Fast Structural Matching for Document Image Retrieval through Spatial Databases

Hongxing Gao, Marçal Rusiñol, Dimosthenis Karatzas, Josep Lladós

Computer Vision Center, Dept. Ciències de la Computació
Edifici O, Univ. Autònoma de Barcelona
08193 Bellaterra (Barcelona), Spain

ABSTRACT

The structure of document images plays a significant role in document analysis thus considerable efforts have been made towards extracting and understanding document structure, usually in the form of layout analysis approaches. In this paper, we first employ Distance Transform based MSER (DTMSER) to efficiently extract stable document structural elements in terms of a dendrogram of key-regions. Then a fast structural matching method is proposed to query the structure of document (dendrogram) based on a spatial database which facilitates the formulation of advanced spatial queries. The experiments demonstrate a significant improvement in a document retrieval scenario when compared to the use of typical Bag of Words (BoW) and pyramidal BoW descriptors.

Keywords: Document image retrieval, distance transform, MSER, spatial database

1. INTRODUCTION

Document image classification and retrieval, being a crucial step in any digital mailroom scenario, is one of the most explored topics within the Document Image Analysis domain.[1,2] The problem of retrieving *similar* document images to a given query has been tackled from different angles, mainly depending on what is understood as the notion of similarity between documents. In each scenario, depending on the user expectations, the document images have been represented and described by, broadly speaking, three different families of descriptors. Documents can be described by either their textual content,[3] their visual appearance,[4] or their layout structure.[5,6] When dealing with administrative documents, such as invoices, it is accepted by the community that a document description in terms of their structure is more discriminative since both the textual contents and the look-and-feel of the invoice might change even within documents from the same provider. However, layout-based document image descriptors present several drawbacks. On one hand, the descriptor is dependent on the good performance of the layout analysis step that segments the document image in regions or blocks with either physical or logical labels assigned. Such a layout analysis step is not straightforward and is a far from a solved research problem. On the other hand, usually comparing structural relationships among those blocks (typically expressed as graphs) requires a computationally expensive alignment process that hinders the scalability of the final retrieval application.

Avoiding a full layout analysis approach, methods for document classification based on local features matching have been proposed. Chen et al[7] employ a direct matching of keypoints to retrieve structured documents. Spatial consistency is ensured by the final homography calculation step, which assumes that part of the content is exactly replicated in all documents of the same class. In a different approach Kumar et al[8] pool local features through recursive horizontal and vertical splits in a variant of the spatial pyramid approach, imposing in this way a degree of spatial consistency. Such approaches do not explicitly encode structural information, but rather the spatial distribution of local patterns.

In order to overcome the drawbacks associated to layout analysis and large scale comparison of structural descriptors, we propose in this paper to combine a lightweight and generic step to extract stable local regions and the spatial relations between them together with the power of *spatial databases* that, to our best knowledge, have been neglected in our community up to now.

Further author information: hongxing@cvc.uab.es, Telephone: +34 935811828

Spatial databases[9] are optimized databases that store geometrical objects such as points, lines, polygons, etc. allowing to cast queries in terms of geometrical relationships among those objects in an efficient fashion. For example such databases support queries such as *retrieve all the objects having a border close to point A that overlap with circle B and intersect with the polygon C*". They are widely used in various Geographical Information System (GIS) applications such as maps, national census, car navigation, global climate change research, etc.

Specifically, in this paper, we propose a new efficient structural matching method for document images based on stable key-region detection and advanced fast structural querying. First of all, Distance Transform based MSER (DTMSER)[10] is employed for detecting multi-level stable key-regions as well as extracting the dendrogram that defines the structural relationships among these regions. Afterwards, SIFT descriptor and hierarchical k-means clustering are used to generate a codebook while hard assignment is applied for assigning one label (codebook keyword) to each key-region. The main contribution of the paper is the introduction of quick structural matching using spatial databases. For fast structural querying, all the key-regions are stored into a spatial database in terms of the assigned labels and their bounding boxes. Advanced indexes on structural relations among all key-regions are then built based on the spatial locations of their bounding boxes. During query time, key-regions are extracted and labelled from the query image and queries are built based on pair-wise relationships between the extracted regions as obtained from the DTMSER dendrogram. Results in a document retrieval scenario show that the proposed method performs significantly better than typical alternatives based on spatial pyramid and BoW descriptors.

