Improving Document Matching Performance by Local Descriptor Filtering

Joseph Chazalon*, Marçal Rusiñol†*, and Jean-Marc Ogier*
*L3i Laboratory, Université de La Rochelle
Avenue Michel Crépeau
17042 La Rochelle Cédex 1, France
†Computer Vision Center, Dept. Ciències de la Computació
Edifici O, Univ. Autònoma de Barcelona
08193 Bellaterra (Barcelona), Spain

***Abstract*—In this paper we propose an effective method aimed at reducing the amount of local descriptors to be indexed in a document matching framework. In an off-line training stage, the matching between the model document and incoming images is computed retaining the local descriptors from the model that steadily produce good matches. We have evaluated this approach by using the ICDAR2015 SmartDOC dataset containing near $25\,000$ images from documents to be captured by a mobile device. We have tested the performance of this filtering step by using ORB and SIFT local detectors and descriptors. The results show an important gain both in quality of the final matching as well as in time and space requirements.**

I. Introduction

Exact document image matching in mobile environments using local descriptors has become in the latest years quite a trend in our community. Applications ranging from model-guided mobile document capture and classification [1], marker-less augmented reality in documents and document retrieval [2] or document stitching and mosaicking [3] are powered by such techniques. At its core, the proposed techniques in the document image analysis domain are no different from classical object recognition scenarios, although many research efforts have been devoted to design accurate and discriminative local descriptors adapted to the specific problem of matching documents.

For instance, Nakai et al. proposed in [2], [4], the Locally Likely Arrangement Hashing (LLAH) method that encodes the word distribution by computing a set of perspective invariant geometric descriptors. In [5], Moraleda and Hull proposed a similar description based also on the way neighboring words are arranged and geometrically distributed.

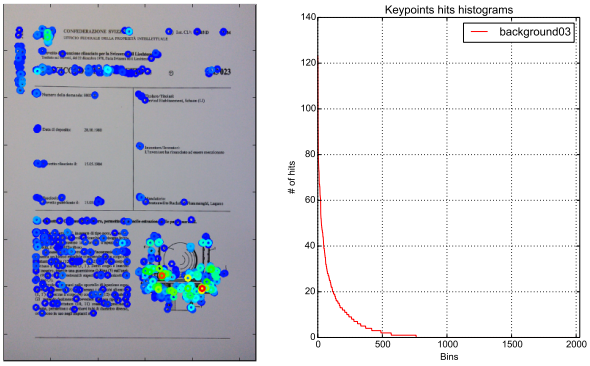
However, despite the importance of designing new and performant local descriptors specifically tailored for document images, one should also consider the efficiency of such frameworks. Specially in the case in which such applications have to run in mobile devices at real time. Although LLAH performs queries in milliseconds in an indexed dataset of millions of pages, it also requires an incredible amount of RAM memory to store such descriptors which is an unrealistic requirement within today's mobile devices specifications.

Binary local descriptors such as BRIEF [6] or ORB [7], in combination with efficient indexing schemes like Locality Sensitive Hashing [8] (LSH), allow to run real-time document matching applications in recent smartphone devices. However, such approach presents a couple of drawbacks. On the one hand, such binary descriptors are not so discriminant when compared with state-of-the-art local descriptors like SIFT [9] or SURF [10] for example. On the other hand, even though LSH allows a direct matching approach in real-time, it might not be that scalable when considering large amounts of documents to index in the dataset.

It is worth to note that usually the main bottleneck in terms of memory requirements is the vast amount of extracted local keypoints. Using default parameters, detecting local keypoints in a 300dpi scanned A4 page with either SIFT, SURF or ORB, usually results in thousands and even tens of thousands of keypoints to be indexed. Obviously, one could trim down such amount of keypoints by adjusting the goodness / saliency parameter of the local detector. However, reducing the amount of local keypoints by adjusting such parameter, does not guarantee that the really discriminative local descriptors are kept, just that the most reliable keypoint locations are retained. One can quickly realize that the most reliable keypoint locations in terms of cornerness, contrast or curvature (delivered by ORB, SIFT and SURF respectively) might not correspond to the most discriminative local descriptors. Specially when applying such frameworks in the document domain, where a lot of repetitive patterns (mostly coming from text) have to be expected.

