Embedding Document Structure to Bag-of-Words through Pair-wise Stable Key-regions

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*—Since the document structure carries valuable discriminative information, plenty of efforts have been made for extracting and understanding document structure among which layout analysis approaches are the most commonly used. In this paper, Distance Transform based MSER (DTMSER) is employed to efficiently extract the document structure as a dendrogram of key-regions which roughly correspond to structural elements such as characters, words and paragraphs. Inspired by the Bag of Words (BoW) framework, we propose an efficient method for structural document matching by representing the document image as a histogram of key-region pairs encoding structural relationships. Applied to the scenario of document image retrieval, experimental results demonstrate a remarkable improvement when comparing the proposed method with typical BoW and pyramidal BoW methods.**

I. Introduction

In the past decades, considerable effort has been made for document image classification and retrieval in digital mail room and digital office scenarios. Depending on the notion of similarity for the user which varies over different applications, retrieving similar images to a given query is tackled from different perspectives. However, generally speaking, the document images can be described and represented by either their textual content [1], their visual appearance [2],or their layout structure [3], [4].

Layout analysis methods explicitly describe the document images as segmented blocks with assigned logical or physical labels [5]. They represents the document images through their structure that is encoded in a group of high-level blocks (e.g. paragraphs, columns or titles) while the contents inside the blocks are ignored. The performance of layout analysis methods highly depend on the quality of image segmentation which is still a problem far from being solved. Besides, another drawback of layout analysis methods is the distance computation between groups of blocks (normally represented as graphs) since computing the similarity between graphs is widely recognized as time consuming. Both the unstable segmentation and the expensive similarity computation hinders the scalability of the final retrieval application.

Document images are also widely described by their content features either globally (one feature vector per image) or locally (groups of local feature vectors per image). For example, document images are represented globally as a sequence of words sizes in [6] and are expressed as a histogram of both object pixel and crossing number (the number of changes from object to background and from background to object) in [7]. Representing each image globally as one feature vector could achieve high efficiency for full page document image retrieval. However, global descriptors are not generally invariant over affine transformation (rotation, translation, scale or viewpoint perspective change) and not applicable for part-based queries. Alternatively, representing documents as groups of local feature vectors provides ways to address the affine transformation problem and allows for performing either full page or part-based image retrieval. Local key-points or key-regions are detected first and then described by a local content descriptor such as Scale-invariant Feature Transform (SIFT)[8] or Histogram of Oriented Gradients (HOG)[9] etc. For example, the local key-points/key-regions are described with SIFT feature vectors in [10], [11] and as HOG feature vectors in [12], [13]. Representing document images with local content features would achieve good performance when exact matches are expected. However, in the scenario of retrieving similar administrative documents such as invoices within which the textual contents might change while the structural similarity is still kept, lacking of document structure information probably impairs the final performance.

Based on local content description, various strategies have been proposed to compensate the drawback of lacking structural information among which the most straightforward and popular option is adding spatial information like Pyramidal Bag-of Words (Pyram BoW)[14] over Bag-of-Words (BoW)[15]. Normally, the document image is iteratively divided into increasing finer sub-images and is represented as a feature vector concatenating features such as density [16], Run Length encoding (RL)[17] or SIFT[18] extracted from all the resulted sub-images. A problem of the pyramidal spatial method is the dimensionality of the feature vector which increases exponentially, In [19] , the image is recursively partitioned into halves instead of a full grid decomposition to solve the problem of exponentially increasing dimension of the representation feature vector. However, for document image analysis, adding such spatial information to local content feature does not explicitly encode document structure but rather the spatial distribution of local patterns.

The Distance Transform based Maximal Stable Extremal Region (DTMSER)[20] algorithm efficiently extracts the document structure as a dendrogram that defines how the structural elements merge to each other (e.g. characters merge to words, words to paragraphs). The extracted dendrogram is a rich source of structural relations among which $inclusion$ is the most obvious one. Nevertheless, employing such structural relations together with the local key-region feature for document analysis in efficient manner is still under challenge.