The rest of this document is structured as follows. In Section 2, we introduce spatial databases and their advantage for processing spatial relationships among the stored objects. In Section 3, we explain the key-region extraction and labelling process . In Section 4, by introducing spatial database into document domain, we propose an efficient method for matching the document structure. The experimental results are discussed in Section 5. Concluding remarks and future work are given in Section 6.

2. SPATIAL DATABASE

A *spatialdatabase* is a special type of database that has convenient characteristics such as optimized storage and querying of data, and the ability to represent data objects according to their spatial properties. Spatial databases are optimised to store and query data related to objects in space such as points, lines and polygons while in contrast a typical relational database is designed to efficiently manage data organised according to the relational model. As such, spatial databases are a cornerstone of contemporary Geographical Information Systems (GIS).

In the spatial database domain, a lot of work has been done for building indexes such as R-tree, R^+-Tree, R^*-tree, which are designed for efficiently and spatially index geometries according to their locations. Taking R-tree as example, as showed in Figure 1, it groups nearby objects and represents them with their minimum bounding rectangle in the next higher level of the tree. Since bounding rectangle **A** intersects with rectangle **C** but not with **B**, when querying for certain spatial relationships (e.g. *intersects* or *contains*) between object **F** and others, all the objects lying within **B**(**G**,**H**,**I**) will be automatically neglected. Furthermore, in the case of querying *contain* relationships about **A**, only the objects stored in the child branch **DEF** and its subsequent child branches will be checked, which is very efficient for spatial database.

This type of spatial relations quite naturally correspond to the arrangement of information in documents in hierarchies such as letters, words, paragraphs, or cells, columns, tables. Assuming that such hierarchies can be defined for a document image (e.g. through a layout analysis algorithm), then a spatial database would provide an efficient mechanism for pair-wise structural querying through building advanced indexes for querying spatial relations such as *contain*,*intersect*, *overlap* etc. Nevertheless, although document structure is perceived as important by the community, the potential use of spatial databases as tools to exploit such structure has been overlooked.

In the present paper, we propose an efficient way to extract and utilize document structure for document retrieval. Our proposal can be roughly divided into multi-scale semantic key-region extraction and structural indexing and retrieving, as the pipeline of Figure 2 depicts. In the next section we describe a fast method for defining an hierarchy of document regions while section 4 details how this hierarchy can be stored within a spatial

database and exploited to allow for structure based document retrieval. In this case, we make use of region pairs to form pair-wise structural queries with *contain* spatial relationship, although the use of other types of pair-wise or higher order queries can be easily introduced.

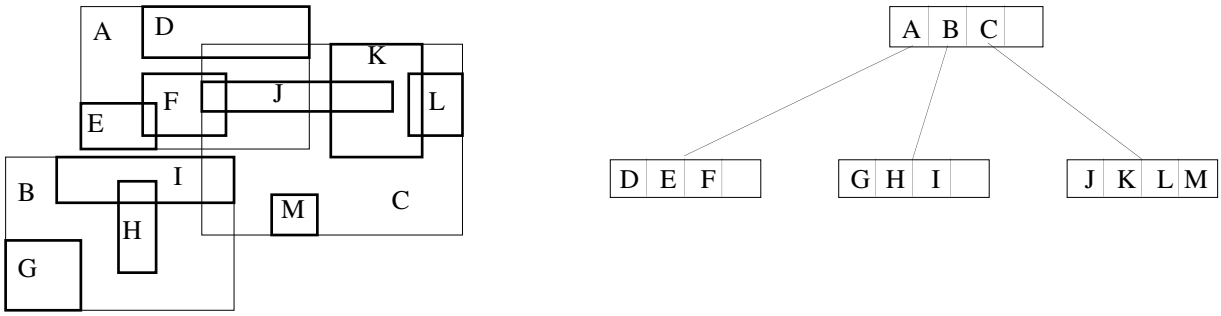


Figure 1. R-tree indexing for spatial database.

3. KEY-REGION EXTRACTION

For document images, the structural relations among document regions offer valuable information for subsequent analysis. Unfortunately, such structure extraction through document layout analysis is impractical because of both inherent instabilities (especially for non-Manhattan layouts) and prohibitive computation complexity for large datasets. As an alternative the authors introduced the Distance Transform Maximally Stable Extremal Regions (DTMSER) detectpr.[10] This is briefly explained below.

