Quantifying Spatial Relations to Discover Handwritten Graphical Symbols

Jinpeng LI, Harold MOUCHERE, Christian VIARD-GAUDIN

IRCCyN (UMR CNRS 6597) - L'UNAM - Université de Nantes, France
{jinpeng.li,harold.mouchere,christian.viard-gaudin}@univ-nantes.fr

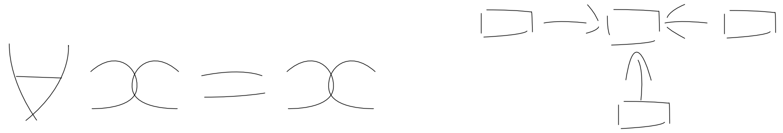
ABSTRACT

To model a handwritten graphical language, spatial relations describe how the strokes are positioned in the 2-dimensional space. Most of existing handwriting recognition systems make use of some predefined spatial relations. However, considering a complex graphical language, it is hard to express manually all the spatial relations. Another possibility would be to use a clustering technique to discover the spatial relations. In this paper, we discuss how to create a relational graph between strokes (nodes) labeled with graphemes in a graphical language. Then we vectorize spatial relations (edges) for clustering and quantization. As the targeted application, we extract the repetitive sub-graphs (graphical symbols) composed of graphemes and learned spatial relations. On two handwriting databases, a simple mathematical expression database and a complex flowchart database, the unsupervised spatial relations outperform the predefined spatial relations. In addition, we visualize the frequent patterns on two text-lines containing Chinese characters.

Keywords: Spatial Relations, Unsupervised Learning, Graphical Symbol Discovery, On-line Handwriting

1. INTRODUCTION

All communication is based on the fact that the participants share conventions that determine how messages are constructed and interpreted. For graphical communication these conventions indicates how arrangements, or layout, of graphical objects encode information. For instance, graphical languages (sketch, mathematical or chemical expressions, etc.) are composed of a set of symbols within some constraints. These constraints could be the grammar of this language, the layout of symbols, and so on. Furthermore, the symbols are also composed of a layout of strokes. In this paper, the layout means that the elements (symbols, strokes) are arranged in the 2 dimensional space, so that we can build a coherent document. Figure 1 illustrates two handwritten documents, a handwritten mathematical expression and a handwritten flowchart. The spatial relations specify how these elements are located in the layout.



(a) (b)

Figure 1: Two different handwritten graphical documents: (a)a handwritten mathematical expression, (b)a handwritten flowchart.

As an example, suppose that we have a set of two different shapes of strokes called *graphemes* $\{\backslash, /\}$. We assume that these graphemes are well detected by a clustering algorithm [1]. Using these two graphemes, we can compose two different symbols:"$\wedge$" and "$\vee$". The only difference between "$\wedge$" and "$\vee$" is the spatial relation; "$\backslash$" is put on the right side in "$\wedge$" and on the left side in "$\vee$". These spatial relations, left and right are easily defined manually.

With more graphemes and more spatial relations, it is possible to design new symbols. For instance, using the set of graphemes $\{\backslash, -, /\}$, we compose the symbol "$\forall$" with the three strokes "$\backslash_{(1)}$", "$-_{(2)}$", and "$/_{(3)}$". We can say "$-_{(2)}$" is *between* "$\backslash_{(1)}$" and "$/_{(3)}$". In this case, *between* implies a relationship among three strokes which is the cardinality of this spatial relation [2]. In this paper, we limit the cardinality of spatial relation to two strokes, from a reference stroke to an argument stroke. However, with only 3 strokes, we have to consider

6 different pairs of strokes to envisage all appropriate alternatives, for example "$\backslash_{(1)} \rightarrow -_{(2)}$", "$-_{(2)} \rightarrow \backslash_{(1)}$", "$\backslash_{(1)} \rightarrow /_{(3)}$", etc. The number of spatial relation couples will grow rapidly with the increasing number of strokes in a layout [1]. In this paper, we focus on the automatic modeling of these spatial relations: can we learn the spatial relations as we learn shapes of strokes?

A traditional modeling of spatial relation is represented in three levels [2]: the topological relations, the orientation relations, and the distance relations. The topological characteristics are preserved under topological transformations for example translation, rotation, and scaling. A simple example of topological relation is the intersection of two strokes. The orientation relations calculate the directional information between two strokes [3]. For instance the stroke A is on the right of the stroke B. The distance relations describe how far two strokes are.

