The Role of the Users in Handwritten Word Spotting Applications: Query Fusion and Relevance Feedback

Marçal Rusiñol and Josep Lladós
Computer Vision Center, Dept. Ciències de la Computació
Edifici O, Universitat Autònoma de Barcelona
08193 Bellaterra (Barcelona), Spain
{*marcal,josep*}*@cvc.uab.cat*

***Abstract*—In this paper we present the importance of including the user in the loop in a handwritten word spotting framework. Several off-the-shelf query fusion and relevance feedback strategies have been tested in the handwritten word spotting context. The increase in terms of precision when the user is included in the loop is assessed using two datasets of historical handwritten documents and a baseline word spotting approach based on a bag-of-visual-words model.**

***Keywords*-Handwritten word spotting, Relevance feedback, Query fusion**

I. Introduction

Within the field of document image analysis, handwritten word spotting has received a lot of attention and is today a quite mature research topic –the firsts word spotting approaches applied to handwritten document images were presented in the mid 90's [1], [2].–

Handwritten word spotting methods can be broadly categorized into two main families. The first group consists of the word spotting methods that are aimed at detecting just a set of predefined words. These methods usually entail a training step in which a model for each of the possible words that the user wants to spot is built. Usually, these methods are preferred in multi-writer scenarios, where the user wants to assess whether a document contains one of the predefined keywords or not. Some examples of this family are the works proposed by Rodríguez-Serrano and Perronnin in [3] or by Frinken et al. in [4]. One the other hand, there is another set of word spotting methods which are more retrieval-oriented. In that case, given a document collection which has been indexed off-line, the user casts a word query and he wants to retrieve from the image collection similar instances of that word. In that case there is no training stage involved and the user can query whatever word he wants. Some examples of this family are the works proposed by Fornés et al. in [5] or Terasawa and Tanaka in [6]. We target our work in that second group of handwritten word spotting methods.

Although these word spotting methods can be seen as a particular application of the information retrieval (IR) field, very few works have taken advantage of common strategies from the IR field. A clear example is the lack of word spotting methods that include the user in the loop. Just some works like the method by Bhardwaj et al. [7] or the one by Cao et al. [8] propose to include a relevance feedback step. They both use the Rocchio's [9] well-known relevance feedback method and they both show significant improvements when including this feedback from the user. Similar conclusions were drawn in the case of typewritten word spotting in the work presented by Konidaris et al. [10] and Kesidis et al. [11].

We present in this paper a study on the effect of taking the user into account in a handwritten word spotting framework. We test in this paper two different approaches, namely, query fusion and relevance feedback. The former consists of asking to the user to cast several queries instead of a single one and somehow combine the results. The latter consists of retrieving the similar words from the dataset and asking to the user to provide some feedback about which results were correct and which were incorrect. This relevance feedback allows to provide an enhanced result list in a subsequent iteration. Several off-the-shelf IR methods are applied in the word spotting context. The increase in terms of precision is assessed using two datasets of historical handwritten documents and a baseline word spotting approach based on a bag-of-visual-words model.

The remainder of this paper is organized as follows. We overview in Section II the baseline handwritten word spotting method and in Section III we present the document image datasets and the evaluation measures. Section IV is focused on the query fusion experiments whereas Section V deals with relevance feedback. We provide in Section VI the experimental results. We conclude and present some discussion on Section VII.

II. Baseline Method

In this section, we give the details of our word spotting baseline method. Here, we assume that the words in the document pages have been previously segmented by a layout analysis step. Both the queries and the items in the database are thus segmented word snippets. The way we describe those word images is based on the bag-of-visual-words (BoVW) model powered by SIFT [12] descriptors. We present below an overview of the steps of this baseline

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.282

method (the interested reader is referred to our original publication in [13]). We start with a clustering of SIFT descriptors to build the codebook. Once we have the codebook, word images are encoded by the BoVW model. In a last step, in order to produce more robust word descriptors, we add some coarse spatial information to the orderless BoVW model.