3.1 DTMSER detector

In the past decade, various powerful detectors such as SIFT[11] and MSER were proposed for natural scene image analysis. They were also successfully employed in many applications in document image analysis although the semantics of the image contents are distinctly different. For example, the SIFT detector selects extrema of Difference of Gaussian maps which in our case typically correspond to letter corners and also to blank spaces between words. MSER's detection is roughly equivalent to connected component analysis for bi-level images, and would effectively select a bunch of characters in a document image. For document analysis, it is desirable to efficiently identify semantic key-regions at different levels namely characters, words, lines and paragraphs, as well as the hierarchical structure among these elements.

The notion of scale in document image is tightly linked to the distance between the elements of document: characters are usually placed closer to each other than words are, which are in turn placed closer to each other than paragraphs or columns are. Additionally, the hierarchy of these structures is well defined and informative. On the other hand, the MSER algorithm can efficiently perform multi-scale analysis, based on the lightness of neighbouring pixels.

The key idea of the DTMSER detector[10] is to take advantage of MSER algorithm's efficiency to identify stable regions, where stability is re-defined here as a function of distance of a region to neighbouring ones. Effectively, stability implies good separation of a region from neighbouring ones (or equivalently, the existence of adequate white space around it). This notion of stable key-regions directly relates to the semantics of text. Given the spacing of different elements it is reasonable to expect that many key-regions would correspond to semantically meaningful components such as letters, words and paragraphs although needs not be always the case.

Apart from detecting such stable regions, the DTMSER process creates a dendrogram that defines a document structure in terms of what key-regions are eventually combined (contained) into higher-level key-regions (one could think about the semantic relationship of letters to words, words to paragraphs etc).

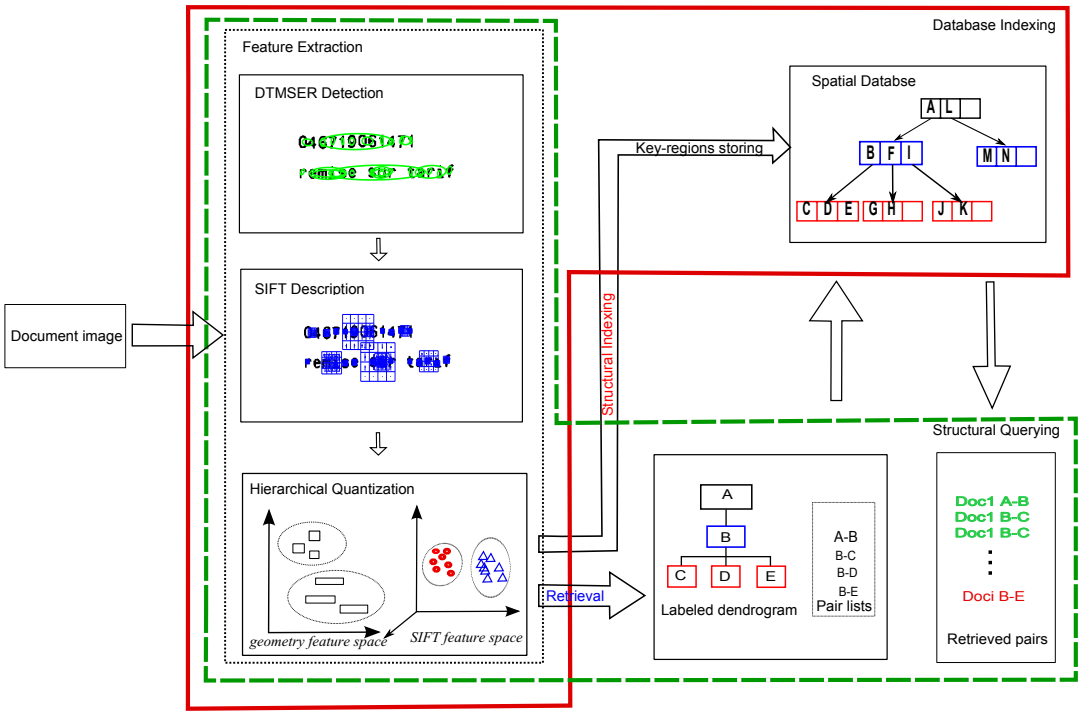


Figure 2. Pipeline of the proposed structural matching.