Most of existing systems dealing with handwriting need some spatial relations between strokes. For instance, [3] uses a fuzzy relation position (orientation relations) for the analysis of diacritics on on-line handwritten text. In [4], authors add a distance information to design a structural recognition system for Chinese characters. In the context of handwritten mathematical expression recognition in [5], authors use the three levels of spatial relations to create a Symbol Relation Tree (SRT) using six predefined spatial relations: inside, over, under, superscript, subscript and right. Spatial relations are also useful in automatic symbol extraction as in our previous works in [1, 6]. In short, we extract automatically the graphical symbols from a graphical language with a simple set of predefined spatial relations. Our approach was tested successfully on a simple mathematical expression database. We predefined three domain specific relations (right, below, and intersection) to create a relational graph between strokes. The creation of this relational graph starts with the top-left stroke because of the left to right handwriting orientation. In the relational graph, the repetitive sub-graphs composed of graphemes and predefined spatial relations are considered as graphical symbols.

Using a simple set of predefined spatial relations is obviously not enough for describing "a new" (or "an unknown") complex graphical language. We may lose some unknown spatial relations which are important for a specified graphical language. Let us consider the difference between 9 different layouts of the 2 previous strokes $\{\backslash, /\}$: "$\backslash/$" "$\vee$", "$/\backslash$", "$\wedge$", "$\leq$", "$<$", "$\geq$", "$>$" and "$\times$". We want to distinguish these 9 layouts. We assume "$\backslash$" as the reference stroke and "/" as the argument stroke. If we categorize these layouts by intersection, two groups will be obtained: {"$\backslash/$","$/\backslash$", "$\leq$", "$\geq$"} and {"$\vee$", "$\wedge$", "$<$", "$>$", "$\times$"}. If we categorize these layouts by four predefined directions (right, left, above, and below) of "$\backslash$", four groups will be obtained: {"$\backslash/$","$\vee$"}, {"$\wedge$", "$/\backslash$"}, {"$\leq$", "$<$"}, and {"$\geq$", "$>$"} with the confusing layout "$\times$". The combination of left (directional relations) and intersection (topological relations) allows the distinction of these layouts. However, there are many combinations of spatial relations in a complex graphical language. It is hard to predefine manually all the useful combinations of spatial relations.

In this paper, we introduce the proposed strategy extended from our previous works [1, 6] in Section 2. The contribution of this paper is to discuss how to use a clustering technique to discover the spatial relations rather than some predefined spatial relations in previous works after the construction of relational graphs. Then, we use the learned spatial relations to discover the graphical symbols (sub-graphs) in three different domains.

2. SYSTEM OVERVIEW

We give an overview of our proposed strategy to construct the relational graph between strokes, which will be processed by a graph mining technique to discover the handwritten graphical symbols. Our proposed strategy is divided into four main steps: i) the pre-processing of strokes, ii) the construction of relational graphs between strokes, iii) the quantization of spatial relation couples using a clustering technique, iv) and the handwritten graphical symbol discovery using a graphs mining algorithm [1, 6, 7].

2.1 Pre-processing of strokes

First of all, given some graphical evidences (a set of strokes $\{str_i\}$), a strokes pre-processing is required. Since the strokes may be collected by different input devices or written by different individuals, the same layout may be found at different scales. We define a graphical sentence as a layout organized by a set of strokes where this set contains only full graphical symbols, not a part of symbol. Each graphical sentence is independent of the other graphical sentences in terms of the spatial relations, but all the graphical sentences use the same graphical symbol set. For example, a set of graphical sentences can be a set of pages which are homogenous but independent. Figure 2 shows two graphical sentences produced by a handwritten flowchart language. Two graphical sentences are independent in terms of the spatial relations. All the strokes in a graphical sentence have to be normalized by a local unit size. We analyze the spatial relations on this normalized layout. Considering the shape of strokes, we use a hierarchical clustering to regroup the strokes into a finite set of n_g graphemes, named *codebook*, and the strokes are labeled with the nearest grapheme [6]. This step is the quantization of strokes. In the generation of codebook, we do not consider the size of strokes, but only the shape of strokes is considered. In next sections, we construct firstly the relational graph between strokes and quantify secondly the spatial relations (edges in the relational graph) to discover graphical symbols.

