A Novel Learning-free Word Spotting Approach Based On Graph Representation

Peng Wang[1,2], Véronique Eglin[1], Christophe Garcia[1]
[1]INSA-Lyon
LIRIS, CNRS UMR5205
F-69621, France
Email: peng.wang@liris.cnrs.fr

Christine Largeron[2]
[2]Université Jean Monnet
LAHC, CNRS UMR 5516
F-42023, France
Email: christine.largeron@univ-st-etienne.fr

Josep Lladós[3], Alicia Fornés[3]
[3]Universitat Autònoma de Barcelona
Computer Vision Center
08196, Spain
Email:{josep, afornes}@cvc.uab.es

***Abstract*—Effective information retrieval on handwritten document images has always been a challenging task. In this paper, we propose a novel handwritten word spotting approach based on graph representation. The presented model comprises both topological and morphological signatures of handwriting. Skeleton-based graphs with the Shape Context labelled vertexes are established for connected components. Each word image is represented as a sequence of graphs. In order to be robust to the handwriting variations, an exhaustive merging process based on DTW alignment result is introduced in the similarity measure between word images. With respect to the computation complexity, an approximate graph edit distance approach using bipartite matching is employed for graph matching. The experiments on the George Washington dataset and the marriage records from the Barcelona Cathedral dataset demonstrate that the proposed approach outperforms the state-of-the-art structural methods.**

I. Introduction

As more and more documents, especially handwritten documents, are converted into digitized version for the long-term preservation, the demands for efficient information retrieval techniques in such document images are increasing. Word spotting in handwritten document images is one of such expected techniques. As a specific pattern recognition task, handwritten word spotting faces many challenges such as the high intra-writer and inter-writer variability. Nowadays, it has been admitted that OCR techniques are unsuccessful in handwritten offline documents, especially historical ones. Therefore, the particular word spotting approach for handwriting is strongly required. Generally, there are two groups of methods addressing the handwritten word spotting problem. One is dedicated to the detection of predefined words (associated to a training process, like the works presented in [1],[2]), and the other one is retrieval-oriented with dedicated matching scheme based on image sample comparison without any necessary associated training process [3],[4]. In both cases and from all works dealing with the access to handwritten content, it has been proved that an accurate and comprehensive representation of the image content is necessary [1],[2],[3]. Moreover, the similarity measure is supposed to be enough robust to tolerate the inner variations in the handwriting.

Graph-based representation is a popular approach to capture and model the structural properties of objects. It has been widely used in the pattern recognition domain, for instance, in molecular compound recognition in chemistry and symbol recognition in document analysis. However, the employment of graph representation in the handwritten word spotting application is still very rare. There are only a few attempts made in handwritten recognition researches with graph representation or relative concepts up to now. Researchers started using graphs in the context of single character recognition, such as the graph-based recognition of Chinese characters reported in [5],[6]. Even though the hierarchical model proposed in [5] reflects the hierarchical nature of handwritten characters, especially of Chinese Characters, which has a more complex structure than Latin languages, in general, a more sophisticated representation mechanism is required to distinguish similar characters. Later, Fischer et al. developed a graph representation model based on the skeleton of word image, which contains only vertexes without edges [7], [8]. By adding enough intersection points among keypoints, the structural information is still preserved and meanwhile graph representation is robust to the bad quality of skeleton. However, it also leads to the fact that graphs may contain too many vertexes. Recently, due to the expensive computational cost of graph matching, the transformation of graphs into feature vectors calls the concern. In [9], a directed acyclic graph is constructed for each subword first, and then is converted into a topological signature vector that preserves the graph structure. Moreover, Lladós et al. [10] adapted the graph serialization concept in graph matching for handwritten word spotting. By extracting and clustering the acyclic paths of graphs, one-dimensional descriptor, bag-of-paths (BoP), is generated for describing the words. This attempt is interesting but the performance is far away from satisfying.

In our work, the motivation for adapting graph-based representation to handwritten word spotting rather than using the appearance-based representation model is mainly due to the fact that the structure is more stable and relevant to the handwriting than the pixels. Moreover, topological characteristics are very important for describing the handwriting and the two-dimensional nature of handwriting is difficult to be well described by a single one-dimensional sequential scalar feature vector [1],[9].

