e-Crowds: a mobile platform for browsing and searching in historical demography-related manuscripts

Pau Riba, Jon Almazán, Alicia Fornés, David Fernández-Mota, Ernest Valveny, Josep Lladós
Computer Vision Center, Dept de Ciències de la Computació
Universitat Autònoma de Bacelona
pau.riba.fierrez@gmail.com
{*almazan,dfernandez,afornes,ernest,josep*}*@cvc.uab.es*

***Abstract*—This paper presents a prototype system running on portable devices for browsing and word searching through historical handwritten document collections. The platform adapts the paradigm of eBook reading, where the narrative is not necessarily sequential, but centered on the user actions. The novelty is to replace digitally born books by digitized historical manuscripts of marriage licenses, so document analysis tasks are required in the browser. With an active reading paradigm, the user can cast queries of people names, so he/she can implicitly follow genealogical links. In addition, the system allows combined searches: the user can refine a search by adding more words to search. As a second contribution, the retrieval functionality involves as a core technology a word spotting module with an unified approach, which allows combined query searches, and also two input modalities: query-by-example, and query-by-string.**

***Keywords*-word spotting, handwriting recognition, historical documents, mobile application**

I. Introduction

With the advent of devices like Ebooks, tablets and smartphones the reading paradigm is moving from an static to a dynamic mode. During the next years, more and more the information will not be read on printed paper anymore, but rather on electronic devices. The possibilities of electronic readers change the way humans interact with information. Books are no longer static information containers that are read in a sequential way, but reading becomes a human-centered activity where the information flows bidirectionally and generates different narratives depending on the interactions. Thus, users can jump between information items following hyper-links, add annotations, get contextual information or search in dictionaries clicking at words, etc. Let us refer to this way of accessing to the information as active reading.

Digitally born documents are generated with the needed metadata that allows the user to perform the described actions. However, printed books must be digitized and processed prior to the active reading activity. In this work we focus on historical handwritten books, where the active reading paradigm can give value to the access to digitized historical documents. In particular, the application scenario of our system is a collection compiled from the Marriage Licenses Books from the Cathedral of Barcelona, covering five centuries. In these volumes, each marriage record contains information about the couple and their parents, which can be used for demographic research.

Search centred at people is very important in historical research, including family history and genealogical research [1], [2]. Queries about a person and his/her connections to other people allow focus the search to get a picture of a historical context: a person's life, an event, a location at some period of time. The use that a scholar makes of a marriage license book is the genealogical links. Thus, given a particular license recording the marriage of a person, the historian uses the family name to physically access to another book dated between twenty to forty years before and search his/her parent's marriage. This may require significant time and effort, including pooling and cross-referencing many different data sources. This process of jumping from one name in a book to another one can be dramatically improved with the active reading paradigm. Imagine a scholar accessing to an archive of thousands of digitized pages using his/her tablet device, and on the way to his/her work reconstructing the genealogy of a person.

There are very few platforms for visualizing, browsing and searching in historical documents collections in e-books readers and tablet devices. Marinai [3] proposed an off-line tool based on document image processing techniques, that is used for reflowing and annotating scientific documents. Also, Marinai *et al.* [4] proposed a platform for visualizing and reflowing historical documents which is based on the recognition of the printed characters. However, as far as we know, there is still the need of a platform for visualizing and searching in handwritten historical documents. Since the fully automatic recognition of handwritten historical documents is still an open problem (because of different writing styles, degraded documents, etc.), word spotting [5] offers a viable solution for searching and browsing these documents.

The contribution of this paper is twofold. First, we propose a platform for browsing and searching in historical manuscripts. The application runs on an Android tablet, and integrates the functionality of retrieving, while browsing

through, those pages containing instances of queried words (e.g. names, surnames, places).