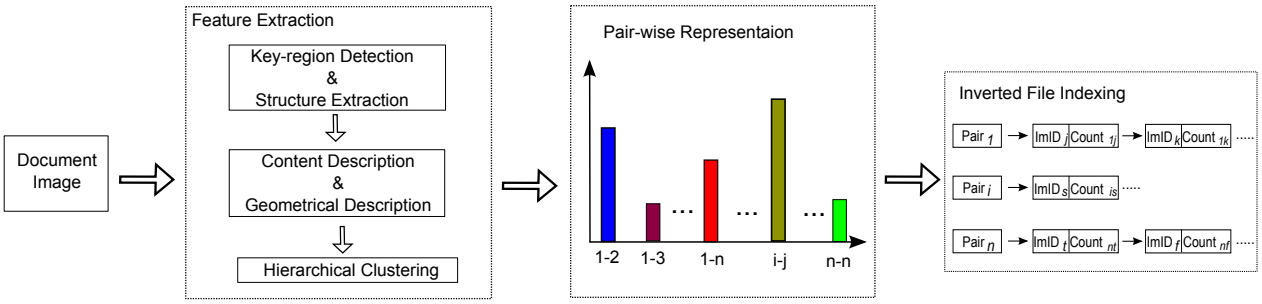

Fig. 1. The pipeline document retrieval based on the proposed pair-wise representation

In this paper, we propose an efficient method that incorporates the BoW method with the *inclusion* structural information which commonly exists between structural elements of document images. The main contribution of this paper is that we embed the explicit document structure together with corresponding content feature into a BoW framework through pair-wise key-region representation. As illustrated in Figure 1, we efficiently extract the document structure as a dendrogram of key-regions that roughly correspond to characters, words, paragraphs and then are described by two types of features: geometrical features and local content features. To generate the codebook, hierarchical k-means algorithm is then employed to quantize the geometrical and local content features in two separate stages. As alternative of BoW which represents images as a histogram of separate key-regions, we propose to express the document image as a sparse histogram of key-region pairs within which *inclusion* structural relation is encoded. Inverted file indexing strategy is employed to efficiently compute the distance between the sparse key-region pair based BoW histograms (will be explained in Section III).

The rest of this document is organized as follows. In Section II, we explain the key-region extraction, label assigning process and the pair-wise image representation. In Section III, inverted file indexing method is explained. In Section IV,the experimental results are discussed. Concluding remarks and future work are given in Section V.

II. Document Structure Extraction

As argued before, layout analysis represents the document structure explicitly, but it is generally exhaustive and inherently unstable. Consequently, in this paper we represent the document structure as a dendrogram defining how characters merge to words and words to paragraphs and so on, which is efficiently extracted by the DTMSER algorithm[20].

A. Distance Transform based MSER (DTMSER)

In the document image analysis domain, it is widely accepted that the document structure (element topology) is tightly related to the varying distance among different level elements. For example, characters are placed closer to each other than words are, which are in turn located closer than paragraphs are. Consequently, DTMSER casts MSER analysis process on the distance transformed image which efficiently returns a hierarchical tree (dendrogram) whose nodes correspond loosely to characters, words, paragraphs. Generally speaking,

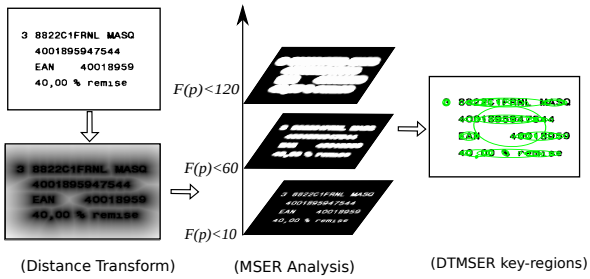

Fig. 2. Progress of Distance Transform MSER algorithm

as showed in Figure 2, DTMSER is implemented in two steps: distance transform and MSER analysis.

1) Distance transform: The distance transform algorithm is proposed to compute the minimum distances of all image pixels to the set of foreground object pixels. For each background point, it computes the distance to the corresponding nearest object point. An efficient computing strategy proposed in [21] with linear computation complexity is employed here.

Formally, assuming p is a background point and q a point from the set of foreground objects Q, the distance transformation could be defined as follows,

$$DT(p,q) = \begin{cases} 0 & p \in Q \\ \min_{q \in Q} d(p,q) & p \notin Q \end{cases}$$

where $d(p,q)$ is the Euclidean distance. And since MSER algorithm only takes grayscale images as input, the computed distance value $DT(p)$ is then normalized to [0-255].