3.1.1 Distance Transform

For each pixel of the image,the distance transform finds the minimum distances between the corresponding pixel and the set of foreground pixels. The transformed image is a matrix of the same size as the image, where each element is assigned the smallest distance between the corresponding image pixel and the closest foreground object.

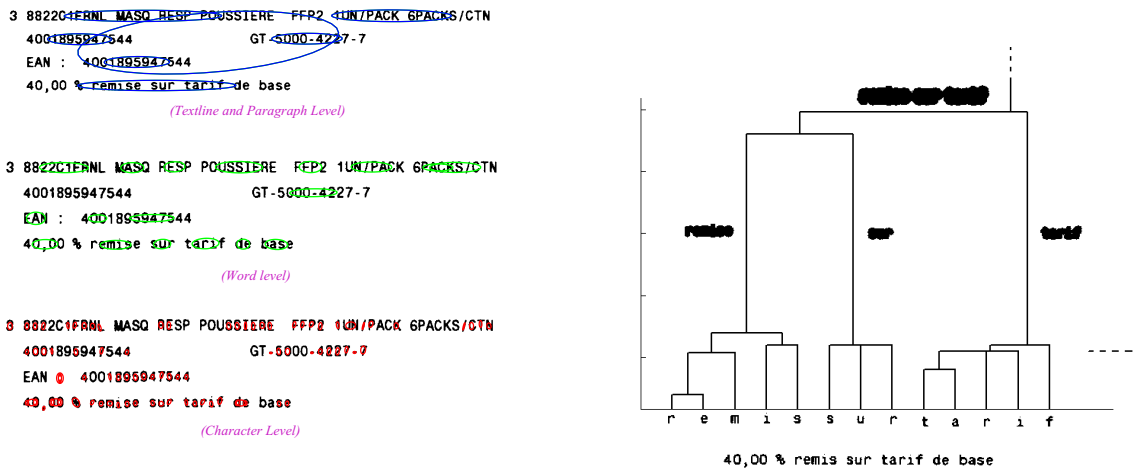
The two pass algorithm proposed by Porikli and Kocak[12] is employed here to compute the distance transform of the document image efficiently. Formally, let p be a background point and q a point from the set of foreground objects Q. The distance transform $f(p)$ assigns at each background point p its distance to the nearest object point by:

$$f(p) = \min_{q \in Q} d(p, q)$$

where $d(p, q)$ is the Euclidean distance.

3.1.2 MSER Detection

The MSER algorithm extracts a set of extremal regions corresponding to all the connected components produced over all possible thresholdings of the input image (see also watershed algorithms) and identifies a set of maximally stable extremal regions, corresponding to extremal regions that remain stable over a certain number of thresholds. In contrast, DTMSER is based on applying MSER algorithm over the distance transform result, thus the resulting maximally stable extremal regions (referred to as "key-regions" from this point onwards) roughly correspond to well separated elements on the page (such as characters, words, text lines, paragraphs).



a) b)

Figure 3. The output of DTMSER method: a) key-regions which roughly correspond to characters, words, text lines and paragraphs; b) dendrogram defining the hierarchy of the key-regions.

Another output of DTMSER detector is the dendrogram of key-regions which explicitly specifies the *contain* relations between them, subsequently used for structural pair-wise querying. In the dendrogram, the leaf regions correspond to the foreground objects, while the subsequent merges depend solely on the distance between the regions. An example of the DTMSER key-regions as well as the corresponding dendrogram is shown in 3.

3.2 Region Descriptor

The SIFT descriptor[11] provides an efficient way to describe the content of a given image patch through a histogram of gradients. We make use of the SIFT descriptor to describe all the extracted key-regions. For each key-region, affine normalization[11] is performend beforehand. The use of an affine-invariant descriptor such as SIFT ensures that scale and rotations variations are efficiently tackled by the framework.

Here, the parameter setting of SIFT descriptor is as follows. Each of the normalized patches is divided into 4 by 4 grids and linear method is employed for interpolating between the spatial locations. The gradients orientations of each of these 16 squares are accumulated into 8 bins. However, all the votes are weighted uniformly (in contrast to the Gaussian weighting of the original implementation) as in the case of document patches the boundary of text is as important as the central part. The length of SIFT feature vectors equals 4*4*8 = 128.