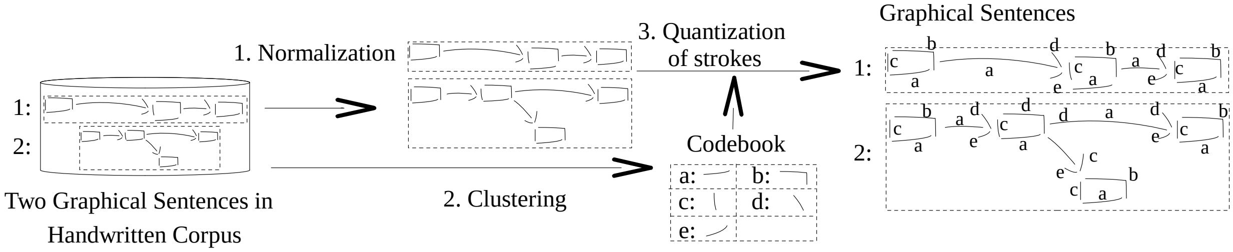

1. Normalization
3. Quantization of strokes
Graphical Sentences
1:
2:
Codebook
a:
b:
c:
d:
e:
Two Graphical Sentences in Handwritten Corpus
2. Clustering

Figure 2: Stroke Pre-processing

2.2 Construction of the Relational Graph

Given a graphical sentence, we describe the layout of strokes with a relational graph. We consider the nodes as the strokes and the edges as the spatial relations, as shown in Figure 3. A spatial relation is defined from a reference stroke to an argument stroke. In other words, the edge is directed. This allows for instance to distinguish between two different layouts, such as "$->$" and "$>-$". To select a couple of reference stroke and argument stroke, we have to account for the limited visual angle of the human vision system [8]. Consequently, the strokes which are far away from the reference stroke are not necessary to be linked with the reference stroke. We prefer the symbols composed of the closest strokes since the symbols are naturally defined by human. We create the directed edges from a reference stroke to the n_{cstr} closest strokes. The distance for the closest strokes is defined by the Euclidean distance between two closest points in the two sequences of points (the two strokes) respectively. Formally, suppose that we have two strokes, $str_x = (..., pt_i, ...)$ and $str_y = (..., pt_j, ...)$ where pt_i and pt_j are the points in the strokes, we define the distance between two strokes as:

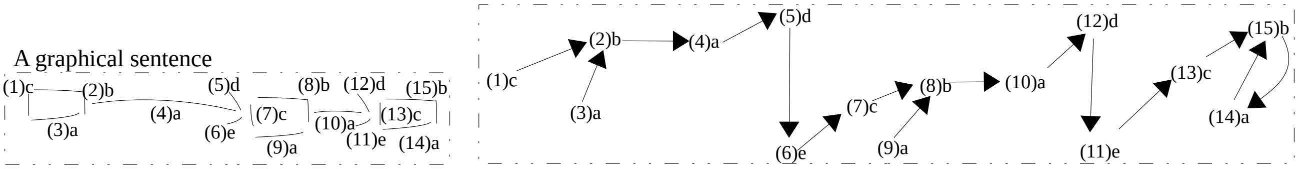
$$dist_{str}(str_x, str_y) = \min_{pt_i \in str_x, pt_j \in str_y} dist_{pt}(pt_i, pt_j) \tag{1}$$

where $dist_{pt}(pt_i, pt_j)$ is the Euclidean distance. Therefore, considering a reference stroke str_{ref}, we can find the closest stroke, $CStr(str_{ref}) = \underset{str_p \in \{str_i\}}{\arg\min}\ dist_{str}(str_{ref}, str_p)$ where $CStr(str_{ref})$ is not necessary equal to $CStr(CStr(str_{ref}))$. At the end, we can extract the spatial relation couples from the edges of relational graph for the clustering to find the spatial relation prototypes.

For instance, we consider the stroke (1) as the reference stroke in Figure 3. We can see that the reference stroke (1) has intuitively some obscure symbol relationships with the nearby strokes (2) and (3). As an example, we choose the $n_{cstr} = 1$ closest stroke to create the relational graph. The relation graph shows that $(CStr(Stroke(1)) = Stroke(2)) \neq (CStr(Stroke(2)) = Stroke(4))$. Since we have 15 edges in the relational graph, we have 15 spatial relation couples for the clustering. In next section, we vectorize the spatial relations for the clustering.