2) MSER: In the MSER analysis step, distance transformed image is taken as input to find the stable regions that survive longer during the thresholding process as showed in the middle column in Figure 2. To extract as many details as possible, the parameter δ defined as the minimum lifetime that the stable region should have during the MSER analysis process, is set to be 1 provoking that all the regions generated by during the thresholding process are recognized as stable regions as long as the corresponding *variation* is less than 1. Afterwards, duplicated regions are filtered out by setting *minimium diversity* to 0.5 provoking that the area of parent stable region should be no less than 2 times of the area of the given stable region. The diversity of a given key-region is

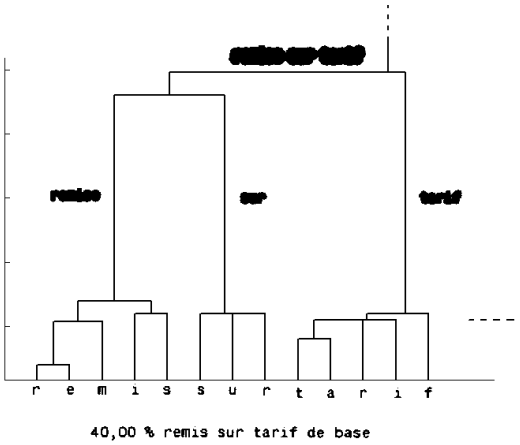

Fig. 3. Dendrogram defining the hierarchy of the structural elements of document images.

defined as follows,

$$div = \frac{P\_area - C\_area}{P\_area}$$

where P_area and C_area represent the area of parent and given stable regions respectively.

The DTMSER algorithm reserves the efficiency of the MSER algorithm for extracting document structure which is represented as a dendrogram roughly defining how characters merge to word, words merge to paragraph and paragraphs to full document. Such type of structural representation is explicit and contains a rich source of $inclusion$ structural relations between parent and child nodes.

B. Feature Description

For each key-region extracted by the DTMSER algorithm, affine normalization [22] process is performed to transform the key-regions with arbitrary aspect ratio into key-regions with squared size facilitating the content feature description algorithm. In the paper, three different descriptors including HOG, SIFT and RL are tested to figure out the optimal solution for describing the extracted key-regions.

In SIFT implementation, each normalized regions is divided into 4 by 4 grids while gradients in each grid are distributed into 8 bins and the feature vector is obtained by concatenating the bins of all grids. Hence, the content of each key-region is described as a feature vector with 4*4*8=128 dimensions. Differently to the standard SIFT implementation, the Gaussian weighting process employed to give more importance to the central part than the border of the patch is ignored here since the border part is considered as important as the central part in the case of document patches. Because of the revised weighting strategy, slightly better performance is observed in our experiment.

For each key-region, HOG features are computed by dividing each normalized region in 4 by 4 cells and then 31 features are extracted from each cell [23]. At the end, 4*4*31=496 dimensional feature vector is returned for each key-region.

The RL descriptor also computes the histogram of the content features but in terms of run length which is defined as the number of pixels with the same value in a sequence. As discussed in [17], we quantize run lengths in a logarithmic manner into 9 bins as follows: [1],[2],[3-4],[5-8],[9-16], ..., [129-Inf]. For binary images in our case, run length yields 2*9=18 bins for both black and white sequences. Besides, we compute runlength feature in horizontal, vertical, diagonal and anti-diagonal directions resulting in 4*18=72 dimensions in the final feature description.

Affine normalization is widely applied in computer vision domain for easily computing the content feature of detected patches. However, it destroys the geometrical feature of the regions. For example, both two key-regions whose sizes are 10*1000 and 50*200 will be normalized into patches with 100*100 provoking that the original geometrical difference is not encoded in content feature descriptions. Hence, to compensate for the lost geometrical information, we describe each key-region with two feature vectors: content feature vector and geometrical feature vector. In this paper, we represent the geometrical features in two dimension: the aspect ratio of the bounding box and solidity ratio of the key-region (the area ratio between key-region and corresponding bounding box).

In summary, each key-region is described in two feature vectors: geometrical feature vector and the content feature vector. The content feature of each key-region could be described by either RL, SIFT or HOG while a validation process for testing which descriptor is the best solution is presented in IV-A.

C. Codebook

To easily search for the matches between the extracted feature vectors, we compute a codebook based on all the key-regions extracted from the dataset images and assign a label to each key-region afterwards. Since each key-region is described as two separate feature vectors, to obtain the codebook, we apply the hierarchical k-means method which is implemented in two stages: k-means for geometrical feature and then k-means for content feature.