The affine normalization performed before extracting SIFT descriptor by definition suppresses certain geometric features of the key-region such as the absolute size and aspect ratio of the region. On the other hand, such geometrical features can also be helpful as quite different in nature regions may present a similar gradient distribution if the type of contents is similar (e.g. a region corresponding to a text line and one corresponding to a paragraph). Confusing such regions is not desirable. In order to solve this representation drawback, we incorporate two more features of geometric nature, the compactness (area ratio between the key-region pixels and the corresponding bounding box) and the aspect ratio. Including such features is a trade-off as it affects the descriptor's invariance to rotation, but in practical terms it proves to be a reasonable trade-off.

3.3 Codebook Creation

Storing full key-region feature vectors and calculating their pair-wise distances between key-regions is impractical in real life retrieval scenario, we thus quantize the feature space creating a codebook and subsequently use the keywords to label each key-region.

The creation of the codebook takes place in two steps: first based on the geometrical feature quantization and subsequently based on the content feature (SIFT) quantization while an hierarchical k-means clustering algorithm

is employed for both steps. Experimentally, we quantize the geometrical feature space into 10 centroids (sub-spaces) and the SIFT features that lay within each sub-space are then clustered into 100 centroids each. Hence, at the end the codebook consists of 10*100=1000 words.

In summary, key-region extraction generates local key-regions as well as corresponding assigned labels, similarly to a typical BoW framework. However, the strategy described above generates multi-level key-regions that typically carry some semantic meaning (correspond to characters, words, paragraphs), while the relationships between the key-regions is also encoded through the obtained dendrogram.

4. STRUCTURAL INDEXING AND RETRIEVING

Even though the above key-region extraction strategy provides an efficient way to extract the structure of document images in terms of dendrograms of multi-level stable key-regions, the problem of explicitly matching document structures is still unsolved due to computation complexity. The main contribution of this paper is that we explore a strategy for representing document structure as a list of paired key-regions with associated *contain* structural relations, and an efficient retrieving method for this representation through the use of a spatial database. We introduce here how such a database can be used for efficient structure matching, a process which could be divided into spatial indexing and structural retrieving.

4.1 Spatial database indexing for document collections

We store the key-regions into spatial database in the following way: each record corresponds to one key-region which is represented in terms of *document id*, *key_region id*, *label*, *bounding box* and the *area* of the key-region as demonstrated in Table 1. Here, *document id* indicates which document the key-region lies in and *key_region id* indicates which key-region it is in the specific document, *label* specifies which codeword the key-region is assigned given the codebook calculated and the *area* means the size of key-region in term of its number of pixels. Both document id and key-region id are applied as primary key for identifying the key-regions. Based on the location of the corresponding bounding boxes, all the stored key-regions are spatially indexed facilitating the pair-wise *contain* relation retrieval during query time.

Table 1. Data Structure Stored in Spatial Database.

Document id	Key-region id	Label	Bounding box	Area
1	1	680	(107,82),(93,78)	56
1	2	898	(126,82),(111,77)	75
1	3	616	(167,1942),(150,1939)	51
2	1	59	(1718,1748),(1682,1725)	828
2	2	893	(1723,319),(1602,296)	2783
3	1	858	(3267,82),(3251,78)	64
3	2	460	(3281,2202),(3214,2128)	4985

The coordinates of the key-regions are defined locally on the corresponding images. A problem arising with this is that key-regions stemming from different document images may overlap if compared only in terms of their coordinates, while in reality there is no structural relationship as they belong to different documents. There are several ways to prevent from generating such artificial 'structural relations' one of which is to place each document on a different layer, which may lead to a complicated spatial index. Instead, we propose another simple way to solve this problem, namely to define the coordinate system for the document images of the set as showed in Figure 4. Our solution is equivalent to placing all the images one after the other along the x-axis which explicitly guarantees that key-regions from different documents do not present artificial *contain* or *overlap* relations.