In this paper we propose a simple but effective method aimed at reducing the amount of local descriptors to be indexed in a document matching framework. Inspired by the work by Kurz et al. [11], we propose to build a "training" sequence of images containing one specific instance of the model document to index. In such sequence, the matching between model and images is computed retaining the local descriptors from the model that steadily produced good matches against this training sequence of images. We have evaluated this approach by using the ICDAR2015 SmartDOC dataset containing near $25\,000$ images from documents to be captured by a mobile device. We have tested the performance of our proposal against the use of both ORB and SIFT local detectors and descriptors. The results show an important gain both in quality of the final matching as well as in time and space requirements.

The paper is organized as follows. In Section II we briefly overview the document matching framework when using local descriptors. Section III introduces the proposed local descriptor



a) b)

Fig. 1. Actual use of local descriptors when matching a document model with some test images. a) Visualization of keypoints, b) histogram of keypoints usage.

filtering stage. In Section IV, we present our experimental setup and the associated results in Section V. Finally, Section VI presents our conclusions.

II. Document Matching

We followed a standard architecture for document matching with local descriptors. Given a set of model documents $D = \{d_1, d_2, ..., d_M\}$ to index, we compute local detectors to end up with N keypoints $K = \{k_1, k_2, ..., k_N\}$ for each model document from D. Since we have run our experiments using the SIFT and ORB detectors, the keypoints from K are at some extend invariant to rotation, scale, illumination and perspective changes. Each keypoint k_i is then described by a feature descriptor f_i from either SIFT or ORB descriptors. Such descriptors are then indexed in an inverted file efficiently implemented with the FLANN architecture [12], which either uses KD-trees for SIFT descriptors or LSH [8] for ORB binary descriptors.

For any incoming image from the mobile device, keypoints and local descriptors are extracted as well and matched against the inverted file. In order to produce reliable matches, we use the ratio-test proposed by Lowe [9], in which a match is considered to be correct if the ratio between the nearest and the second nearest local descriptor is above a certain threshold.

From this set of putative matches, a RANSAC [13] step is performed in order to filter out the outlier matches that to not agree geometrically and to find the homography between the recognized model document and its instance appearing in the scene.

III. Local Descriptor Filtering

As mentioned in the previous section, the document matching framework relies heavily on the keypoints extracted from each document model. Such approach suffers from two major drawbacks. First, the heuristics used by keypoint detectors are based on local criteria (cornerness, contrast or curvature) and do not guarantee a stable matching over various perspectives or illumination conditions. Second, common keypoint detectors tend to extract many ambiguous elements which are usually filtered along the steps of the process.

Those two claims are illustrated with Figure 1. Figure 1a) shows a document model and its keypoints (small circles) originally extracted from it. We appreciate the presence of keypoints covering text areas and other ambiguous parts of the image. Furthermore, this illustration shows the usage frequency of each keypoint with a color scale: blue and green indicate keypoints which are rarely used to estimate the perspective transformed, while orange and red indicate good supports for this estimation. We can see that an important part of the keypoints retained by the local detector are not relevant for the document matching problem. This is is confirmed by Figure 1b), the histogram of usage for each keypoint as a support for the estimation of the perspective transform (i.e. not filtered by the ratio-test and the RANSAC stage).

To improve the selection of robust local descriptors, we followed an approach inspired by the work of Kurz et. al [11]: we propose to build a "training" sequence of images containing one specific instance of the document model to index. In such sequence, the matching between a document model and sample images is computed retaining the local descriptors from the model that steadily produced good matches against this training sequence of images.

