Sketch-a-Doc: Using Sketches to Find Documents

Manuel J. Fonseca Daniel Gonçalves
Dept. of Computer Science and Engineering
INESC-ID/IST/TU Lisbon, Lisbon, Portugal
{mjf,daniel.goncalves}@inesc-id.pt

Abstract

With the vast amount of documents that users tend to accumulate in their hard drives, it is natural that they often forget where a certain file is stored or even its name. However, sometimes they still recall a mental image of the document layout. To explore this, we propose a new approach to document retrieval, based on sketches, that capitalizes on human visual memory to help users find their personal documents. The users can sketch the layout of the desired document, using a calligraphic interface, and the system will present to them those that match the sketched query. Documents are processed to extract their relevant features, blocks are segmented and classified according to their contents and a description of the layout is created. A visual language is used to identify the blocks specified using sketches.

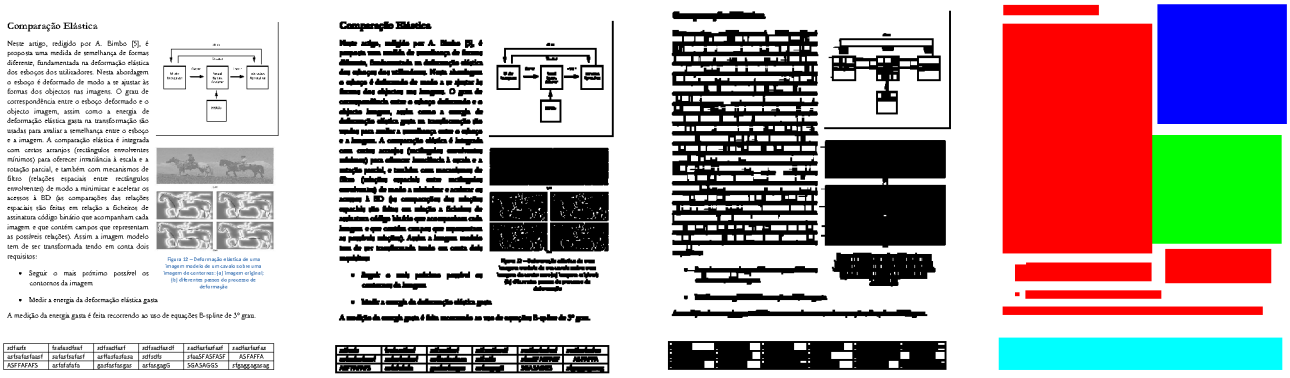
1. Introduction

Nowadays, we sometimes find it hard to locate that document that we know to be somewhere in our hard drive. Usually, this is not a trifling task, especially for older documents. Sometimes when looking for a document, we can not remember its name, location or the typical attributes used in common search tasks, to get satisfactory results. However, occasionally we can recall its appearance, how the first page looked like (it was written on two columns, it had a picture at the top of its rightmost column and a table at the bottom of the document, etc.). The visual memory of the user plays a crucial role in the recognition of objects and also documents. Human beings can easily remember and perceive images rather than words.

In the last decades, as an attempt to help users retrieve their documents based on their appearance, various systems were studied and developed. Some addressed only part of the problem (document processing and segmentation, image indexation, etc.) while others tried to produce complete document retrieval solutions. One system that tried to employ such a method was WISDOM++ [2], which extracts the semantic structure from document images and uses machine learning techniques to automatically label the different blocks identified. Hashimoto and Igarashi [6] presented an approach for web page retrieval based on its layout. Their solution compares a specified layout (using predefined icons) with the layout defined by HTML tags. This approach supports elements like text, images and tables, but does not allow users to freely sketch the page layout. Watai et al. [9] also presented an approach for web page retrieval using visual queries. In their approach users can find web pages by sketching colored regions, which are then compared to screenshots of the web pages using image retrieval techniques. More recently, Lecerf and Chidlovskii [7] presented a schema for querying large document collections by document layout. To speed up search in large databases, authors developed a technique based on clustering to identify a set of representative blocks (images and text) from all the collection and use these to compare with the query.