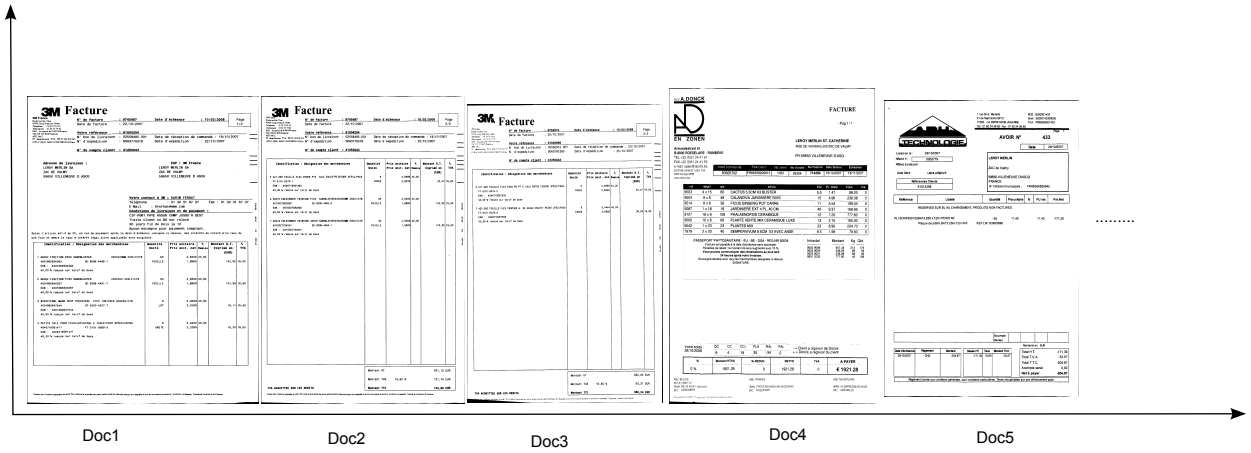


Figure 4. Image coordinate definition.

4.2 Structural retrieval

During query time, we represent the document image as a list of paired key-regions with a *contain* relationship. Such a pair-wise key-region list is extracted based on the dendrogram by considering $parent-child$ pairs which by definition indicate that the *child* key-region is *contained* in the *parent* key-region. For each pair, such structural *contain* relation as well as their labels are used to retrieve all the pairs that possess the same properties from the database. A preliminary ranking list of retrieved pairs for each stored image is computed. Then RANSAC algorithm is applied to refine the matches and re-rank according to the number of returned inliers.

For measuring the structural similarity of full page document images, pairs of key-regions corresponding to higher representation levels (e.g. texline-paragraph level or paragraph-document level) play a more important role than lower representation key-regions do levels (e.g. character-word level). However, the number of lower level pairs is by definition much higher than higher level pairs, provoking the dominant role of low-level pairs during the voting process. To address this problem, we filter out key-regions based on a probabilistic criteria which is the inverse proportion of their corresponding areas. This allows us to filter out a lot of small regions (usually corresponding to character or word level) while bigger regions (usually corresponding to paragraphs) are more likely to be kept. The filtering strategy is also aiming at reducing the retrieval time since less region pairs are generated per query image.

For boosting the pair based querying, various indexes and techniques are employed including:

- **Key-region label indexing**. The first step of pair-wise querying from spatial database is to search for the records that possess the same labels with query pair, resulting in two key-region lists each of which carries the same label of a key-region belonging to the query pair. To boost the label comparing process, we build a b-tree index which is very convenient for dealing with numerical relations like *bigger/smaller than* or *equals* to the given query label (integer).

- **Controlled joining**. After obtaining the two lists of records, the following step for spatial database is to join all records of these two lists together. Theoretically this would lead to the full Cartesian product but, since it is impossible for the key-regions from different documents to have any structural relations, we restrict it to the records that hold the same document id, resulting in a partial Cartesian product. During our experiments, significant improvement on querying time consumption is observed due to this strategy.

- **Spatial index**. The last step to retrieve the spatial database is to examine if the joined records possesses *contain* structural relation. The GiST index is employed for efficiently checking whether the joined records possess the same structural relations as query pair.

5. EXPERIMENT

We perform an experiment on an invoice dataset aiming at retrieving all the invoices that come from the same provider which is assumed to be visually/structurally similar. The dataset we used for the experiment consists of 4109 images from 249 unbalanced classes (providers) and 4.7 million key-regions are extracted from the whole dataset. Leave-one-out strategy is employed to generate query images. The evaluation measures we employ are the Mean Average Precision (MAP) as well as the Precision-Recall curve. To compare against our method we performed the same experiment using typical state of the art descriptors, namely pyramidal decomposition, BoW, pyramidal BoW. The results are summarised in Figure 5 and Table 2. The details of these algorithms' implementation are as follows.