More specifically, the steps of the method are the following, given a document model d_i and a train sequence $S_{i,j}$:

1) use a local detector to extract an important amount of keypoints K_i candidates from d_i;
2) build the histogram $h_{i,j}$ of keypoints usage (inliers) when matching d_i (using K_i) with each image of $S_{i,j}$;
3) build a new filtered keypoint set $K_i \backslash t$ with the t keypoints which exhibit the highest histogram values in $h_{i,j}$;
4) use the filtered keypoint set $K_i \backslash t$.

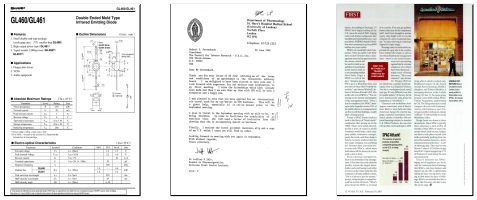
Is is worth noting that the training sequence does not require any special ground truth, except the reference of the document model captured. The selectivity of the matching ensures that only the best local descriptors are retained.

IV. Experimental Setup

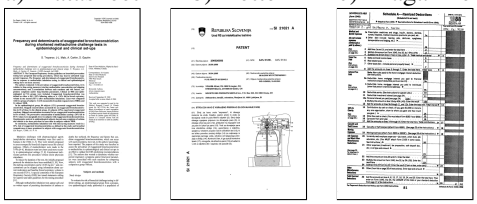
This section presents the dataset, evaluation protocol, metrics we used to evaluate the usefulness of local descriptor filtering.

A. Dataset

Our test dataset is the SmartDOC database for document capture (challenge 1) [14]. It consists of six different document types coming from public databases and five document images per class. The different types have been chosen so that they cover different document layout schemes and contents (either completely textual or having a high graphical content). In particular, the dataset consists of data-sheet documents and patent documents retrieved from the Ghega dataset [15], title-pages from medical scientific papers from the MARG dataset [16], color magazine pages from the PRIMA layout analysis dataset [17], american tax forms from the NIST Tax Forms Dataset (SPDB2) [18], and finally typewritten letters from the Tobacco800 document image database [19]. An example of each of those six different document types is shown



a) Datasheet b) Letter c) Magazine



d) Paper e) Patent f) Tax

Fig. 2. Sample documents used in our dataset. a) Data-sheet from Ghega, b) letter from Tobacco800, c) magazine from PRIMA, d) paper from MARG, e) patent from Ghega and f) tax form from NIST.

Background 1 Background 2 Background 3

Background 4 Background 5

Fig. 3. Sample backgrounds used in our dataset when capturing the same magazine document.

in Figure 2. Some small noise and margins from the original document images were removed and finally the images were rescaled to all have the same size and fit an A4 paper format.

Each of these documents were printed using a color laser-jet and captured using a Google Nexus 7 tablet. Small video clips of around 10 seconds for each of the 30 documents in five different background scenarios were captured (details on each of the backgrounds and capture conditions are given in Table I and Figure 3). The videos were recorded using Full HD 1920x1080 resolution at variable frame-rate. Since the videos were captured by hand-holding and moving the tablet, the video frames present realistic distortions such as focus and motion blur, perspective, change of illumination and even partial occlusions of the document pages. Summarizing, the database consists of 150 video clips comprising near 25 000 frames. In addition of the video clips, an 8Mp picture of each of the documents was captured to be used as models.

This collection was ground-truthed by semi-automatically annotating the quadrilateral coordinates of the document position for each frame in the collection (see details in [20], [14]).

B. Evaluation Protocol

The evaluation we conducted compares the original keypoint selection scheme based on local detectors heuristics (hereafter named *baseline*), and the *filtering* scheme under investigation, detailed in the Section III. We describe here the generation of the test results for each scheme, that we later evaluated with performance measures presented in Section IV-C.