In this paper, we explore users' visual memory and the fact that one of the best and easiest ways to describe a visual representation of something, in this case a document, is by sketching it, to define a new document retrieval model based on sketches. Our approach combines a richer document layout description with the use of sketches to specify queries. We identify a set of blocks from the document according to their content, namely text areas, images, graphics, tables, horizontal lines and vertical lines. To describe the layout of the document page we developed two techniques. One uses a topology graph to code the spatial arrangement of the page, the distance between the blocks and their type; and another that uses a predefined grid. For the retrieval we developed a calligraphic interface, capable of supporting free hand sketches to specify the layout of the desired document. These sketches are then recognized and their semantic meaning identified through a visual language.

In the remainder of the paper, we explain how we analyze and describe the documents in Section 2, while in Section 3 we explain the use of sketches to specify the queries. Section 4 presents some preliminary results and Section 5



a) b) c) d)

Figure 1. Document analysis process. Left to right: a) the original document; b) the document after thresholding and applying the erosion filter; c) the result of the RSLA; d) the final result, with all blocks colored according to their type (red - text, blue - graphic, green - image, light blue - table).

concludes the paper and enumerates the future work.

2. Document Analysis and Description

Our solution is composed of two main processes. One is responsible for analyzing the document and extract the information about the different blocks (text, image, etc.) that constitute it. The second component produces descriptions of the document in terms of its layout and relationships between the various blocks.

2.1. Document Analysis

To extract the required features from a document we first transform it into a format more amenable for processing, abstracting from the underlying file format (pdf, doc, etc.). To that end, we grab the first page of the document and convert it into an image [5], taking advantage of the several techniques already available for image processing and block segmentation.

After grabbing the first page of the document we convert it into a black and white image, using a basic threshold algorithm. This led to an image reflecting the overall structure of the document page, as illustrated in Figure 1-b. To eliminate some noise caused by the thresholding process we applied an erosion filter with a simple cross flat structuring element (which essentially removes all black pixels not surrounded by other black pixels). This minimizes the number of spurious pixels and improves the efficiency of the block segmentation algorithm.

Many studies have already been made about image and document block segmentation and classification. Some of them are strict rule-based approaches and others more dynamic, resorting to machine learning techniques. For simplicity and effectiveness sakes, we use an improved two-step block segmentation version of the original RLSA (Run-Length Smoothing Algorithm) [8]. This algorithm is able to detect content blocks in a document image, while at the same time classifies those blocks according to their type. To detect content blocks, the RLSA algorithm starts by finding content lines (uninterrupted horizontal sequences of pixels), and then group those closer than a predefined threshold (see Figure 1-c). Although this algorithm is able to identify blocks of the type text, image, graphic, horizontal line and vertical line, it is not able to identify tables, which were one of the block types we needed to identify. To support this we changed RLSA rules and parameters. Now, our algorithm first identifies tables, images, graphics and lines, letting the text blocks for the "otherwise" rule, since the other blocks are easier to recognize than text. Figure 1-d shows the identified blocks.

2.2. Description using a Grid

The grid description is based in spatial organization features only. To that end, we divide the document layout according to a pre-defined 4x10 grid. We chose to partition the document page in 40 units (four columns and ten rows) because after some analysis we concluded that almost no document was formatted in more than three columns. The number of rows was chosen empirically, and ten was the number that held enough expressiveness.

Figure 2 shows a document with the grid overlaid on it. According to this type of description we would get

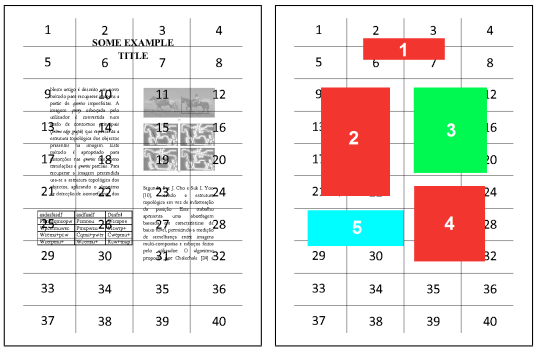


Figure 2. Grid of a document layout.

a specification like this: There is a text block in cells $\langle 2, 3, 6, 7\rangle$, another in $\langle 9, 10, 13, 14, 17, 18, 21, 22\rangle$ and another one at $\langle 23, 24, 27, 28, 31, 32\rangle$; there is an image at $\langle 11, 12, 15, 16, 19, 20\rangle$; and there is also a table located at cells $\langle 25, 26, 29, 30\rangle$. Although this description is very simple, it is good enough to portray the layout of the first page of a document. In this case there are no intersections on the same grid cell, but if there were the algorithm would use the same cell number on the description of both blocks.