In this paper, we propose a novel learning-free word spotting approach based on the graph representation. The proposed approach is mainly designed for addressing the Query-by-example (QBE) word spotting problem. In order to preserve the two-dimensional structural information of handwriting, we es-

(a) (b) (c) (d)

Fig. 1: (a) a handwritten word (b) the contour (c) the skeleton (d) the structural points (red—starting/ending points, green—high-curved points, blue—branch points)

tablish a graph-based representation to model the handwriting images. Morphological properties of handwriting described by the Shape Context descriptor [11] are also integrated into the representation model to exploit the contextual information in the neighborhood of the vertexes,which makes the description more comprehensive. In the end, a word image is represented as a sequence of graphs. Each graph corresponds to one connected component. The comparison between word images is realized by taking average graph edit distance among the corresponding graph pairs. Before this step, DTW (Dynamic Time Warping) with a merging process is applied to find out the correspondences among graphs. An approximate graph edit distance approach [12] is chosen, because it gives a substantial speed-up over the classical graph edit distance method. The proposed word spotting approach belongs to the second group of methods where we do not have any predefined word model and the similarity measure is robust to the inner variations of the handwriting.

The rest of this paper is organized as follows. First, we give a brief introduction on the preprocessing of the skeleton, contour and structural point detection. Afterwards, the graph-based representation model for word image is presented in Section III. Next, the similarity measure between word images based on DTW and exhaustive merging with an optimal graph edit distance is described in Section IV. The experimental results are provided and discussed in Section V. Finally, the conclusion and perspectives are given in Section VI.

II. PREPROCESSING

Because this paper is focused on word image representation and comparison, we assume that all texts have already been segmented into isolated words.

In order to obtain the topological information of the handwriting, we apply the Zhang & Suen's [13] parallel thinning algorithm on the binarized images to extract the skeleton of the text, as shown in Fig.1(c). Afterwards, three types of structural points related to the nature of the language and the handwriting style are detected on the skeleton image by using the method introduced in [14]. These structural points are respectively starting/ending points, high-curved points and branch points. An example of three types of structural points is given in Fig.1(d).

According to the location of structural points, the handwriting is segmented into strokes.

For the morphological information, we choose to use the outer contour of the handwriting. After filling the holes in the handwriting, Canny edge detector is used to extract edges.

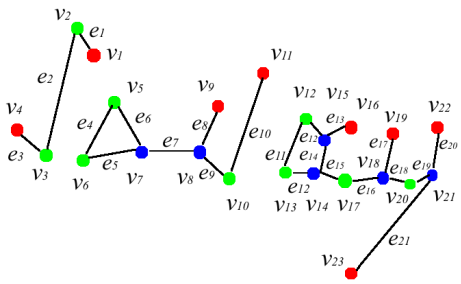


Fig. 2: Graph representation of a word

Then, a contour tracing approach based on a border following algorithm is applied to the detected edges (see Fig.1(b)).

III. GRAPH-BASED REPRESENTATION

Graph representation is a very powerful method of representing images in terms of preserving structural properties. Our graph-based representation model is established on the skeleton of the handwriting. The vertexes of the graph correspond to the structural points and the strokes connecting them are considered as the edges (see Fig.2). The graph representation at this step is too simple to exhaustive describe handwritten shapes, because it only contains the structural information. In order to integrate morphological properties in the graph representation model, we use the Shape Context descriptor [11] as the attribute of the vertexes. Shape Context descriptor is a point descriptor which captures the distribution over relative positions of other shape points and thus summarizes global shape (all contextual information in the neighborhood of the point) in a rich, local descriptor. Considering the detected structural points as reference points, the Shape Context descriptors are calculated based on the extracted contour. For the edges, we take the length of stroke (the amount of pixels on the skeleton of the stroke) as the attribute. The definition of the graph in our case is as follows.

Definition 1. (Graph) A graph is a four-tuple $g = (V, E, \mu, v)$, where

- V is the finite set of vertexes, corresponding to the structural points
- $E \subseteq V \times V$ is the set of edges, corresponding to the strokes
- $\mu : V \rightarrow L$ is the vertex labelling function, corresponding to the Shape Context descriptor
- $v : E \rightarrow L$ is the edge labelling function, corresponding to the length of stroke

In this way, each connected component can generate an individual graph. Usually, a handwritten word is composed of several connected components. Therefore, a word image in our approach is represented as a sequence of connected graphs. For example, the word "*Savay*" in Fig.2 is represented by three graphs respectively corresponding to "*S*", "*av*" and "*ay*".