1) Baseline Scheme: The generation of document matching results using local descriptors selected with default heuristics is performed with the following algorithm, where D is the set of document models, T the set of the different numbers of keypoints to *extract*, S_i is the set of video sequences (one for each background) for a given document model d_i.

for each document $d_i \in D$ **do**
  for each keypoint value threshold $t \in T$ **do**
    build K_i^t, a set of t keypoints extracted from d_i with default heuristics
    for each test sequence $s_{i,j} \in S_i$ **do**
      process $s_{i,j}$, matching each frame against d_i using K_i^t
    end for
  end for
end for

2) Filtering Scheme: The generation of document matching results using local descriptors selected by a filtering based on there usage frequency on a training set is performed with the following algorithm, where D is the set of document models, T the set of the different numbers of keypoints to *select*, S_i is the set of video sequences (one for each background) for a given document model d_i.

for each document $d_i \in D$ **do**
  extract K_i^{init}, a set of $init$ keypoints extracted from d_i with default heuristics
  for each training sequence $s_{i,j} \in S_i$ **do**
    build $h_{i,j}$, the histogram of matches of K_i^{init} against each frame of $s_{i,j}$
    for each keypoint value threshold $t \in T$ **do**
      build $K_i^{init} \backslash t$, a set of the t most useful keypoints from $h_{i,j}$
      for each test sequence $s_{i,t} \in S_i, s_{i,t} \neq s_{i,j}$ **do**
        process $s_{i,t}$, matching each frame against d_i using $K_i^{init} \backslash t$
      end for
    end for
  end for
end for

3) Tested Methods: To conduct our experiments, we studied the behavior of the ORB and SIFT detectors and descriptors. We used the following parameters.

- for ORB: $init = 2\,000$, and $T = \{200, 400, 600, \ldots, 2\,000\}$
- for SIFT: $init = 4\,000$, and $T = \{1\,000, 2\,000, 3\,000, 4\,000\}$

For each method, we extracted $1\,000$ keypoints from each frame of the train and test sequences.

C. Performance Measures

1) Segmentation Accuracy: To assess the quality of the results produced by the variations of each method, we used the Jaccard index measure [21] that summarizes the ability of the different methods at correctly segmenting page outlines while also incorporating penalties for methods that do not detect the presence of a document object in some frames.

TABLE I. SMARTDOC DATASET DETAILS

Background	Illumination	Background color	Blur and artifacts	Extra objects
01	Ambient	Brown	Low motion blur	No
02	Ambient	Gray	Low motion blur	No
03	Low light	Light Salmon	Motion and out-of-focus blur	No
04	Intense	Light Gray	Motion blur and highlights	No
05	Ambient	Light Gray	Motion blur	Yes

Using the document size and its coordinates in each frame, we start by transforming the coordinates of the quadrilaterals of a matching method S and of the ground-truth G to undo the perspective transform and obtain the corrected quadrilaterals S' and G'. Such transform makes all the evaluation measures comparable within the document referential. For each frame f, we compute the Jaccard index (JI) that measures the goodness of overlapping of the corrected quadrilaterals as follows:

$$JI(f) = \frac{area(G' \cap S')}{area(G' \cup S')}$$

where $G' \cap S'$ defines the polygon resulting as the intersection of the detected and ground-truth document quadrilaterals and $G' \cup S'$ the polygon of their union. The overall score for each method will be the average of the frame score, for all the frames in the test dataset.

2) Processing Speed: The processing speed is estimated using the average time required to process each frame of the test set. This later is obtained using instrumented Python code and running all the tests on the same dedicated machine.

3) Memory Impact: The memory gain can be directly deducted from the relative reduction of the size of the set of local descriptor for each document model.

V. EXPERIMENTAL RESULTS

This section presents experimental results for ORB and SIFT local detectors and descriptors, and discusses how the training stage should be performed.