To determine the optimal number of clustering centroids in each stage, a validation process is performed while *Num_geom* $\in \{5, 10, 15, 20, 25\}$and *Num_des* $\in \{50, 100, 150, 200, 250\}$ where *Num_geom* and *Num_des* represent the number of centroids for geometrical and content feature respectively.

D. Pair-wise representation

We employed DTMSER to extract the document structure as a hierarchical key-region tree (dendrogram) which contains a rich source of *inclusion* structural relations between the parent node and child node. However, in the standard BoW or spatial BoW framework, each image is represented as a histogram of separate key-regions which ignores the structural relations among them. Consequently, we proposed to alternatively represent each image as a histogram of key-region pairs within which $inclusion$ structural relation is encoded.

The problem of the proposed method that assigns labels to key-region pairs is the size of the codebook is squared. For example, assume the numbers of clustering centroids for geometrical and content feature are set to be 20 and 200

respectively, the codebook size of standard BoW would be $s\_codebook = 20 * 200 = 4000$ while in the case of paired regions it would be $s\_codebook = 4000 * 4000 = 16Million$.

The higher dimensionality of BoW representation would lead to the increased computation complexity for calculating the similarity between images. To address this problem, we apply Inverted File Indexing (IVF)[24] which is independent to codebook size for calculating the similarity between two images.

III. Inverted File Indexing

The proposed method is applied for retrieving images based on structural information, allowing for slight variation on key-region locations. As an example consider in an administrative application, for invoice images from the same provider, the logo location may change from one document to another. Hence the homography calculation process that is usually employed to check the spatial consistency of the matched local patterns is ignored in our case. This strategy could significantly reduce the required key-region storage space and the time consumption of query process.

As showed in Figure 1, the words are stored with the image id and its occurrence time represented here as $count$. During query time, the distance calculation process is only performed for the database images that have at least one matched key-region pairs while other images that do not share any key-region pair are directly ignored. Since the codebook size of the proposed pair-wise method is the square of the standard BoW method, the corresponding histogram vector would be very sparse. Hence, when computing the distance between query and target images, only the non-zero dimension in their representation vector is actually computed. As argued in [25], we employ Cosine distance to calculate the dissimilarity of two images while L2 normalization process is performed in advance. To give more importance to the rare key-region pairs which are more discriminative, the *tf-idf* (Term Frequency - Inverse Document Frequency) [26] weighting scheme is applied.

IV. Experiment

We apply the proposed method to an invoice retrieval scenario. It consists in retrieving invoices from the same provider which hold similar structure while the content (e.g. address, phone number, price, quantity) could change. The experiment is performed based on an invoice dataset containing 4,109 images offered by 249 providers (classes). Overall, 4.7 million multi-level stable key-regions are extracted by the DTMSER algorithm corresponding to approximately 1000 key-regions per image on average. Leave-one-out strategy is applied here to obtain query images (full pages) resulting in a ranked list of 4108 images that is returned according to the similarity scores obtained during query time. To obtain the ground truth, we assume that two images would be structurally similar if they come from the same provider and they would be different if they come from different providers. Mean Average Precision (MAP) is employed here as the evaluation method.

The experiment is discussed in two parts: parameter validation on number of clustering centroids and different type of content feature descriptor and then performance comparison of the proposed method with BoW and Spatial BoW.

A. Parameters Validation

The parameters such as the number of clustering centroids for geometrical feature and the number of centroids for content feature which in turn determine the size of codebook. This size could significantly affect the performance of retrieval methods. Consequently, to Figure out the optimal parameter configuration, a validation process is performed on the mentioned two parameters while the type of content feature descriptor including SIFT, HOG, RL is also taken into account.

To fairly compare the performance of the proposed method with BoW and spatial BoW in Section IV-B, the validation process is also performed for BoW and spatial BoW. The corresponding results of BoW, spatial BoW and the proposed method are illustrated in Figure 4, 5, 6 respectively.