The second contribution consists in the combined word spotting strategy, allowing not only the combination of queries, but also two input query modalities: query-by-example and query-by-string. Firstly, and concerning the combination of query searches, the aim is to allow the user to add more words to the search in order to refine it. The idea behind is that the contextual information has shown to improve the performance of recognition systems [6], [7], and also, word spotting approaches [8]. Secondly, and to take advantage of the capabilities of portable devices, our system allows two input modalities of keyword search. First, clicking on a word in the current image; second, typing a word. This involves respectively a query-by-example or query-by-string strategy.

To the best of our knowledge, this is a pioneer work presenting a prototype of a system running on an Android tablet unifying and combining different handwritten word spotting modalities in an integrated service.

The rest of this paper is organized as follows. Section II is devoted to describe the system architecture, the application functionalities and the word spotting approach. Section III describes the dataset, protocol and experimental results. Finally, in section IV the conclusions are drawn and the main continuations lines are proposed.

II. System Architecture and Components

The application has been designed with an intuitive front-end interface so the user can easily browse and search by using gestures. This application has different functionalities, that will be described in the following sections. Since two modalities of search are implemented, namely query by example and query by string, a unified word representation has been designed, so both image and string based searches are allowed. The implementation details related to the word representation and distance computation will be described in Section II-B.

A Samsung Galaxy Note 10.1 tablet has been used for implementing the Android application. The specifications are the following: Quad-core processor at 1.4GHz, 2GB of RAM, Screen size of 10.1 inches, and Screen resolution at 1280x800 pixels. It runs Android 4.0.

A. Functionalities

The application has two main functionalities: *browsing* and *searching*. The search functionality can be performed in two ways, namely *simple search* or *incremental search*. In both cases, the two modalities of word spotting are allowed (query by example and query by string).

1) Browsing: The user can browse through the document collection by touching the screen. The next or the previous page is showed by a swiping movement to the left or right. The user will always know the page that is currently displayed because the page number is shown in the bottom left corner of the screen. Zoom in and zoom out are performed by 2-finger press, pinch open zooms into content and pinch close zooms out of content. The user can focus to different parts of the document by a drag movement. This set of gestures follow the standard gesture language of touch devices like tablets and smartphones.

2) Searching: The search interface divides the screen in two panels, the browsing panel and the search panel. When a page is displayed in the main panel (the browsing one), and the user searches a query word, then the search panel shows the query results, sorted by word similarity.

The system allows two searching modalities:

- *Query By String*: The user can select the option "*QBS*" from the menu shown in the top right corner of the screen. When it is selected, a dialog appears asking the user to introduce the query word to be searched.
- *Query By Example*: The user can select a word in the main panel by a long press in the screen. Then, the system shows the selected word in a red box, and asks for confirmation.

If the user agrees, the system shows, in the right side panel, the list of retrieved words in the document collection.

Whenever the user touches a word in the list, the application shows in the main panel the page that contains that word. In case there are more instances of that word in the page, they are all displayed in red bounding boxes (see Figure 1).

3) Combined Query Search: Once a word has been searched, the user can apply two different logic operations, `and` or `or`, in order to search more words related to the previous one. The idea behind is that the user can refine the search: after searching a first word, the user can add more words to the search, emulating a coarse-to-fine search. With the `and` operation, the second word is searched within the results, whereas with the `or` operation, the user is adding an alternative search to the results. These two options are described in detail next:

- `and` (refined search): The menu has an option to allow the user to search a new word that is near the previous word that has been searched. It is similar to the "search within the results" functionality of web search engines. With this option, the user can search for a word (e.g. name, surname) at the same time in order to focus the search in one person for example. The objective of this operation is to retrieve regions of the document that are both likely to contain the query words that are related between them.
- `or` (alternative search): The menu has an option to allow the user to search more than one word in the documents. This functionality can be very useful in case the same word can be written with different spellings or abbreviations, like "*Barcelona*" and "*Barna*". In this way, the user can search both words at the same time.