A. Codebook generation

For each word image in the reference set, we densely calculate the SIFT descriptors over a regular grid by using the method presented by Fulkerson et al. in [14]. Three different SIFT descriptor scales are considered. The grid and scale parameters are dependent on the word sizes, and in our case have been experimentally set. We can see in Figure 1 an example of dense SIFT features extracted from a word image. Because the descriptors are densely sampled, some SIFT descriptors calculated in low textured regions are unreliable. Therefore, descriptors having a low gradient magnitude before normalization are directly discarded.

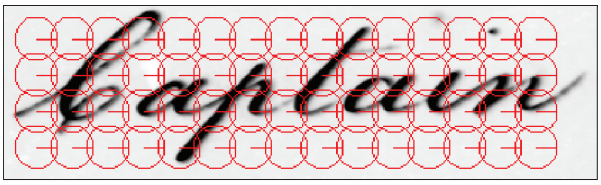


Figure 1. Dense SIFT features extracted from a word image.

Once the SIFT descriptors are calculated, by clustering the descriptor feature space into k clusters we obtain the codebook that quantizes SIFT feature vectors into visual words. We use the k-means algorithm to perform the clustering of the feature vectors. In this work, we use a codebook with dimensionality of $k = 20.000$ visual words.

B. BoVW feature vectors

For each of the word images, we extract the SIFT descriptors, and we quantize them into visual words with the codebook. Then, the visual word associated to a descriptor corresponds to the index of the cluster that each descriptor belongs to. The BoVW feature vector for a given word snippet is then computed by counting the occurrences of each of the visual words in the image.

C. Spatial information

One of the main limitations of the bag-of-words-based models is that they do not take into account the spatial distribution of the features. In order to add spatial information to the orderless BoVW model, Lazebnik et al. [15] proposed the Spatial Pyramid Matching (SPM) method. This method roughly takes into account the word distribution over the image by creating a pyramid of spatial bins.

This pyramid is recursively constructed by splitting the images in spatial bins following the vertical and horizontal axis. At each spatial bin, a different BoVW histogram is extracted. The resulting descriptor is obtained by concatenating all the BoVW histograms. Therefore, the final dimensionality of the descriptor is determined by the number of levels used to build the pyramid.

In our experiments, we have adapted the idea of SPM to be used in the context of handwritten word representation. We use the SPM configuration presented in Figure 2 where two different levels are used. The first level is the whole word image and in the second level we divide it in its right and left part and its upper, central and lower parts. With this configuration we aim to capture information about the ascenders and descenders of the words as well as information about the right and left parts of the words. Since we used a two levels SPM with 7 spatial bins, we therefore obtain a final a descriptor of 140.000 dimensions for each word image.

Figure 2. Second level of the proposed SPM configuration. Ascenders and descenders information and right and left parts of the words is captured.

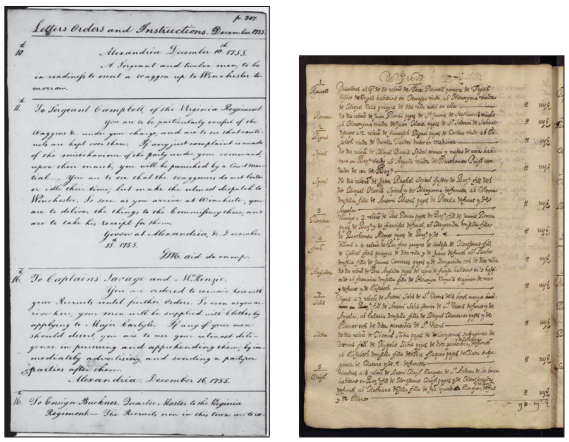
D. Normalization and Distance Computation

Finally, all the word descriptors are normalized by using the $L2$-norm. In order to assess whether two word images are similar or not, we use the cosine distance between its feature vectors.