We represent the combination of geometrical and content feature clustering centoirds as Num_geom and Num_des respectively and the two parameters is configured as $Num_geom \in \{5, 10, 15, 20, 25\}$and $Num\_des \in \{50, 100, 150, 200, 250\}$ resulting in 25 parameter combinations. As demonstrated in Figure 4, 5, 6, despite of the descriptor types and the retrieval methods, increasing the number of centroids of either geometrical or content feature will result in a performance improvement. That is because increasing the number of clustering centroids actually leads to the enhanced discriminative power of corresponding features. However, for structural retrieval, this does not indicate that the bigger the codebook size is the better performance since the feature may become to be too discriminative. When the $Num\_geom > 15$ and $Num\_des > 150$, increasing the number of centroids (either Num_geom or Num_des) does not leads to obvious improvement on retrieval performance indicating that the ceiling point is most probably reached. However, taking the slight performance improvement into account, we choose $Num\_geom = 25$ and $Num\_des = 200$ as the optimal configuration for the number of clustering centroids. Besides, since inverted file indexing is applied here, increasing of number of centroids does not leads to higher computation complexity.

Among the considered content descriptors, for most cases, RL performs worst and HOG performs best. This makes sense because RL simply encode the information about number of object pixels which is less discriminative for representing the local content than the gradients information that employed by both SIFT and HOG. For the same type of information (SIFT and HOG), increasing the dimensionality would probably lead to the enhancement of discriminative power. Consequently, the RL descriptor with less discriminative power performs worse than the SIFT and the HOG descriptors. Taking the advantage of higher dimensionality, the HOG descriptor achieves the slightly better performance than the SIFT descriptor. Generally speaking, SIFT obtains more than 2% better performance than RL descriptor and around 1 % or less worse performance than HOG descriptor. Considering their dimension and the resulted computation complexity for assigning labels, SIFT is recognized as the best descriptor here even it performs 1 percent less than HOG because at 4 times calculating time for label assigning process resulting in 1 percent better performance is not "economic" especially in the case of large scale retrieval.

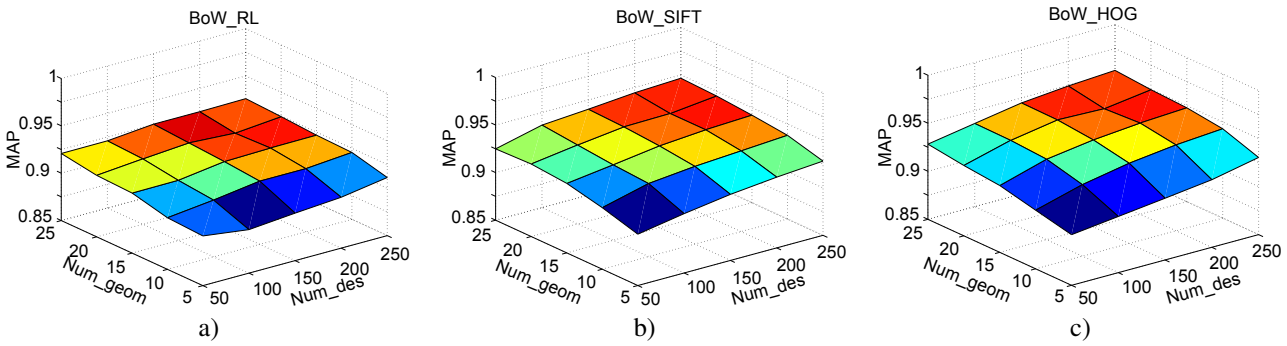

Fig. 4. Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on BoW.

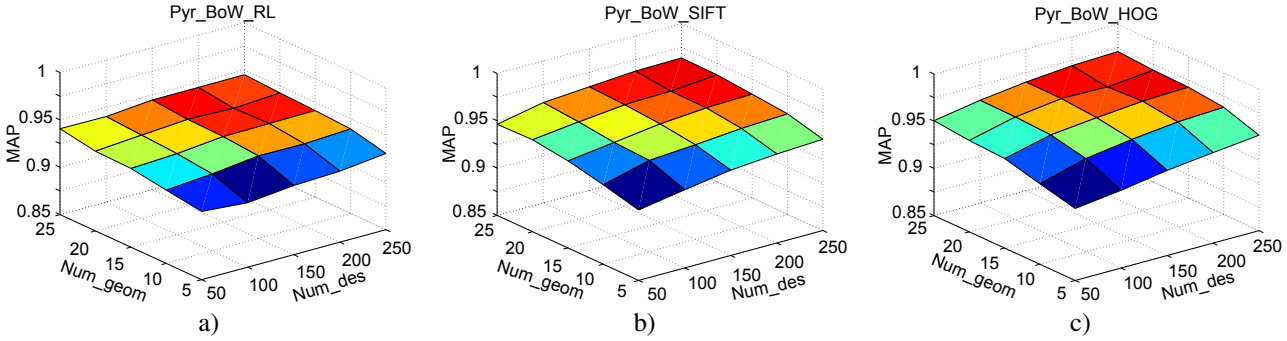

Fig. 5. Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on Pyramidal BoW .