- **Pyramidal Decomposition** We first evaluate the pyramidal decomposition descriptor, namely a spatial pyramid descriptor based on the density of black pixels in each cell. The descriptor is efficient to calculate while the structural information of document images is spatially encoded into the feature vector.[13,14] Since higher pyramid level decomposition only corresponds to further detailed content rather than the overall document layout structure, we set the pyramidal level to a medium scale that is 3, so the resulting dimension of the feature vector for each invoice image is 1+4+16+64=85.
- **Bag of Words (BoW)** We also test the performance of a simple BoW method, based on the same key-regions that we make use of for our method. This method serves as a baseline that allows us to evaluate the contribution of the structural information to the process as it codifies exactly the same information but without making use of any spatial relationship.

  Besides, we also compare our method with the spatial pyramid BoW method that is meant to improve the standard BoW method incorporating certain structural information of the image. As there is a limitation on high dimensional distance calculation, the pyramidal level is set to 3 resulting to a 85,000 (1000+4*1000+16*1000+64*1000) dimension features, because if we used level 4 pyramid the dimensionality would explode to 341,000.
- **Spatial Database** The method proposed here.

To demonstrate the improvement achieved by the proposed structural matching method over BoW and Pyramidal BoW, the key-regions and corresponding labels returned by the key-region extraction process are shared by these three methods.

Table 2. Average MAP of compared methods.

Method	Average MAP
Pyramidal Decomposition	0.8667
BoW	0.9248
Pyramidal BoW	0.9441
Proposed Method	**0.9631**

As demonstrated in Figure 5 and Table 2, since Pyramidal Decomposition encodes only the black density at different cells of the grid while all other details are neglected, it holds much less discriminative power and performs worse than the other three SIFT feature based methods. Between Pyramidal BoW and BoW, a 2 percent improvement is obtained with the Pyramidal BoW benefiting from adding certain structural information. The proposed method, thanks to the explicit representation of document structure gains a significant further 2 percent improvement over the compared state-of-the-art methods.

Benefiting from the advanced techniques, we adapted spatial database for a document dataset, fast retrieval performance is observed during our experiment: $10ms$ per pair yielding $5s$ per query image with an average of 500 key-region pairs employed for structural matching. Another key advantage of the proposed method is that

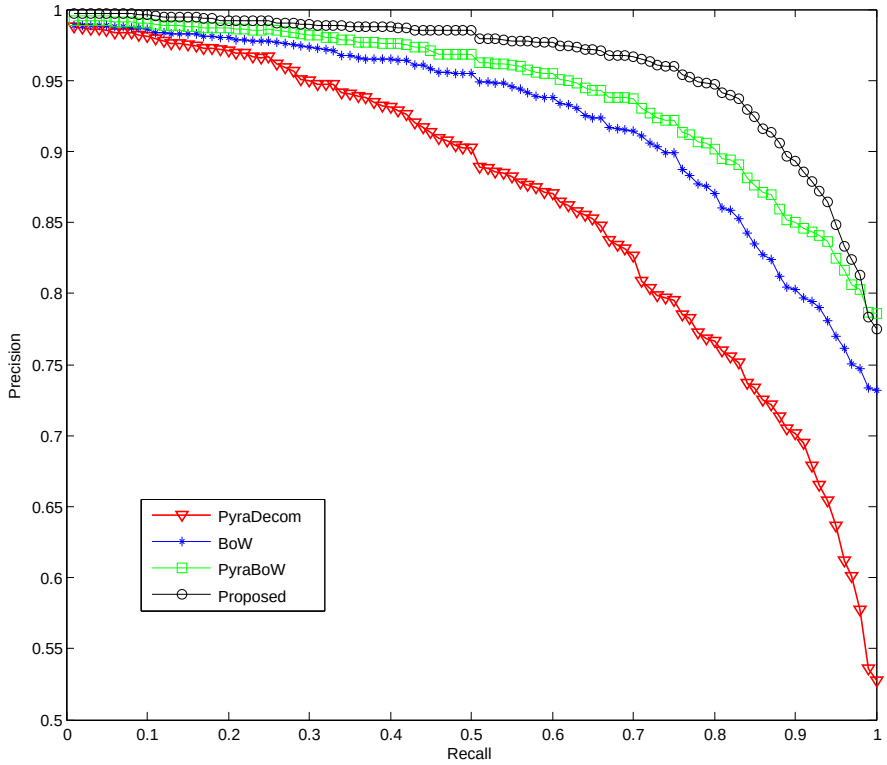


Figure 5. Performance comparison.