III. Datasets and Evaluation Measures

To perform the experiments, we used two datasets of handwritten documents that are accurately segmented and transcribed. All the words having at least three characters and appearing at least ten times in the collections were selected as queries. The first image corpus (GW dataset) consists of a set of 20 pages from a collection of letters by George Washington [16]. It has a total of 4860 segmented words with 1124 different transcriptions. That is 1847 word snippets that are taken as queries, and that correspond to 68 different words. The second evaluation corpus (BCN dataset) contains 27 pages from a collection of marriage registers from the Barcelona Cathedral [17] having 6544 word snippets with 1751 different transcriptions. In that collection we use 514 queries from 32 different words. We can see an example of both datasets in Figure 3

In order to evaluate the performance of the different user interaction methods in a word spotting framework we



a) b)

Figure 3. Example of pages from the a) George Washington and b) Barcelona Cathedral collections.

have chosen to report the mean average precision mAP measure [18]. Given the retrieved and relevant sets to a query, ret and rel respectively, the mean average precision is computed using each precision value after truncating at each relevant item in the ranked list. For a given query, let $\mathrm{r}(n)$ be a binary function on the relevance of the n-th item in the returned ranked list and $P@n$ the precision considering only the n topmost results returned by the system. The mean average precision is then defined as follows:

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

IV. Query Fusion

One of the classic ways to enhance the retrieval results in an IR scenario is to cast several queries instead of a single one and somehow combine the results. This is particularly interesting when the queries come from different modalities. In the case of word spotting, asking the user to provide several instances of the sought word might be advantageous in order to overcome the variability of handwritten words.

We have tested three different fusion strategies. One early fusion strategy where the queries are combined before performing the retrieval and two late fusion strategies where we perform as many retrieval as different queries and the ranked lists are then combined. Let us detail these three fusion methods.

- **Early fusion** is achieved by simply averaging the query image descriptors and then normalizing again by the $L2$-norm.
- **CombMAX** is a late fusion method that assigns the maximum of all the scores per word image in the result lists and re-sorts the final list.
- **Borda Count** is also a late fusion method in which the topmost image on each ranked list gets n votes, where n is the dataset size. Each subsequent rank gets one vote less than the previous. The final ranked list is obtained by adding all the votes per image and sorting.

V. Relevance Feedback

The most natural way to take into account the user in an IR application is by means of relevance feedback. After an initial retrieval step, the user is asked to provide some feedback about which results were correct and which were incorrect. This feedback about relevance allows to provide an enhanced result list in the subsequent iterations.

Here, we have tested three different relevance feedback methods from two different families. The Rocchio and the Ide methods, are relevance feedback algorithms that follow the idea of query reformulation whereas the relevance score method is a re-ranking method. Relevance feedback methods that follow the idea of query reformulation try to find, given the relevance assessments, a new query point in the vector domain that is closer to the positive samples and farther to the negative ones than the original query point. On the other hand, re-ranking methods, such as the relevance score method, try to reorganize the original resulting list in terms of the relevance assessments without casting any new query. Let us detail these three relevance feedback methods.

A. Rocchio's Algorithm

The Rocchio's algorithm [9] is one of the most widely used relevance feedback strategies in the IR field. At each relevance feedback iteration, the Rocchio's algorithm computes a new query point in the descriptor space aiming to incorporate relevance feedback information into the vector space model. The modified query vector $\mathbf{q}_m$ is computed as

$$\mathbf{q}_m = \alpha\mathbf{q}_o + \frac{\beta}{|D_r|}\sum_{\mathbf{d}_j \in D_r}\mathbf{d}_j - \frac{\gamma}{|D_n|}\sum_{\mathbf{d}_j \in D_n}\mathbf{d}_j, \quad (2)$$

where $\mathbf{q}_o$ is the original query vector, and D_r and D_n the sets of relevant and non-relevant handwritten word images that the user has marked respectively. α, β and γ are the associated weights that shape the modified query vector with respect to the original query, the relevant and non-relevant items. In our experimental setup we have chosen the following values $\alpha = 1$, $\beta = 0.75$ and $\gamma = 0.25$.