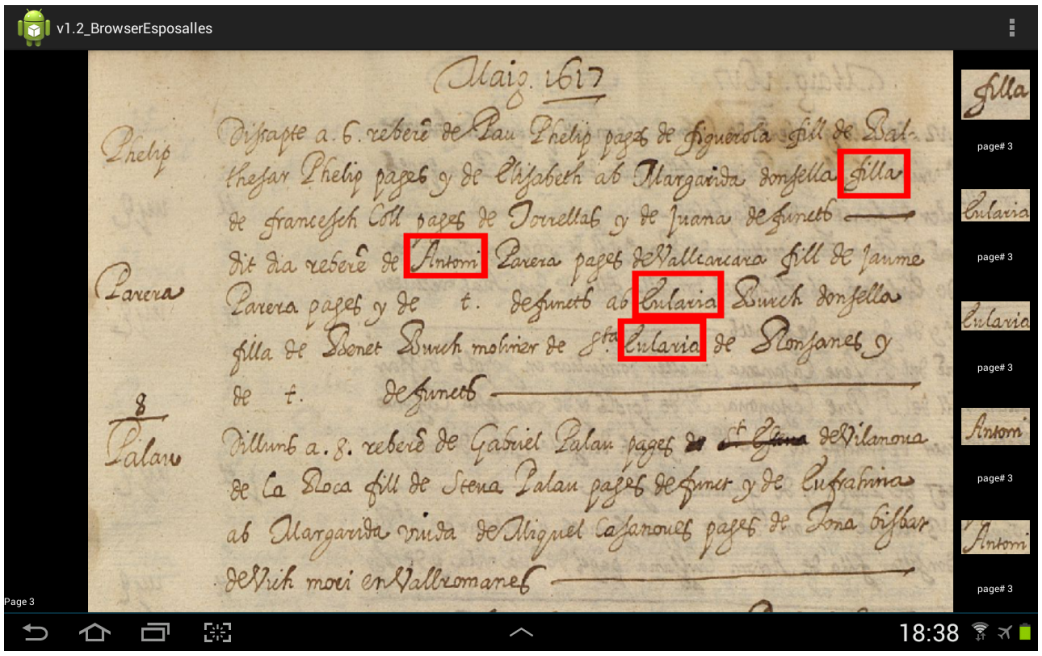

Figure 1. User interface of the Android application. The example shows the regions retrieved with the query "*eularia or filla and antoni*".

These two options can be searched by *Query By Example* too. When the confirmation dialog appears, the user can choose between these two options or perform a new search. Moreover, this combined search can also be done directly in a single string *e.g.* the user can type "*John and Doe*" or "*cat or dog*".

B. Methodology

The application proposed has to rely on a robust method that extracts and represents the information of the documents of the dataset, *i.e.* the words that are contained. We propose to divide this process in two independent steps: first, a segmentation process to extract the individual words of the documents, and then, a feature extraction and representation method of the word images that allows us to query the database.

Different methods have been recently proposed to segment words from handwritten documents [9], [10]. However, we argue that the analysis and comparison of these techniques is out of the scope of the paper. For the evaluation of the application we will rely on the groundtruth information to segment the words of the dataset.

Regarding word representation, we need a robust, low dimensional representation of words – both strings and images – that has to be fast to compute and specially fast to compare. These are hard restrictions imposed by the limitations of the portable device. Arguably, the best representation for this purpose is the attribute-based representation proposed by Almazán *et al.* [11], which proposes a unified representation of string and image words. This method consists in using character attributes (based on a Pyramidal Histogram of Characters (PHOC)), to learn a semantic representation of the word images and then perform a calibration with Canonical Correlation Analysis (CCA) that puts images and text strings in a common subspace. After that, spotting becomes a simple nearest neighbor problem in a very low dimensional space. We refer the reader to [11] for more details about the method.