IV. SIMILARITY MEASURE

The similarity measure between two word images is computed in two steps. In the first step, the comparison between

each pair of connected components of the query and the test image is conducted. Next, the distance between two word images is calculated based on the results in the first step.

A. Connected Component Comparison

Since connected components are represented as graphs, the comparison among words can be solved based on graph matching. To avoid the computational burden, we choose to use an approximate graph edit distance algorithm proposed by K. Riesen and H. Bunke [12]. The edit distance between two attributed graphs is defined by the minimum cost edit path between two graphs.

Definition 2. (Graph edit distance) Let $g_1 = (V_1, E_1, \mu_1, v_1)$ be the source graph and $g_2 = (V_2, E_2, \mu_2, v_2)$ be the target graph. The graph edit distance between g_1 and g_2 is defined by

$$ged(g_1, g_2) = min_{(e_1,\cdots,e_k)\in\gamma(g_1,g_2)} \sum_{i=1}^{k} c(e_i) \qquad (1)$$

where $\gamma(g_1, g_2)$ denotes the set of edit paths transforming g_1 into g_2, and $c(e_i)$ denotes the cost function measuring the strength of edit operation e_i (insertion, deletion and substitution of vertexes and edges).

This suboptimal algorithm is based on the decomposition of graphs into sets of subgraphs. These subgraphs consist of a vertex and its adjacent structures. In this case, the graph edit distance is approximately reformulated as finding the optimal match between vertexes and their local structures of two graphs. Instead of using dynamic programming, bipartite matching is adapted to find the optimal match. The subgraph matching is treated as the assignment problem. the Munkres (Hungarian) assignment algorithm [15] is used to find an optimal assignment of all subgraphs, which minimizes the sum of the assignment cost.

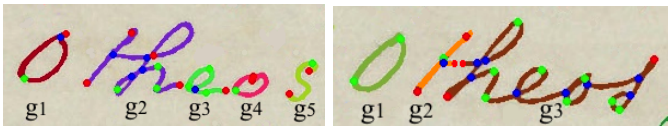
To involve the consideration of local structure of graph, the cost of vertex substitution is composed of two parts. The first one is the cost calculated from the attributes of two vertexes. The second part comes from the local structure comparison, as shown in Eq.2. We have tested the performance of graph edit distance with several different sets of weights ($w_1 = 0.2$ and $w_2 = 0.8$, $w_1 = 0.5$ and $w_2 = 0.5$, $w_1 = 0.8$ and $w_2 = 0.2$). The results reveals that in our scenario, the description given by the Shape Context is more discriminant and important than the information depicted by the adjacent structure in terms of the subgraph. Hence, we assign 0.8 to w_1 and 0.2 to w_2.

$$C_{substitution} = w_1 C_{SC} + w_2 C_{local\_structure} \qquad (2)$$

$$C_{insertion} = a \qquad (3)$$

$$C_{deletion} = d \qquad (4)$$

where *a* and *d* are constant values experimentally set. Since the range of substitution cost is from 0 to 1, we use 0.5 as the insertion and deletion cost. C_{SC} is the cost of the Shape Context. $C_{local\_structure}$ is the edit operation cost on the adjacent edges. The costs of the deletion and insertion of edge are constant values, while the substitution cost equals to $1 - e_{short}/e_{long}$, where e_{short} denotes the length of the shorter edge and e_{long} denotes the length of the longer edge.



(a) (b)

Fig. 3: Two instances of the word "*Otheos*" with different amounts of connected components

Typically, the graph edit distance is calculated with a tree search of an exponential time complexity. However, the chosen approximate graph matching approach can solve the bipartite matching problem in polynomial time.

B. Word Image Matching

Since a word image is represented as a sequence of connected components, DTW is adapted to find the assignments among connected components using the graph edit distance introduced in the previous part. As a major variation of the handwriting, the amount of connected components of the same word can vary a lot in different instances, even in the same writing style. It potentially results in *n* to *m* mapping from one sequence of graphs to the other (*n* and *m* are respectively the numbers of graphs in two words). Fig.3 gives an example that two instances of one word have respectively 5 and 3 connected components.