B. Ide Dec-hi Method

The Ide dec-hi method [19] is a variant of the Rocchio's algorithm usually known to perform slightly better in most of the IR scenarios. Instead of considering all the non-relevant items, it just takes into account the topmost ranked non-relevant item d_{non} in order to compute the modified query vector as

$$\mathbf{q}_m = \alpha\mathbf{q}_o + \beta\sum_{\mathbf{d}_j \in D_r}\mathbf{d}_j - \gamma\mathbf{d}_{non}. \quad (3)$$

Table I
mAP FOR VARIOUS QUERY FUSION STRATEGIES

	Baseline	Early F.	combMAX	Borda
GW	0.4219	**0.50409**	0.46813	0.44749
BCN	0.3004	**0.43471**	0.38803	0.39929

Table II
mAP FOR VARIOUS RELEVANCE FEEDBACK METHODS

	Baseline	Rocchio	Ide	RS
GW	0.4219	0.48215	**0.60345**	0.56977
BCN	0.3004	0.41532	**0.47197**	0.36321

In our setup we experimentally set the weighting values to $\alpha = \beta = \gamma = 1$.

C. Relevance Score

Finally, the relevance score algorithm presented in [20] by Giacinto and Roli is a re-ranking method. The idea behind the algorithm is that for each word image in the resulting list we assign the ratio between the nearest relevant and the nearest non-relevant word images as the new score for this particular image. The relevance score RS is computed as follows:

$$RS(\mathbf{x}, (D_r, D_n)) = \left(1 + \frac{\min_{\mathbf{d}_j \in D_r} d(\mathbf{x}, \mathbf{d}_j)}{\min_{\mathbf{d}_j \in D_n} d(\mathbf{x}, \mathbf{d}_j)}\right)^{-1}, \quad (4)$$

where $\mathbf{x}$ is the feature vector of any image in the dataset and $d(\cdot, \cdot)$ is the cosine distance between two handwritten word descriptors. The new resulting list is obtained by re-ranking the word list in terms of their relevance scores.

VI. Experimental Results

First, we can see some qualitative results for both collections in Figure 4. Although some false positives appear in the first ten responses, it is interesting to notice that this false positive words are visually similar to the query.

A. Query Fusion

In order to test the fusion methods we ask the user to cast three simultaneous queries to the system. For each collection all the possible combinations of three queries for all the word classes are tested and the mAP averaged. We can observe the obtained results in Table I. We can see that all the fusion methods outperform the baseline method in both collections. In addition, early fusion performs better than the two late fusion strategies for both collections as well. There are no significant differences between the two late fusion strategies.

B. Relevance Feedback

In order to test the three relevance feedback methods, we ask the user to give relevance on the first ten retrieved images. We guarantee that at least one positive and one negative sample are provided by taking the topmost ranked from each category. We can see in Table II the obtained results.

We can observe that when using any of the relevance feedback strategies, the results clearly outperform the baseline handwritten word spotting system for both collections. In both cases the best method is the Ide Dec-hi method which clearly performs better than the rest.

In Figure 5 we show the evolution of the mAP measure depending on how many retrieved images the user has provided feedback. Obviously, the more images the user is asked to mark, the best the final performance is. Although in Table II the performance between Rocchio's method and relevance score varied depending on the dataset, we can see from Figure 5, that when asking for more relevance assessments, we have the same behavior in both datasets, where the Ide and relevance score methods outperform Rocchio's algorithm. Of course, depending on the application, asking for a manual labeling of so much images would not be feasible and a trade-off between manual effort and system's performance has to be achieved.