These type of description allows us to know the type (or types) of blocks that are present on each of the grid cells. During retrieval, we are able to compare these descriptions with the query grid submitted by the user through a sketch.

2.3. Description using a Topology Graph

We also developed another description mechanism using a topology graph to code the spatial organization of the document page. We extract two types of spatial relationships between blocks (horizontal and vertical connections), the block type (text, image, etc.) and the relative sizes of the blocks to the page size. The block type and the relative size are stored in the graph nodes, while the type of relationship is coded in the links. Figure 3 presents the topology graph for the document page depicted in Figure 2.

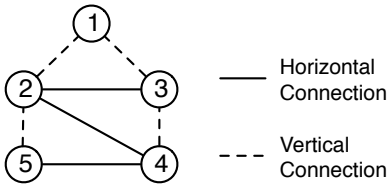


Figure 3. Topology graph describing the layout of the document page in terms of horizontal and vertical connections among blocks.

3. Sketches to Find Documents

To take advantage of the human visual memory, we devised a solution to retrieve documents where users define the layout and the type of blocks through sketches. Users can draw blocks at the desired position and use a simple visual grammar to specify their types.

3.1. Visual Grammar

We defined a visual grammar, based on the studies performed on our previous works [1, 3], to identify the six types of blocks (text, image, graphic, horizontal line, vertical line and table).

$$\texttt{Text} \rightarrow \{\texttt{Rectangle WavyLine}\} \quad if \text{ Contains}(\texttt{Rectangle,WavyLine})$$

$$\texttt{Image} \rightarrow \{\texttt{Rectangle Circle}\} \quad if \text{ Contains}(\texttt{Rectangle,Circle})$$

$$\texttt{Graphic} \rightarrow \{\texttt{Rectangle Triangle}\} \quad if \text{ Contains}(\texttt{Rectangle,Triangle})$$

$$\texttt{Table} \rightarrow \{\texttt{Rectangle Cross}\} \quad if \text{ Contains}(\texttt{Rectangle,Cross})$$

$$\texttt{HLine} \rightarrow \{\texttt{Line}\} \quad if \text{ Horizontal}(\texttt{Line})$$

$$\texttt{VLine} \rightarrow \{\texttt{Line}\} \quad if \text{ Vertical}(\texttt{Line})$$

Terminal symbols on the productions correspond to geometric shapes identified from the sketches drawn by users, using the CALI recognizer [4], as illustrated in Figure 4.

3.2. Matching

Currently we have two algorithms to compare the query with the documents in our database. One uses the grid description while the other uses the topology graph. Our aim is to evaluate both descriptions to figure out which is the best, or if we can obtain better results by combining them.

To perform the matching using the grid description we look for every document that have the same type of block from the query in the same grid cells. For each match the document awarded one point. After processing all blocks from the query we sort the documents by the number of points and present to the user those with the highest scores.

For the description using the topology graph, we compare the topology graph extracted from the sketched query with the topology graphs from the documents.