Once the word candidates of the dataset documents have been represented and projected to the common subspace – this process can be done offline in a desktop computer in a few hours –, these can be stored in the device and loaded by the application when it starts. Then, given a query, this is again projected to the attribute space, in case of an image word, or to the PHOC space, in case of a string word, and then to the common subspace. Now, both dataset images and query lie in the same subspace, and we can compute a fast similarity measure, *e.g.* cosine similarity, to rank the dataset and retrieve first those regions that are more likely to contain the query. We describe now how the lists of results are combined in `or` and `and` queries.

We define $S_1, S_2 \in \mathbb{R}^N$ as the sorted list of similarity scores between the query words w_1 and w_2 and the word candidate regions of the dataset, where N is the number of these regions. In the case of the `or` operation we want to indistinctly retrieve results matching either w_1 or w_2. For that, we simple merge S_1 and S_2 into a single list $S \in \mathbb{R}^{2N}$, which is again sorted and returned as the final result. In the case of the `and` we want to find regions that match both w_1 and w_2 with and additional condition: they

have to be related, *i.e.* they belong to the same record or they are close in the document. In other words, we want to maximize the similarity of the retrieved regions with w_1 and w_2, but minimize the distance between these regions. For this purpose we evaluate all possible combinations between regions retrieved in S_1, S_2, and we weight this combinations according to the distance between them. We compute the final list $S \in \mathbb{R}^{N^2}$ of combined results using the following equation:

$$S_{ij} = S_{1i} * S_{2j} * f(d(i,j), \mu, \sigma),\ i \in 1,..,N, j \in 1,..,N, \quad (1)$$

where $d(i,j)$ is the euclidean distance between regions i and j, and f computes the probability that both regions are related according to the distance between them. For that, function f uses a normal distribution with center μ equals to the location of region i and standard deviation σ as a parameter to be validated. In our case, since the idea of "regions related" means that both regions belong to the same record in the experimental dataset, we set σ equal to an approximate mean height of a record.

III. Experiments

We have evaluated the performance of our application for its main functionality: the *combined query search* task, where different query words are integrated with logical operations, *e.g.* `and` and `or`. We start by describing the dataset used in the experiments and the implementation details related to the proposed method. We then describe the protocol used to measure the performance of the application, and after that, we present the results.

A. Dataset

We have evaluated the performance of our application using as navigation collection the Marriage Licenses Books conserved at the Archives of the Cathedral of Barcelona. These manuscripts, called *Llibre d'Esposalles* [12], consist of 244 books written between 1451 and 1905, and include information of approximately 550,000 marriages celebrated in over 250 parishes (Fig. 2). Each marriage record contains information about the couple, such as their names and surnames, occupations, geographical origin, parents information, as well as the corresponding marriage fee that was paid (this amount depends on the social status of the family). Each book contains a list of individual marriage license records and the corresponding tax payments (analogous to an accounting book) of two years and it was written by a different writer. Information extraction from these manuscripts is of key relevance for scholars in social sciences to study the demographical changes over five centuries. Therefore, the use of a mobile device to access to a document collection of this type illustrates the active reading paradigm where a user can browse through the marriages of a particular year, and search for names of people for a genealogical analysis. The use of combined queries allows searching for the names of a couple in the same query, or to increase the recall making queries combining variations of the target word.

(a) License from 1618 (b) License from 1729

Figure 2. Examples of marriage licenses from different centuries.

B. Implementation details

As we say in Section II-B, we use the attribute-based representation proposed by Almazán *et al.* [11] to represent both strings and images. We split the dataset in two sets: a training set of 134 pages, which contains 43,475 words, to learn the GMM, PCA and attribute models, and a test set of 40 pages, which contains 13,163 words, to evaluate the functionalities of our mobile platform. For training we use the parameters that were validated in [11] for the George Washington dataset. We refer the reader to the experimental section of this work for further details. When computing the attribute representation, we use levels 2, 3, 4 and 5, as well as 50 common bigrams at level 2, leading to 604 dimensions since we consider the 26 characters of the English alphabet and the 10 digits. The common subspace that we learn with the kernelized version of CCA has 256 dimensions, which is therefore the dimensionality that has the representation of the dataset words stored in the mobile platform.