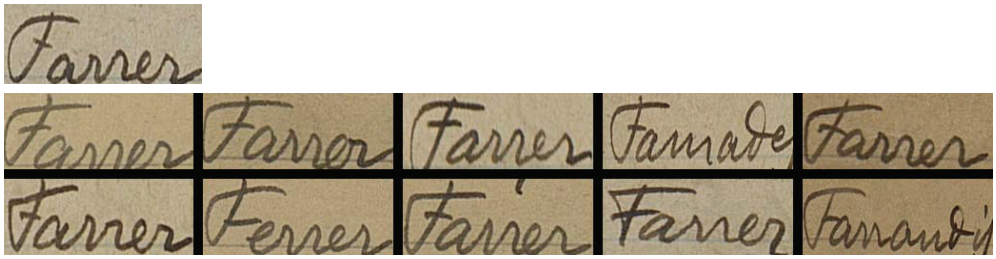
C. Time Complexity

Finally, we report in Table III the average times taken for each of the methods. Regarding the query fusion methods, the early fusion strategy is as costly as the baseline, since in both scenarios just one query is casted, on the other hand, the late fusion methods are more computationally expensive since we cast three queries instead of one. Regarding the relevance feedback experiments, the reported times in Table III correspond to the time to compute the second result list. In that case, both Rocchio and Ide methods are like casting a new query to the system whereas the relevance score method is much more faster since it only has to re-rank the first obtained list. On the other hand, the relevance score method needs to have precomputed all the distances among words in the collection.

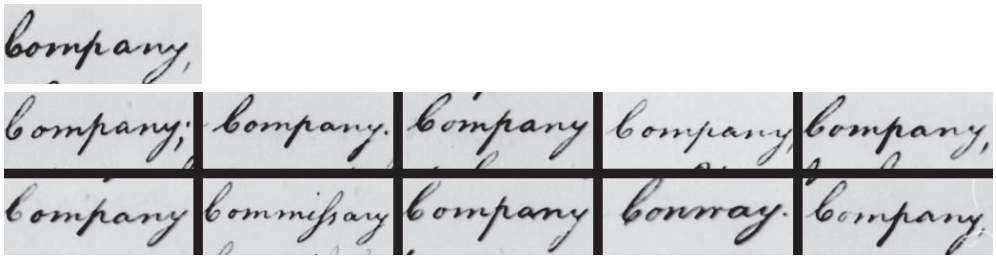
VII. Conclusions and Discussion

In this paper we have presented a study on the inclusion of the user in the loop in a handwritten word spotting scenario. By asking the user to cast several queries instead of a single one or to provide relevance assessments on the retrieval results, we achieve significant increases of performance. Several off-the-shelf methods have been implemented and the performance increase has been demonstrated using two datasets of historical handwritten documents and a baseline word spotting approach based on a bag-of-visual-words model.

Considering that word spotting is a retrieval application, it should be natural that user interaction mechanisms such as relevance feedback are also taken into account when proposing new word spotting scenarios. In our particular setup, the best results were obtained by using the Ide dec-hi method when asking few relevance assessments to the



a)

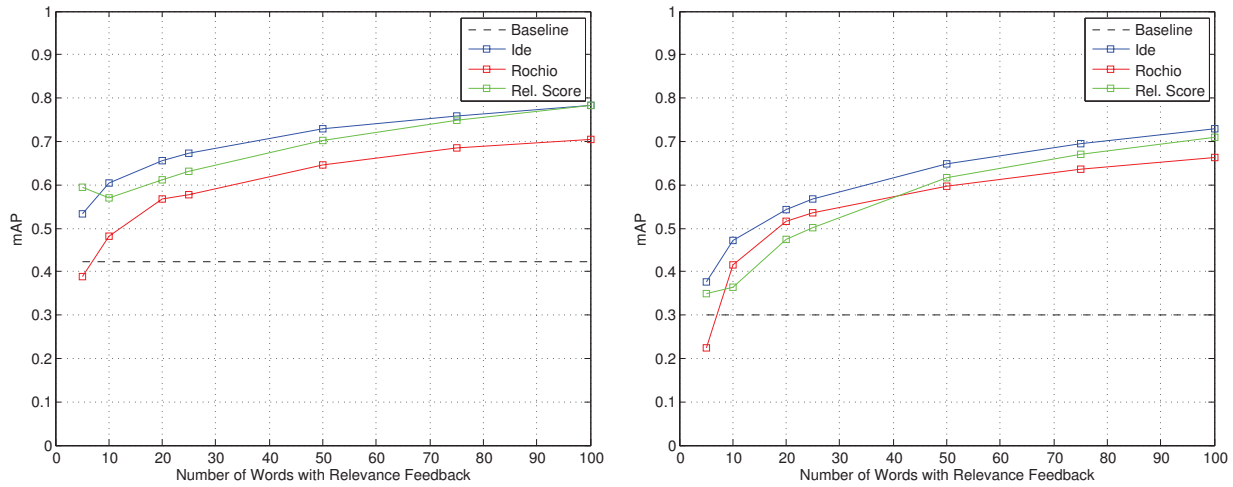


b)

Figure 4. Queries and qualitative results for the a) BCN collection and b) GW collection.