In order to make the similarity measure robust to such variation, based on the matching results obtained in the previous step, an exhaustive merging operation is performed. The graphs assigned to the same connected component are considered as one entire graph. In this way, both the sequences of graphs representing the query and the test image might change. Afterwards, with the new sequences of graph representation, the graph edit distance is calculated again between new corresponding pair of graphs. The average distance among the new corresponding graph matchings is considered as the final distance between two word images. Take two instances of the word "*Otheos*" in Fig.3 for example. Given $a = g_a^1, \cdots, g_a^5$ represents the word in Fig.3 (a) and $b = g_b^1, g_b^2, g_b^3$ represents the word in Fig.3 (b). After the DTW alignment, it shows that g_a^1 corresponds to g_b^1, g_a^2 is assigned to both g_b^2 and g_b^3 , g_a^3, g_a^4, g_a^5 correspond to g_b^3. Therefore, in the next step, $g_a^2, g_a^3, g_a^4, g_a^5$ are merged together as an entire graph $(g_a^2)'$, because all of them are matched to g_b^3. g_b^2 and g_b^3 are merged also as $(g_b^2)'$, because both of them are matched to g_a^2. In this way, both image(a) and image(b) are reformulated as the sequences of two connected graphs. The graph edit distance is calculated again between g_a^1 and g_b^1, $(g_a^2)'$ and$(g_b^2)'$, which are respectively noted as $ged(g_a^1, g_b^1)$ and $ged((g_a^2)', (g_b^2)')$. The final distance between two words $d(a, b)$ is the average of $ged(g_a^1, g_b^1)$ and $ged((g_a^2)', (g_b^2)')$.

V. EXPERIMENT RESULTS

In this part, the proposed approach is applied to two datasets. The results are compared to the performance of other four word spotting approaches. They are respectively sequence alignment using DTW with Manmatha's features [3], a bag

TABLE I: Retrieval result for the George Washington dataset

	P@10	P@20	R-Precision	mAP
Proposed	0.372	0.312	0.206	0.175
DTW	0.346	0.286	0.191	0.169
BoVW	0.606	0.523	0.412	0.422
Pseudo-Struct	0.183	0.149	0.096	0.072
Structural	0.059	0.049	0.036	0.028

of visual word approach as statistical model [10], a pseudo-structural model based on a Loci feature representation [10] and a graph-based approach using the bag-of-paths descriptor [10]. The results are shown in the latter part.

A. Experimental Data and Evaluation Criteria

For the experimental evaluation, the proposed approach is performed on two datasets. The first one consists of 20 pages from a collection of letters by George Washington [3]. The second evaluation corpus contains 27 pages from a collection of marriage registers from the Barcelona Cathedral[16]. Both corpus are accurately segmented into words and transcribed. In the George Washington dataset, there are 4860 segmented words with 1124 different transcriptions, whereas in the Barcelona Cathedral dataset we have 6544 words with 1751 transcriptions. All the words containing at least 3 letters and appearing at least 10 times in the collection are selected as queries. In this way, there are 1847 queries corresponding to 68 different words for the George Washington dataset and 514 queries from 32 different word classes for the Barcelona Cathedral dataset.

In order to evaluate the performance, we choose three performance assessments based on precision and recall measures. Given a query, let us label *rel* the set of relevant objects with regard to the query and as *ret* the set of retrieved elements from the dataset.

- *P@n* — the precision at *n* is obtained by computing the precision at a given cut-off rank, considering only the *n* topmost results returned by the system. In the evaluation, we provide the results at the 10 and 20 ranks.
- *R-Precision* — the *R-Precision* is the precision computed at the *R*th position in the ranking of results, *R* being the number of relevant words for that specific query.
- *mAP* (mean Average Precision) — *mAP* is computed using each precision value after truncating at each relevant item in the ranked list. For a given query, let *r(n)* be a binary function on the relevance of the *n*th item in the returned ranked list, the mean average precision is defined as follows:

$$mAP = \frac{\sum_{n=1}^{|ret|} P@n \times r(n)}{|rel|} \qquad (5)$$

B. Results and Discussion

Fig.4 presents the plots of precision-recall curves for both the George Washington dataset and the Barcelona Cathedral dataset. In table I and II the evaluation of the performance for both datasets are given. We can see that in both scenarios, the proposed approach outperforms the DTW-based, the pseudo-structural and the structural methods. The BoVW method