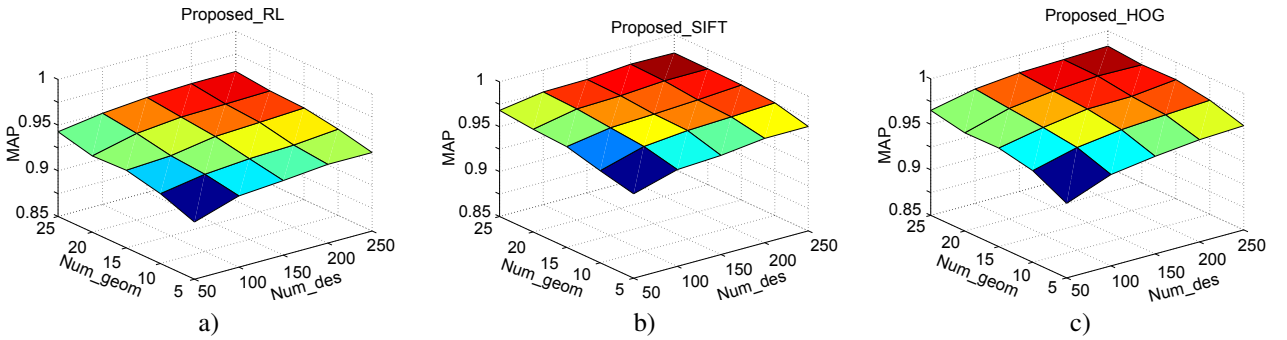

Fig. 6. Clustering parameter Validation of a) Run Length, b) SIFT and c) HOG descriptor based on the proposed method.

In conclusion, $Num\_geom = 25$, $Num\_des = 200$ and SIFT descriptor is considered as the optimal configuration for BoW, spatial BoW and the proposed method.

B. Proposed VS BoW

In this section, we compare the retrieval performance of our method with BoW and spatial BoW based on the parameters validated in the previous section. Both MAP and *precision-recall* curve is employed to demonstrate the performance difference.

TABLE I. MAP PERFORMANCE OF DESCRIPTORS AND FRAMEWORKS($n\_geom = 25$ AND $n\_des = 200$)

	BoW	BoW_Pyram	Proposed
RL	0.9254	0.9448	**0.9559**
SIFT	0.9444	0.9630	**0.9802**
HOG	0.9493	0.9693	**0.9816**

Concerning RL, SIFT and HOG descriptor, table I shows the performance of the considered retrieval methods. As argued in section IV-A, despite of the retrieval methods, SIFT obtains 2% better performance than RL and less than 1% worse performance compared to HOG. Among all the considered retrieval methods, benefiting from the pyramidal spatial information, spatial BoW achieved around 2% improvement over the standard BoW method. Taking advantage of the explicit structure of document images, the proposed method gains around 2% better performance than spatial BoW which represents the document's structure implicitly as spatial distribution of local patterns. The *precison-recall* curve of three compared retrieval methods is plotted in Figure 7 based on SIFT descriptor and optimal number of clustering centroids.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an inverted file indexing based method for structural document image retrieval. The document image is represented as a list of paired multi-level stable key-regions which generally corresponding to character-

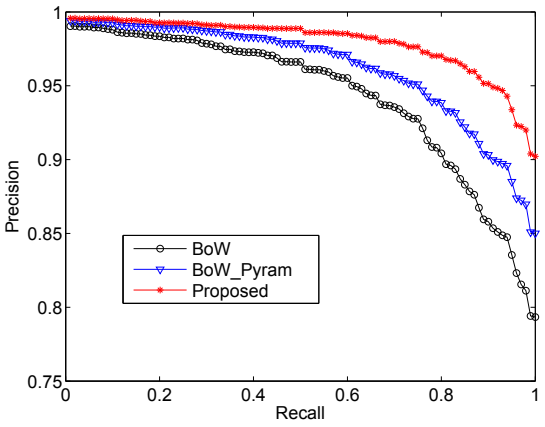

Fig. 7. Precision Recall curve of SIFT descriptor based on BoW, Pyramidal BoW and the proposed method

word or words-paragraph pairs with $inclusion$ structural information explicitly incorporated. Instead of assigning labels to separate key-regions in the case of the BoW method, we assign labels to the key-region pairs provoking that all the assigned labels carry $inclusion$ structural information. The inverted file indexing strategy is employed to solve the computation complexity problem caused by quadratic codebook size.