Table III
AVERAGE TIME PER QUERY FOR ALL THE QUERY FUSION AND RELEVANCE FEEDBACK METHODS

	Baseline	Early F.	combMAX	Borda	Rocchio	Ide	RS
Average time (secs.)	0.3429	0.3559	1.0567	1.0331	0.3672	0.3677	0.0968



a) b)

Figure 5. Evolution of the mAP depending on the amount of words with feedback from the user; a) for the GW collection, b) for the BCN collection.

user, whereas when it is feasible to ask for more manual effort from the user, the performance of the relevance score method is also competitive.

As a future research line, we would like to extend this user interaction to other word spotting methods. Here the main problem we face is that most of the tested methods are just valid when the queries are represented by a feature vector of fixed size. Many times, handwritten words are represented by features extracted from columns or sliding windows, such as in [16]. In those cases early fusion strategies are hard to apply as well as query reformulation based relevance feedback strategies as the Rocchio or Ide methods.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Education and Science under projects TIN2008-

04998, TIN2009-14633-C03-03, Consolider Ingenio 2010: MIPRCV (CSD200700018) and the grant 2009-SGR-1434 of the Generalitat de Catalunya.

REFERENCES

[1] R. Manmatha, C. Han, and E. Riseman, "Word spotting: A new approach to indexing handwriting," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1996, pp. 631–637.

[2] T. Syeda-Mahmood, "Indexing of handwritten document images," in *Proceedings of the Workshop on Document Image Analysis*, 1997, pp. 66–73.

[3] J. Rodríguez-Serrano and F. Perronnin, "Handwritten word-spotting using hidden Markov models and universal vocabularies," *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, 2009.

[4] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 211–224, 2012.

[5] A. Fornés, V. Frinken, A. Fischer, J. Almazán, G. Jackson, and H. Bunke, "A keyword spotting approach using blurred shape model-based descriptors," in *Proceedings of the Workshop on Historical Document Imaging and Processing*, 2011, pp. 83–90.

[6] K. Terasawa and Y. Tanaka, "Slit style HOG feature for document image word spotting," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2009, pp. 116–120.

[7] A. Bhardwaj, D. Jose, and V. Govindaraju, "Script independent word spotting in multilingual documents," in *Proceedings of the International Workshop on Cross Lingual Information Access*, 2008, pp. 48–54.

[8] H. Cao, V. Govindaraju, and A. Bhardwaj, "Unconstrained handwritten document retrieval," *International Journal on Document Analysis and Recognition*, vol. 14, no. 2, pp. 145–157, 2011.

[9] J. Rocchio, "Relevance feedback in information retrieval," in *SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall, 1971, pp. 313–323.

[10] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. Perantonis, "Keyword-guided word spotting in historical printed documents using synthetic data and user feedback," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2–4, pp. 167–177, 2007.

[11] A. Kesidis, E. Galiotou, B. Gatos, and I. Pratikakis, "A word spotting framework for historical machine-printed documents," *International Journal on Document Analysis and Recognition*, vol. 14, no. 2, pp. 131–144, 2010.

[12] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2011, pp. 63–67.

[14] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *Computer Vision - ECCV*, ser. LNCS, 2008, vol. 5302, pp. 179–192.

[15] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

[16] T. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2–4, pp. 139–152, 2007.

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

[18] C. van Rijsbergen, *Information retrieval*. Butterworth-Heinemann Newton, 1979.

[19] E. Ide, "New experiments in relevance feedback," in *SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall, 1971, pp. 337–354.

[20] G. Giacinto and F. Roli, "Instance-based relevance feedback for image retrieval," in *Advances in Neural Information Processing Systems*, 2004, pp. 489–496.