2.3 Spatial Relation Vectorization for Clustering and Quantization

We create a relational graph for the graphical sentence and many spatial relation couples are extracted. In this section, we vectorize the spatial relation between two strokes. We extract firstly the features of spatial relations. The spatial relation can be represented in three levels: distance relations, orientation relations, and topological relations [2]. In our case, the distance relation describes how far an argument stroke is from a reference stroke. We use the $dist_{str}(str_{ref}, str_{arg})$ as the *distance* relation (1 feature). The orientation relations illustrate some



(a) A graphical sentence (b) Corresponding relational graph

Figure 3: Creation of the relational graph (b) for a graphical sentence (a)

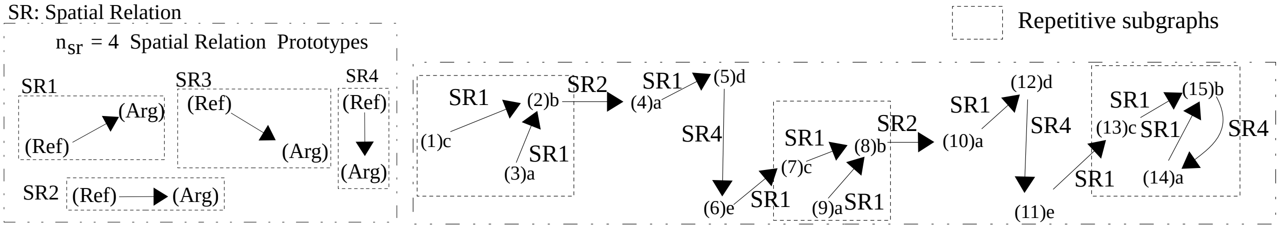
directional relations. A fuzzy model of directional relations between two strokes can be applied [3]. Therefore, we choose four fuzzy directional relations, *right*, *left*, *above*, and *below* which define 4 features in the range [0,1]. Concerning the topological relations, many different relations may be considered [9]. We use only the topological relation of *intersection* (1 binary feature). Since the size of grapheme is ignored in the step of quantization of strokes, we add a *relative size* from a reference stroke to an argument stroke (1 feature). We define the diagonal length of the bounding box of the stroke str_i as $Dig(str_i)$. The relative size from a reference stroke to an argument stroke is $RS(str_{ref}, str_{arg})$ $=Dig(str_{arg})/Dig(str_{ref})$. Table 1 summarizes the value range of each feature. Four groups of features are extracted: the Relative Size (S), the Distance (D), the Fuzzy Directions (F4), and the Intersection (I).

	Relative Size (S)	Distance(D)	Fuzzy Directions (F4)				I
	$RS(str_{ref}, str_{arg})$	$dist_{str}(str_{ref}, str_{arg})$	Right	Left	Above	Below	Intersection
Range	$(0, +\infty)$	$[0, +\infty)$	$[0, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$	0 or 1

Table 1: Value range of each feature in spatial relation

Considering the different value ranges of each feature, we use a double sigmoid function [10] to normalize $RS(str_{ref}, str_{arg})$ and $dist_{str}(str_{ref}, str_{arg})$ into $[0, 1]$. The double sigmoid function reduces the effect of the extreme large and small values (outliers). Thus the 7 features are balanced in terms of their dynamics. Finally, we get a spatial relation vector of 7 dimensions to model the spatial relations between two strokes. The distance between two spatial relation vectors is simply defined as the Euclidean distance. We use the *K-means* clustering algorithm to generate n_{sr} spatial relation prototypes. Therefore the edges (spatial relations) in the relational graph can be labeled with the n_{csr} closest spatial relation prototypes. This labeling step is the quantization of the spatial relations. In this paper, we use $n_{csr} = 1$ which means that it exists only one spatial relation prototype from the reference stroke to the argument stroke. After the quantization of edges, we got the relational graph between strokes.

Figure 4 shows an example where $n_{sr} = 4$ spatial relation prototypes (SR1 to SR4) are used. We quantify the spatial relations on the relational graph. As the targeted application, we will discover the symbols which are the repetitive sub-graphs on the relational graphs in next section.



(a) Spatial relation prototypes (b) Quantified relational graph and repetitive sub-graphs

Figure 4: Quantization of spatial relations and an example of repetitive sub-graphs

2.4 Discovering Graphical Symbols

After the quantization of strokes and spatial relations, we get the relational graph with discrete nodes and edges. Thus, we can discover the symbols on the relational graphs. We introduce briefly the symbol discovery procedure since this paper mainly focuses on the problem of vectorization of spatial relations.