Under the full page invoice image retrieval scenario, we compared the performance of the proposed method with BoW and spatial BoW method while a validation process on content feature descriptor and number of clustering centroids of both geometrical and content feature is performed.

Future work will go to exploit more structural information rather than only $inclusion$ relation from the extracted key-region dendrogram (hierarchical tree) that carries rich resource of information about document structure. Another possible direction for future work is to test the performance of the proposed method in part-based retrieval or even word-spotting scenario. Besides, adding spatial information in terms of key-region location is also considered as future direction.

VI. Acknowledge

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] F. Sebsatiani, "Machine learning in automated text categorization," *Journal ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, March 2002.

[2] P. Sidiropoulos, S. Vrochidis, and I. Kompatsiaris, "Content-based binary image retrieval using the adaptive hierarchical density histogram," *Pattern Recognition*, vol. 44, no. 4, pp. 739–750, April 2011.

[3] A. Bagdanov, "Fine-grained document genre classification using first order random graphs," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 79–83.

[4] C. Shin and D. S. Doermann, "Document image retrieval based on layout structural similarity." in *IPCV*. Citeseer, 2006, pp. 606–612.

[5] J. Hu, R. Kashi, and G. Wilfong, "Document image layout comparison and classification," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, 1999, pp. 285–288.

[6] J. Li, Z.-G. Fan, Y. Wu, and N. Le, "Document image retrieval with local feature sequences," in *The 10th International Conference on Document Analysis and Recognition*, 2009, pp. 346–350.

[7] G. Meng, N. Zheng, Y. Song, and Y. Zhang, "Document images retrieval based on multiple features combination," in *The 9th International Conference on Document Analysis and Recognition*, vol. 1, 2007, pp. 143–147.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the International Conference on Computer Vision*, 1999, pp. 1150–1157.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005, pp. 886–893.

[10] D. Smith and R. Harvey, "Document retrieval using sift image features," *Journal of Universal Computer Science*, vol. 17, no. 1, pp. 3–15, 2011.

[11] S. Vitaladevuni, F. Choi, R. Prasad, and P. Natarajan, "Detecting near-duplicate document images using interest point matching," in *21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 347–350.

[12] J. Almazn, A. Gordo, A. Forns, and E. Valveny, "Efficient exemplar word spotting," in *Proceedings of the British Machine Vision Conference*, 2012, pp. 67.1–67.11.

[13] K. Terasawa and Y. Tanaka, "Slit style hog feature for document image word spotting," in *The 10th International Conference on Document Analysis and Recognition*, 2009, pp. 116–120.

[14] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[15] F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 524–531.

[16] 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, pp. 84–87.

[17] A. Gordo, F. Perronnin, and E. Valveny, "Large-scale document image retrieval and classification with runlength histograms and binary embeddings," *Pattern Recognition*, vol. 46, no. 7, pp. 1898–1905, 2013.

[18] S. Chen, Y. He, J. Sun, and S. Naoi, "Structured document classification by matching local salient features," in *21st International Conference on Pattern Recognition*, 2012, pp. 653–656.

[19] J. Kumar, P. Ye, and D. Doermann, "Learning document structure for retrieval and classification," in *21st International Conference on Pattern Recognition*, 2012, pp. 1558–1561.

[20] H. Gao, M. Rusiñol, D. Karatzas, , J. Lladós, T. Sato, M. Iwamura, and K. Kise, "Key-region detection for document images – application to administrative document retrieval," in *12th International Conference on Document Analysis and Recognition*, 2013, pp. 230–234.

[21] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," Cornell Computing and Information Science, Tech. Rep., 2004.

[22] P. Forssen and D. Lowe, "Shape descriptors for maximally stable extremal regions," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[23] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[24] J. Zobel and A. Moffat, "Inverted files for text search engines," *ACM Comput. Surv.*, vol. 38, no. 2, Jul. 2006.

[25] M. Aly, M. Munich, and P. Perona, "Indexing in large scale image collections: Scaling properties and benchmark," in *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, 2011, pp. 418–425.

[26] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *INFORMATION PROCESSING AND MANAGEMENT*, 1988, pp. 513–523.