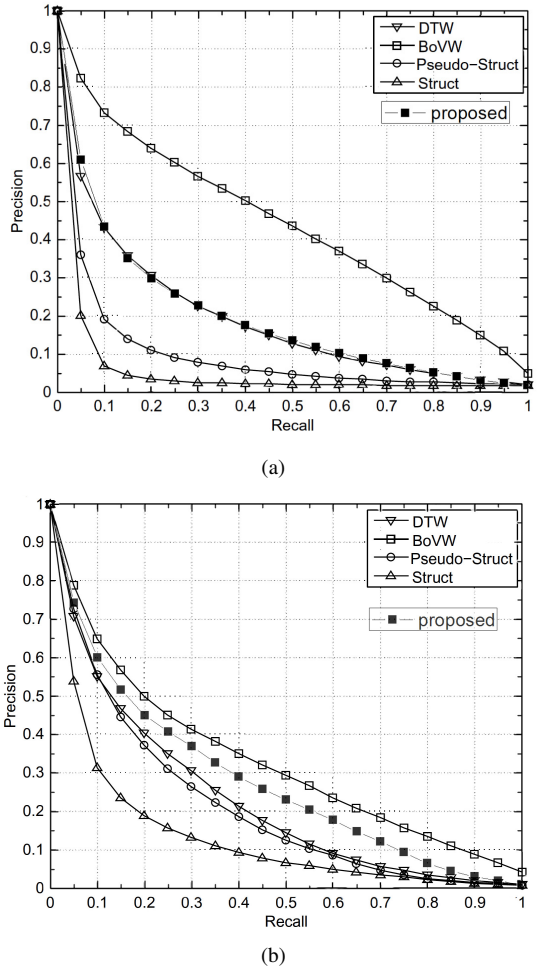


Fig. 4: Precision and recall curves for the (a) George Washington dataset and the (b) Barcelona Cathedral dataset

TABLE II: Retrieval result for the Barcelona Cathedral dataset

	P@10	P@20	R-Precision	mAP
Proposed	0.342	0.241	0.270	0.246
DTW	0.288	0.201	0.214	0.192
BoVW	0.378	0.289	0.303	0.3
Pseudo-Struct	0.273	0.189	0.199	0.178
Structural	0.155	0.12	0.118	0.097

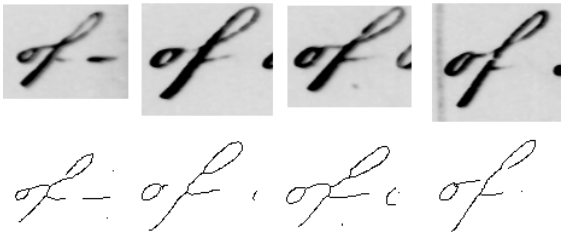


Fig. 5: Skeleton examples for different instances of the same word class for the George Washington dataset

achieves the best performance on both datasets. It is mainly because the BoVW method extracts word description directly from the gray-level image without any binarization process. All the other methods in the comparison establish the representation based on the binarized images. The problem caused by the ink degradation is clearly magnified by the binarizing process. In our approach, the discontinuity of the skeleton caused by the binarization directly leads to the deformation of the graph representation and increases the unstability of connected components (see Fig.5). Even though the specific similarity measure proposed in this paper can tolerate this distortion, the final distance between images can still be slightly impacted. Moreover, compared to other methods, the BoVW method produces a more dense description by extracting visual words based on SIFT detection and represents the handwriting in a more statistical way. However, the classification of visual words and the usage of the Spatial Pyramid Matching in the BoVW method greatly increase the complexity and the computational cost of the approach. It is interesting to point out that the performance of the proposed approach is better on the Barcelona Cathedral dataset than the George Washington dataset. The reason is that most word images in the George Washington dataset have more degradation than the images in the Barcelona Cathedral dataset. As a conclusion, the main constrain of the proposed approach is the quality of the skeleton. It can be solved by adapting more effective approach or adding some intersection points to increase the stability of the skeleton.

Except the BoVW method, the proposed approach is proved to be more competent than other three methods in the comparison. Especially, the fact that the proposed approach greatly outperform the pseudo-structural and the structural methods demonstrates that using structural points to construct the graph is very informative, and furthermore, the integration of the contextual information depicted by the Shape Context in the graph representation is necessary and promising.