The extraction of some symbol and lexicon knowledge has been proposed from texts using natural languages with unsupervised techniques based on the Minimum Description Length (MDL) principle [11–13.] In this case, the search space is linear since a text is a one dimension sequence of characters. We extend this kind of approach on 2D graphical languages[1]. We organize the graphical elements (strokes) as relational graphs which are in a two-dimensional space. We apply SUBDUE (SUBstructure Discovery Using Examples) to discover symbols in the relation graphs with predefined spatial relations, where SUBDUE [7,14] extracts the repetitive substructures in graphs using MDL principle.

Once the substructures are extracted, they are used to perform a segmentation task on some new graphical sentences. We would like that these substructures correspond as much as possible with the symbols defining the ground-truths of the graphical sentences. Four measures are proposed in [1, 6, 11, 12] to evaluate the recovery performance by a segmentation task: recall rate (R_{Recall}), crossing bracket rate (R_{CB}), lost rate (R_{Lost}), and top rate (R_{Top}). The recall rate calculates the percentage of right segmentations which are found in ground-truths. On the contrary the second measure R_{CB} reveals the errors of the segmentation which are crossing with the ground-truths. R_{Lost} corresponds to the percentage of symbols in ground-truths which are lost, where $R_{Lost} = 1 - R_{Recall} - R_{CB}$. As the result of the hierarchical structures of the resulting segmentation, R_{Top} evaluates the performance of the longest possible segments of the hierarchical segmentation. In the next section, we use this protocol [1,6] and measures to show the contribution of learning spatial relations when compared to that using predefined relations.

3. EXPERIMENTS ON DIFFERENT HANDWRITTEN GRAPHICAL LANGUAGES

We evaluate our approach on three different contexts which correspond to three handwriting databases, an artificial mathematical expression database, a realistic flowchart database, and a Chinese handwriting database [15].

3.1 Handwritten Corpus

The first simple database is a synthetic handwriting database named Calculate [16] of realistic handwritten expressions synthesized from isolated symbols. The expressions in Calculate are produced according to the grammar N_1 *op* $N_2 = N_3$ where N_1, N_2 and N_3 are numbers composed of 1, 2 or 3 digits from real isolated $\{0, 1, ..., 9\}$. The distribution of number of digits for $N_{i=\{1,2,3\}}$ is 70% of 1 digit, 20% of 2 digits and 10% of 3 digits randomly. Furthermore, *op* represents the operators $\{+, -, \times, \div\}$. Figure 5a shows an example in Calculate with N_1, N_2, N_3 and *op* containing 3 digits, 1 digit, 2 digits and “×” respectively. Calculate is composed of a training part and a test part. The training part comprises 897 expressions from 180 writers. The test part contains 497 expressions written by another 100 writers.

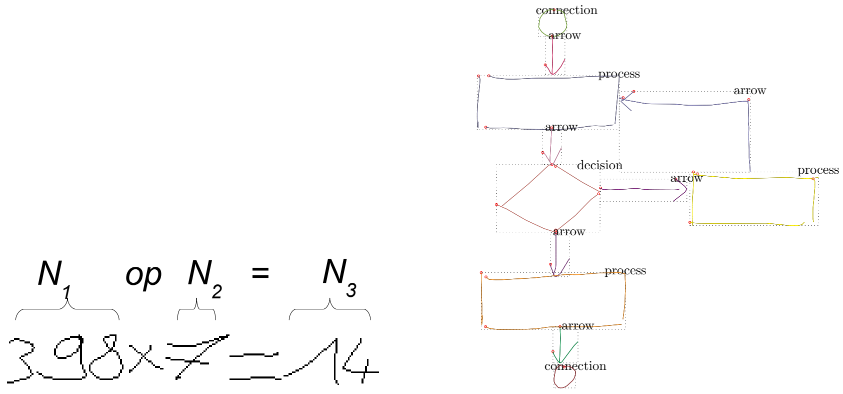
The second handwriting database is a realistic handwritten flowchart database named FC database [17]. We use only the six different graphical symbols that represent the basic operations (data, terminator, process, decision, connection, arrows) without any handwritten text, as displayed in Figure 5b. A total number of 419 flowcharts are written by 36 writers. The training part comprises 248 flowcharts and the test part contains 171 flowcharts.