C. Protocol

Given a string query, which is composed by two words, w_1 and w_2, and a logical operation, either `and` or `or`, we rank the dataset according to the similarity between the word strings and the dataset word images, and the logical operation. Concretely, we first obtain two independent lists of regions that are likely to contain each query word, and then, these lists are combined into a single list, whose order depends on the query operation. We report mean Average Precision (mAP) as accuracy measure, which can be seen as the area below the precision-recall curve. However, the way both lists are combined, and how a retrieved result is considered as relevant or not to the combined query is slightly different for both `and` and `or` operations.

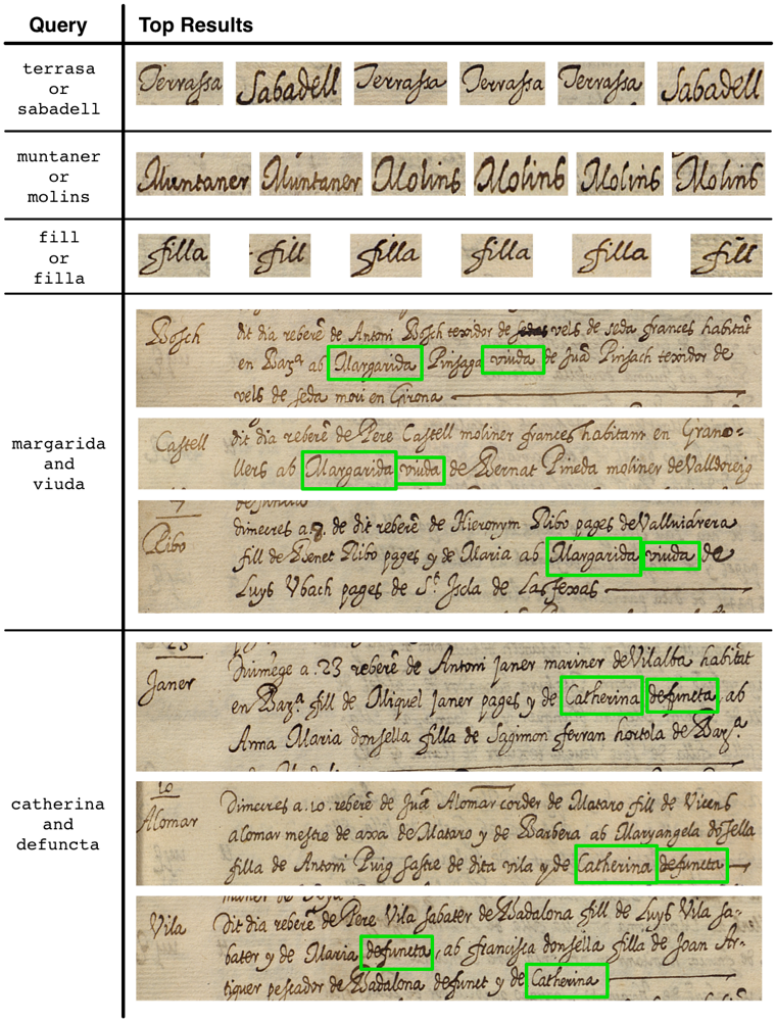

Figure 3. Qualitative results on combined query-by-string word spotting. First three rows use `or` to combine the queried words, where last three rows use `and`.

In `or`, a retrieved result is considered relevant if it matches the class of the word w_1 or the class of the word w_2, and non-relevant on the contrary. The total number of relevant image regions is the sum of the number of relevant regions of w_1 and w_2. In the case of the `and` operation we want to evaluate the ability of our application for retrieving pairs of regions that are both relevant to the queried words and related between them. For that we use the concept of record of the Esposalles dataset: a combined pair of regions is relevant if, first, each word image matches their respective query class, and second, if both regions belong to the same record. The total number of relevant results is the number of records in the dataset that contain both w_1 and w_2.