VI. CONCLUSION AND FUTURE WORK

In this paper, we mainly made two contributions. First of all, we have adapted the graph representation to model the signatures of handwriting in terms of both structure and form. Handwritten words are represented in the form of sequences of graphs. Secondly, we have introduced an innovative word matching method based on the approximate graph edit distance. The proposed graph-based representation model realizes an early-fusion of the morphological and topological description of the handwriting. Skeleton-based graphs capture the structural signatures of the handwriting, meanwhile the adaptation of the Shape Context complements the lack of the description in the morphological aspect. Moreover, the merging process employed in the matching process overcomes the problem caused by the variation of connected components, which makes the similarity measure more robust. To the knowledge of the authors, the similarity measure proposed in this paper is the first attempt to apply the graph matching in the scope of word image without any decomposition and learning. One limit of the proposed approach is the computational cost. Even though the employed approximate graph edit distance approach has sharply reduced the computational expense, two-round graph matching process within a word comparison as well as 60-dimensional Shape Context attribute vectors increases the computational burden. It can be considered as a trade-off of keeping comparing graphs in 2D instead of comparing graphs after converting them into 1D vectors.

In future works, in order to reduce the computational expense, adding a coarse selection to rapidly reject negative samples before applying the graph matching seems interesting. Besides, using the first and last graphs of the query to index the regions of interest on the page can also be investigated.

ACKNOWLEDGMENT

We want to thank Professor Antony McKenna who offered us images. The research is supported by the French Rhone-Alpes region program, CITERE ANR project and partially supported by the Spanish project TIN2012-37475-C02-02, and by the EU project ERC-2010-AdG- 20100407-269796.

REFERENCES

[1] A. Fischer, A. keller, V. Frinken, and H. Bunke, "Hmm-based word spotting in handwritten documents using subword models," in *Proc. of the ICPR*, 2010, pp. 3416–3419.

[2] J. Rodríguez-Serrano and F. Perronnin, "Handwritten wordspotting using hidden markov models and universal vocabularies," *PR*, vol. 42, no. 9, pp. 2106–2116, 2009.

[3] T. Rath and R. Manmatha, "Word spotting for historical documents," in *Proc. of the ICDAR*, vol. 9, no. 2-4, 2007, pp. 139–152.

[4] Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz, "Towards an omnilingual word retrieval system for ancient manuscripts," *PR*, vol. 42, no. 9, pp. 2089–2105, 2009.

[5] S. Lu, Y. Ren, and C. Suen, "Hierarchical attributed graph representation and recognition of handwritten chinese characters," *PR*, vol. 24, no. 7, pp. 617–632, 1991.

[6] M. Zaslavskiy, F. Bach, and J. Vert, "A path following algorithm for the graph matching problem," *PAMI*, vol. 31, no. 12, pp. 2227–2241, 2009.

[7] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke, "A fast matching algorithm for graph-based handwriting recognition," in *Lecture Note in Computer Science*, vol. 7887, 2013, pp. 194–203.

[8] A. Fischer, K. Riesen, and H. Bunke, "Graph similarity features for hmm-based handwriting recognition in historical documents," in *Proc. of the ICFHR*, 2010, pp. 253–258.

[9] Y. Chherawala, R. Wisnovsky, and M. Cheriet, "Tsv-lr: Topological signature vector-based lexicon reduction for fast recognition of pre-modern arabic subwords," in *Proc. of the Workshop HIP*, 2011, pp. 6–13.

[10] J. Lladós, M. Rusinol, A. Fornés, D. Fernandez, and A. Dutta, "On the influence of word representations for handwritten word spotting in historical documents," *Pattern Recognition and Artificial Intelligence*, vol. 26, no. 5, pp. 1 263 002.1–1 263 002.25, 2012.

[11] J. Puzicha, S. Belongie, and J. Malik, "Shape matching and object recognition using shape contexts," *PAMI*, vol. 24, pp. 509–522, 2002.

[12] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image and Vision Computing*, vol. 27, no. 7, pp. 950–959, 2009.

[13] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," in *Communication of the ACM*, vol. 27, 1984, pp. 236–239.

[14] P. Wang, V. Eglin, C. Largeron, A. McKenna, and C. Garcia, "A comprehensive representation model for handwriting dedicated to word spotting," in *Proc. of the ICDAR*, 2013.

[15] J. Munkres, "Algorithms for the assignment and transportation problems," *The Society for Industrial and Applied Mathematics*, vol. 5, pp. 32–38, 1957.

[16] D. Fernandez, J.Lladós, and A. Fornés, "Handwritten word spotting in old manuscript images using a pseudo-structural descriptor organized in a hash structure," *Pattern Recognition and Image Analysis*, vol. 6669, pp. 628–635, 2011.