For these two handwritten databases which are two different graphical languages, we computed spatial relations and graphemes by clustering and extracted the lexicon by the iterative search algorithm [1] on the training part. The learned graphemes, the learned spatial relations, and the learned lexicon are tested on their respective test part.

At the end, we test our approach on the Chinese handwriting database SCUT-COUCH [15] containing handwritten Chinese text-lines with structural Chinese characters. We extract and visualize the frequent patterns in Chinese characters.

3.2 Comparison between predefined spatial relations and learned spatial relations

In this section, we compare the performance of the predefined spatial relation prototypes and unsupervised learned spatial relation prototypes on two different graphical languages, the simple one-line mathematical expressions and the complex flowcharts. Since we mainly focus on the evaluation of learned spatial relations, as an



(a) (b)

Figure 5: Two different handwritten graphical languages: (a) a synthetic expression from the Calculate database composed of real isolated symbols , (b) an example of flowchart in FC database.

example, we set $n_{cstr} = 2$ closest strokes and $n_{csr} = 1$ spatial relation prototype of edge for these two databases. Considering the number of grapheme prototypes, $n_g = 120$ for Calculate database, $n_g = 100$ for FC database.

In our previous work [1], we predefine two directional relations (right, below) and intersection starting with the top-left stroke because of the left to right handwriting orientation. To support a more general 2D graphical language in this experiment setup, we predefine a set of 5 spatial relation prototypes noted (Pre): right, left, below, above and intersection. For instance, the spatial relation prototype of right is a vector $(0, 0, 1, 0, 0, 0, 0)$ in Table 1. Therefore, we choose the number of prototypes as $n_{sr} = 5$ for unsupervised learning so that we can compare the performance between predefined relations and learned relations. To control the clustering and obtain the 5 relational prototypes, we select different feature sets, namely (F4), (I), (F4,I), and (S, D, F4, I), as presented in Table 1. Note the learned relations using the (F4,I) set of features use the same information as the predefined relations (Pre).

We evaluate firstly the learned spatial relations on the test part of the simple Calculate database in Table 2a. (Pre) reports a recall rate of 73.6% on the test part of Calculate database while (F4,I) achieves a slightly lower recall rate of 72.7% (0.9% difference). But the unsupervised relations using the features (S,D,F4,I) report the best recall rate 75% using the same number of prototypes $n_{sr} = 5$.

We evaluate secondly in Table 2b the learned spatial relations on the test part of the more complex FC database. Although (Pre) gets a recall rate 49% which is also slightly higher than that of (F4,I), the unsupervised (F4) achieves the highest recall rate 52.8%. It means that (F4) is more symbol discriminant on FC database.

	R_{Recall}	R_{CB}	R_{Lost}	R_{Top}
(Pre)	73.6%	20.8%	5.6%	12.3%
(F4)	74.5%	19.6%	5.9%	13.4%
(I)	70.7%	25.2%	4.2%	10.7%
(F4,I)	72.9%	21.4%	5.6%	12.2%
(S,D,F4,I)	75%	19.8%	5.2%	11.8%

(a) On the test part of Calculate database.

	R_{Recall}	R_{CB}	R_{Lost}	R_{Top}
(Pre)	49%	21.9%	29.1%	38.4%
(F4)	52.8%	14.9%	32.3%	37.3%
(I)	47.7%	28.9%	23.4%	24.8%
(F4,I)	48.7%	24.7%	26.5%	28.7%
(S,D,F4,I)	46.1%	26.8%	27.1%	35.9%

(b) On the test part of FC database

Table 2: Comparison between “Pre”(5 predefined spatial relation prototypes: right, left, above, below and intersection) and $n_{sr} = 5$ learned spatial relation prototypes using different features respectively: (F4), (I), (F4, I), (S, D, F4, I) on the test part of databases

Then, we evaluate different numbers of spatial relation prototypes on the test part of FC database in Figure 6 using (F4). The recall rate gets the peek 52.8% using the number of prototypes $n_{sr} = 5$. Thus the best tradeoff number of prototypes is 5 because of the high R_{Recall} and the low R_{CB}.