D. Results

In Table I we show the results obtained by our application for both `and` and `or` query modes. We use a total of 40 randomly selected query combinations, obtaining an average mAP of 90.82%.

Table I
RETRIEVAL RESULTS FOR BOTH QUERY MODES IN THE ESPOSALLES DATASET

Query mode	Num. Queries	mAP
`or`	20	92.45
`and`	20	89.20
Total	40	90.82

Finally, in Figure 3 we show some qualitative results. We can see, in the `or` case, how the regions matching both query words are alternated, and how the `and` operator retrieves records that contains both query words.

According to these results, we argue that the application could be used in a real scenario, being a valuable tool for browsing and retrieving information in this kind of documents.

IV. Conclusion

In this paper we have proposed a mobile platform for browsing, searching and retrieving information in handwritten historical documents. The application allows users to query the database using both images and strings words to retrieve regions from the documents that are likely to contain the query word. Moreover, the application includes the possibility to refine searches with new queries using different combination modes. For that, we use a method that represents both strings and images in a common and very low dimensional space, and propose merging techniques of the results retrieved.

As future work we plan to include a new functionality in the application: the *query-by-sketch-based search*. This is to allow the user to draw their own word queries to perform searches. The sketch is treated as an image, which is used to query the dataset using the same framework.

Acknowledgment

This work has been partially supported by the Spanish projects TIN2011-24631, TIN2009-14633-C03-03, TIN2012-37475-C02-02, by the EU project ERC-2010-AdG-20100407-269796 and by a research grant of the UAB (471-01-8/09).

References

[1] D. J. Kennard, A. M. Kent, and W. A. Barrett, "Linking the past: discovering historical social networks from documents and linking to a genealogical database," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*. ACM, 2011, pp. 43–50.

[2] T. L. Packer and D. W. Embley, "Cost effective ontology population with data from lists in ocred historical documents," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, 2013, pp. 44–52.

[3] S. Marinai, "Reflowing and annotating scientific papers on ebook readers," in *Proceedings of the 2013 ACM symposium on Document engineering*. ACM, 2013, pp. 241–244.

[4] S. Marinai, A. Anzivino, and M. Spampani, "Towards a faithful visualization of historical books on e-book readers," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*. ACM, 2011, pp. 112–119.

[5] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 9, no. 2-4, pp. 139–152, 2007.

[6] M. J. Choi, A. Torralba, and A. S. Willsky, "Context models and out-of-context objects," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 853–862, 2012.

[7] A. Fabian, M. Hernandez, L. Pineda, and I. Meza, "Contextual semantic processing for a spanish dialogue system using markov logic," in *10th Mexican International Conference on Advances in Artificial Intelligence*. Springer, 2011, pp. 258–266.

[8] D. Fernández, S. Marinai, J. Lladós, and A. Fornés, "Contextual word spotting in historical manuscripts using markov logic networks," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, 2013, pp. 36–43.

[9] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis, "Handwritten document image segmentation into text lines and words," *Pattern Recognition*, 2010.

[10] J. Kumar, L. Kang, D. Doermann, and W. Abd-Almageed, "Segmentation of Handwritten Textlines in Presence of Touching Components," *International Conference on Document Analysis and Recognition*, 2011.

[11] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Handwritten word spotting with corrected attributes," in *International Conference on Computer Vision*, 2013.

[12] V. Romero, A. Fornés, N. Serrano, J. A. Sánchez, A. H. Toselli, V. Frinken, E. Vidal, and J. Lladós, "The ESPOSALLES database: An ancient marriage license corpus for off-line handwriting recognition," *Pattern Recognition*, 2013.