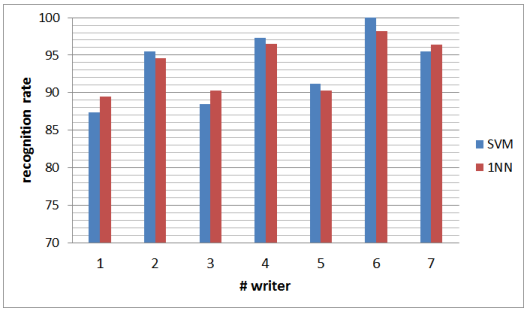
[1] See http://www.irisa.fr/intuidoc/HBF49.html

Table 3. Writer-dependent recognition results.

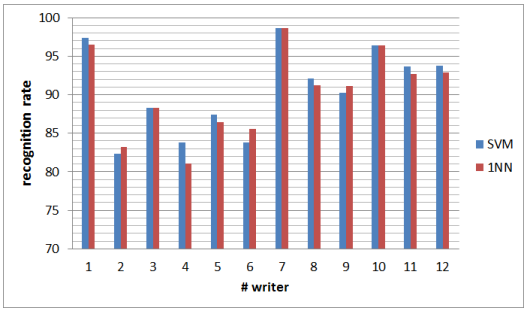
Dataset	% 1NN	% SVM
Group1	93.26	93.48
Group2	93.75	93.75
Group3	90.57	90.80
Total	92.52	92.70



(a) *group 1*



(b) *group 2*



(c) *group 3*

Figure 5. Recognition rate per writer.

all performance of about 92% of good recognition indicates that **ILGDB** is challenging for feature-based approaches. On the other hand, the consistency of results over the three datasets is remarkable, and demonstrates that the chosen feature set is very efficient. Surprisingly, the most difficult dataset is the *group 3*, where only predefined gestures where used. Figure 5 represents the per-writer results for the same experiments. The charts from (a) demonstrate that a good stability of recognition accuracy can be reached even in the case of user-defined gestures. The lowest rates are obtained for writer 11, for which rate drops to about 75 % with both classifiers. For all other users, the rates are above 85 % with the two classifiers. The same stability can be noticed from writers of *group 2* (b), with recognition rates above 85% for everyone, in the context of SCGC gesture vocabulary. Results from *group 3* (c) show that having a stable dictionary of classes does not make the WD recognition task easier. Obviously, these results could be improved by tuning the recognition system to this specific set of gestures (for example by dataset-oriented design of features).

6 Conclusion

We have presented **ILGDB**, a new database of pen-based gestural commands that can be found at http://www.irisa.fr/intuidoc/ILGDB.html. The sets of 49 features extracted in our experiments are also available for download. The database contains a large quantity of gestures drawn by 40 users, showing a great diversity of content (including user-defined gestures), collected in a realistic process within an immersive environment, ensuring that the gestures are as close as possible to what can be produced by users in a real application. The temporal order of gestures is preserved so one can track the distortions applied by a user to a gesture over time, considering the memorization effect that arises in real life usage.

References

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

[2] P. Li, N. Renau-Ferrer, E. Anquetil, and E. Jamet. Semi-customizable gestural commands approach and its evaluation. In *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition*, 2012. in press.

[3] R. Niels, D. Willems, and L. Vuurpijl. The nicicon database of handwritten icons. In *Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition*, pages 296–301, 2008.

[4] C. Viard-Gaudin, P. Lallican, S. Knerr, and P. Binter. The ireste on/off (ironoff) dual handwriting database. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pages 455–458. IEEE, 1999.