Figure 7 illustrates a real example of symbol recovery from a flowchart. In three sub-figures, the dashed rectangle boxes represent the segmentations. The different colors represent the different longest segmentations in the hierarchical structure. The small red circle means where is the pen-down (starting) point in a stroke. The

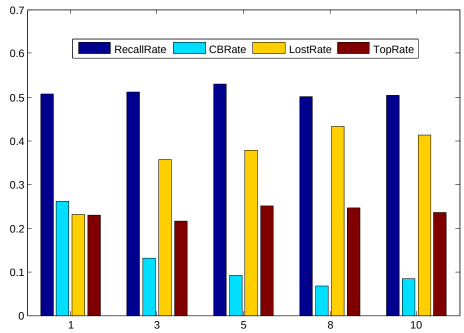
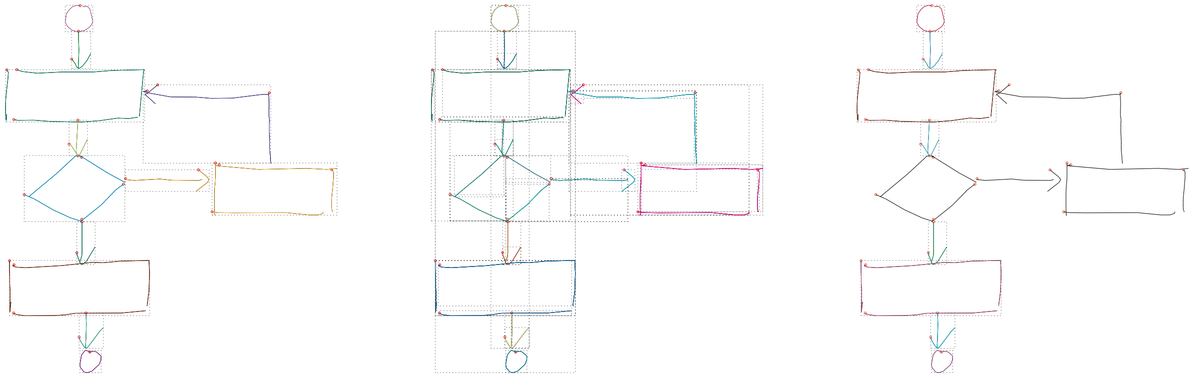


Figure 6: Performance of different numbers of spatial relation prototypes using the feature (F4) on the test part of FC database

ground-truths of flowchart are shown in Figure 7a. Figure 7b displays the learned hierarchical segmentations for this flowchart. We show the common segmentations which are R_{Recall} between the ground-truths and learned segmentations in Figure 7c. Thus we can find that the frequent arrows, rectangles, and circles are extracted.



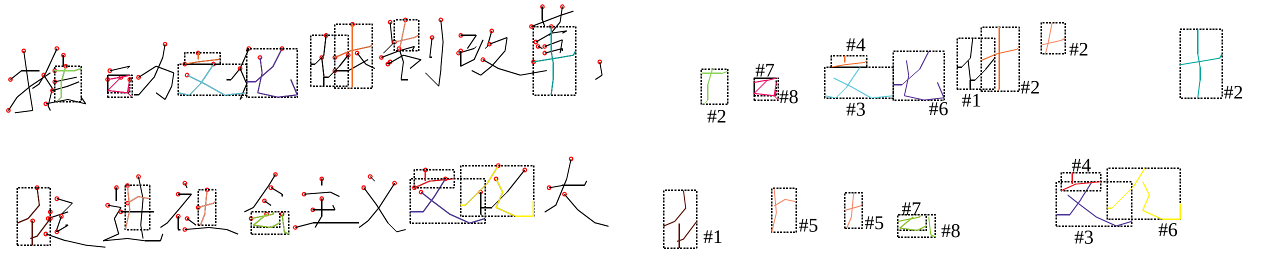
(a) (b) (c)

Figure 7: Example of symbols recovery: a) the ground-truths of flowchart, b) the hierarchical segmentations using unsupervised spatial relations, and c) the intersection between the ground-truths and the hierarchical segmentations which are the definition of R_{Recall}.

At the end, we test qualitatively our strategy on only two text-lines in the Chinese database SCUT-COUCH [15] in Figure 8. The same symbol extraction as previous examples is applied on learning data and the Figure 8a shows the resulting hierarchical segmented Chinese texts. The frequent stroke layouts in Chinese texts are illustrated in Figure 8b where $\#i$ means we extract the patterns in the i iteration, starting from the most frequent and large pattern (more strokes) to reduce the description length [1]. We can find 8 different stroke layouts using the MDL principle. Some of them are Chinese radicals. For instance, the extracted layouts in #3 and #4 iterations are radicals in the character “ 文 ”.