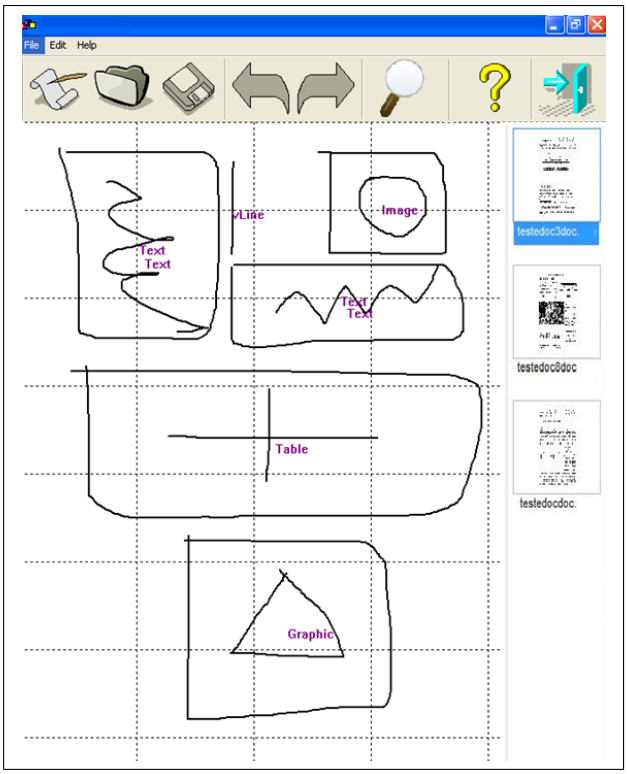


Figure 4. Example of a query specified using sketches and the visual grammar. Results are presented on the right.

4. Experimental Evaluation

Since this work is at a preliminary stage, we did not yet evaluate the two description and matching techniques. Currently we were only able to evaluate the modified RLSA algorithm for block identification.

The RLSA algorithm is sensitive to various parameters used in the identification rules. To choose the most appropriate values for these parameters we performed an experiment in which several values were tried and the quality of the results evaluated. We used a set of 46 documents, representative of different block types, sizes and combinations commonly found in personal documents.

At the end, we were able to infer the parameter values that produces, on average, the best results. Overall, our algorithm is able to correctly identify and classify 87.5% of all blocks.

5. Conclusions and Future Work

In this paper we presented an approach for document retrieval based on their layout and on sketches to specify queries, which takes advantage of the visual memory owned by humans. To that end we modified the RLSA algorithm to identify six types of blocks from the first page of a document, and we developed two approaches to describe the spatial organization of the blocks. One uses a grid while the other uses a topology graph. For query specification we rely on a calligraphic interface where the users describe the layout of the desired document through sketches. We also developed a simple visual grammar to interpret the users' sketches and identify the type of blocks drawn.

Our next steps will be the evaluation of both description approaches to see which one presents better retrieval results and the optimization of the matching algorithms to provide a more efficient solution. One possibility is the inclusion of an indexing mechanism to avoid the comparison of the query with all the documents in the database.

Acknowledgments

This work was supported by FCT (INESC-ID multi-annual funding) through the PIDDAC Program, project A-CSCW (PTDC/EIA/67589/2006) and project Crush (PTDC/EIA-EIA/108077/2008). We thank Filipe Alves for coding the prototype.

References

[1] M. P. Albuquerque, M. J. Fonseca, and J. A. Jorge. Visual Languages for Sketching Documents. In *IEEE Int. Symposium on Visual Languages (VL'00)*, 2000.

[2] M. Berardi, M. Lapi, and D. Malerba. An integrated approach for automatic semantic structure extraction in document images. In *Document Analysis Systems (DAS'04)*, 2004.

[3] A. Caetano, N. Goulart, M. Fonseca, and J. Jorge. Javasketchit: Issues in sketching the look of user interfaces. In *AAAI Spring Symposium - Sketch Understanding*, 2002.

[4] M. J. Fonseca and J. A. Jorge. Experimental Evaluation of an on-line Scribble Recognizer. *Pattern Recognition Letters*, 22(12), 2001.

[5] D. Gonçalves and J. A. Jorge. In search of personal information: narrative-based interfaces. In *Int. Conference on Intelligent User Interfaces (IUI '08)*, 2008.

[6] Y. Hashimoto and T. Igarashi. Retrieving web page layouts using sketches to support example-based web design. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM'05)*, 2005.

[7] L. Lecerf and B. Chidlovskii. Scalable indexing for layout based document retrieval and ranking. In *ACM Symposium on Applied Computing (SAC'10)*, 2010.

[8] F. Shih and S. Chen. Adaptive document block segmentation and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26(5), 1996.

[9] Y. Watai, T. Yamasaki, and K. Aizawa. View-based web page retrieval using interactive sketch query. In *Int. Conference on Image Processing (ICIP'07)*, 2007.