it can be readily applied for part-based document retrieval (as opposed to full page queries) while the rest of the compared methods are only suitable for full page document image.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented an efficient structural matching method for document image retrieval based on spatial database as an alternative to layout analysis, which is impractical due to its inconsistency and computational expensiveness. The layout structure of the document is extracted by DTMSER in terms of a dendrogram, which is then represented as a list of paired key-regions based on the *contain* structural relationship. The experiments demonstrate the retrieval performance improvement of the proposed method over BoW and Pyramidal BoW on document structure matching. Through the employment of advanced techniques such as data storage design, advanced GiST index, controlled joining and label indexing, we addressed the problem of time consumption for structural retrieving by pair-based query ($10ms$ per query from a database of 4.7 million records).

Future work may fall into querying more detailed spatial relation like 'overlap', 'neighbouring' or 'left/right of' rather than 'contains' only. Another possible area of research is to improve the sensitivity of the distance transform employed by the DTMSER detector and generalize our proposed method to part-based document image analysis.

ACKNOWLEDGMENTS

This work has been supported by the Spanish projects RYC-2009-05031, TIN2011-24631, TIN2012-37975-C02-02, and China Scholarship Council grant (No.2011674029).

REFERENCES

1. D. Doermann, "The indexing and retrieval of document images: A survey," *Computer Vision Image Understanding* **70**, pp. 287–298, June 1998.
2. N. Chen and D. Blostein, "A survey of document image classification: problem statement, classifier architecture and performance evaluation," *International Journal on Document Analysis and Recognition* **10**, pp. 1–16, June 2006.
3. F. Sebsatiani, "Machine learning in automated text categorization," *Journal ACM Computing Surveys* **34**, pp. 1–47, March 2002.
4. P. Sidiropoulos, S. Vrochidis, and I. Kompatsiaris, "Content-based binary image retrieval using the adaptive hierarchical density histogram," *Pattern Recognition* **44**, pp. 739–750, April 2011.
5. A. Bagdanov, "Fine-grained document genre classification using first order random graphs," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pp. 79–83, 2001.
6. C. Shin and D. Doermann, "Document image retrieval based on layout structural similarity," in *International Conference on Image Processing,Compute Vision and Pattern Recognition*, pp. 606–612, 2006.
7. S. Chen, Y. He, J. Sun, and S. Naoi, "Structured Document Classification By Matching Local Salient Features," in *21st International Conference on Pattern Recognition*, (Icpr), pp. 653–656, 2012.
8. J. Kumar, P. Ye, and D. Doermann, "Learning document structure for retrieval and classification," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1558–1561, 2012.
9. R. H. Güting, "An introduction to spatial database systems," *The VLDB Journal* **3**, pp. 357–399, Oct. 1994.
10. H. Gao, M. Rusinol, D. Karatzas, J. Lladós, T. Sato, M. Iwamura, and K. Kise, "Key-region detection for document images–application to administrative document retrieval," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pp. 230–234, IEEE, 2013.
11. P. Forssen and D. Lowe, "Shape descriptors for maximally stable extremal regions," in *IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.
12. F. Porikli and T. Kocak, "Fast distance transform computation using dual scan line propagation," in *Real-Time Image Processing*, 2007.
13. H. Gao, M. Rusiñol, D. Karatzas, A. Antonacopoulos, and J. Lladós, "An interactive appearance-based document retrieval system for historical newspapers," in *8th International Conference on Computer Vision Theory and Applications*, 2013.
14. P. Héroux, S. Diana, A. Ribert, and E. Trupin, "Classification method study for automatic form class identification," *Proceeding of the International Conference on Pattern Recognition* **1**, pp. 926–928, 1998.