4. CONCLUSION

In this paper, we discuss the construction of a relational graph between strokes (nodes) and the quantization of the spatial relations (edges). As the application, we extract the handwritten graphical symbols (the frequent sub-graphs) using an algorithm proposed in [1]. Through the experiments, we compare the performance between the predefined spatial relations and learned spatial relations on two handwriting databases. On two handwritten different contexts, mathematical expressions and flowcharts, the learned spatial relation prototypes outperform the predefined spatial relations. We got a fair recall rate of 52.8% against 49% using predefined spatial relations on the test part of FC database. In addition, we visualize the frequent layout of strokes on two Chinese text lines to understand more explicitly our strategy.



(a) (b)

Figure 8: Unsupervised learning of Chinese characters: a) the hierarchical segmentation for two text-lines and b) the extracted frequent patterns in $\#i$ iteration.

However, the spatial relation between two strokes is quantified into only $n_{csr} = 1$ edge (one spatial relation prototype). This quantization lose some information. The fuzzy edge matching of spatial relations could improve the recall rate which would be our future work.

Acknowledgment: This work is supported by the French Région Pays de la Loire in the context of the DEPART project www.projet-depart.org.

REFERENCES

[1] Li, J., Mouchère, H., and Viard-Gaudin, C., "Unsupervised handwritten graphical symbol learning-using minimum description length principle on relational graph," in [*KDIR2011*], (2011).

[2] Clementini, E., *A Conceptual Framework for Modelling Spatial Relations*, PhD thesis, INSA, LYON (2009).

[3] Bouteruche, F., Mac, S., and Anquetil, E., "Fuzzy relative positioning for on-line handwritten stroke analysis," in [*IWFHR'06*], (October 2006).

[4] Delaye, A., Mac, S., and Anquetil, E., "Modeling relative positioning of handwritten patterns," in [*14th Biennial Conference of the International Graphonomics Society (IGS 2009)*], 152–156 (2009).

[5] Rhee, T. H. and Kim, J. H., "Efficient search strategy in structural analysis for handwritten mathematical expression recognition," *Pattern Recognition* **42**(12), 3192 – 3201 (2009).

[6] Li, J., Mouchère, H., and Viard-Gaudin, C., "Symbol knowledge extraction from a simple graphical language," in [*ICDAR2011*], (2011).

[7] Cook, D. J. and Holder, L. B., "Substructure discovery using examples," (2011).

[8] Baird, J. C., [*Psychophysical analysis of visual space*], Oxford, London: Pergamon Press (1970).

[9] Egenhofer, M. and Herring, J., [*Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases*], Department of Surveying Engineering, University of Maine, Orono, ME (1991).

[10] Jain, A., Nandakumar, K., and Ross, A., "Score normalization in multimodal biometric systems," *Pattern Recognition* **38**, 2270–2285 (Dec. 2005).

[11] Marcken, C. D., *Unsupervised Language Acquisition*, PhD thesis, Massachusetts Institute of Technology (1996).

[12] Marcken, C. D., "Linguistic structure as composition and perturbation," in [*In Meeting of the Association for Computational Linguistics*], 335–341, Morgan Kaufmann Publishers (1996).

[13] Rissanen, J., "Modeling by shortest data description," *Automatica* **14**(5), 465 – 471 (1978).

[14] Cook, D. J. and Holder, L. B., "Substructure discovery using minimum description length and background knowledge," *J. Artif. Int. Res.* **1**, 231–255 (February 1994).

[15] Jin, L., Gao, Y., Liu, G., Li, Y., and Ding, K., "A comprehensive online unconstrained chinese handwriting database and benchmark evaluation," *International Journal of Document Analysis and Recognition* (2010).

[16] Awal, A. M., *Reconnaissance de structures bidimensionnelles : application aux expressions mathématiques manuscrites en-ligne*, PhD thesis, Ecole polytechnique de l'université de Nantes, France (2010).

[17] Awal, A.-M., Feng, G., Mouchere, H., and Viard-Gaudin, C., "First experiments on a new online handwritten flowchart database," in [*Document Recognition and Retrieval XVIII*], **7874**, 78740A–78740A–10 (2011).