A. ORB

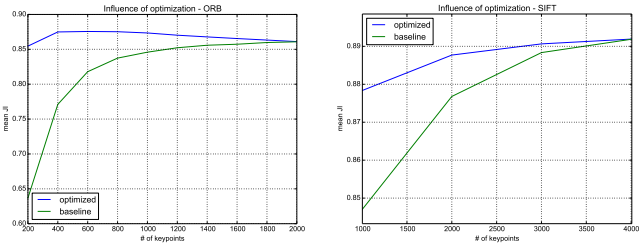
Experimental results exhibit an important performance gain for ORB when filtering local descriptors. Figure 4a) shows an important improvement when progressively removing unnecessary keypoints from the model, until a peak after which the discriminative power of the model starts being hindered. Table II summarizes the values obtained for the Jaccard Index measure. We can see here that a 5-10 factor reduction of the descriptor set size can be achieved almost without harming results' quality. The processing time is also reduced significantly, with a gain of 25% for thresholds of 200 and 400.

B. SIFT

Contrary to ORB, SIFT benefits less from local descriptor filtering which only slows the performance drop, as illustrated by Figure 4b), and summarized in Table III. The slight gain in computing time may not balance the quality loss, and this can make ORB a competitive challenger for some situations.

TABLE II. AVERAGE PERFORMANCE COMPARISON WHEN FILTERING 2000 LOCAL DESCRIPTORS, WITH AVERAGE PROCESSING TIME PER FRAME (ORB)

# descr.	Time (s)	Mean JI	
		Baseline	Filtered
200	0.127	0.637 ± 0.006	0.855 ± 0.002
400	0.127	0.771 ± 0.005	0.875 ± 0.002
600	0.132	0.818 ± 0.004	0.876 ± 0.002
800	0.137	0.837 ± 0.004	0.875 ± 0.002
1000	0.143	0.846 ± 0.004	0.873 ± 0.002
1200	0.149	0.852 ± 0.004	0.870 ± 0.002
1400	0.155	0.856 ± 0.004	0.868 ± 0.002
1600	0.161	0.857 ± 0.004	0.865 ± 0.002
1800	0.166	0.860 ± 0.004	0.863 ± 0.002
2000	0.171	0.861 ± 0.004	0.861 ± 0.002



a) ORB b) SIFT

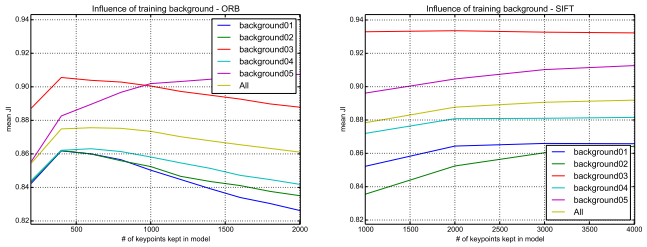
Fig. 4. Influence of filtering over result quality.

C. Training Set Influence

The influence of the training set over the performance of the filtered approach is significant, as illustrated by Figure 5. The average quality difference can be explained by the uneven difficulty of each background, as training on a harder background makes testing easier. We note here, for both ORB and SIFT, that training on videos with extra objects should be avoided (background05), as it leads to poorer performance. In the case of ORB, Figure 5a) shows that almost any other training conditions produces interesting improvements. SIFT, on the other hand, seems to suffer from training with high illumination (background02) and to benefit from low-light or blurry conditions (background03). Standard conditions (background01) also seem an acceptable choice for training SIFT.

TABLE III. AVERAGE PERFORMANCE COMPARISON WHEN FILTERING 4000 LOCAL DESCRIPTORS, WITH AVERAGE PROCESSING TIME PER FRAME (SIFT)

# descr.	Time (s)	Mean JI	
		Baseline	Filtered
1000	1.488	0.847 ± 0.004	0.878 ± 0.002
2000	1.624	0.877 ± 0.004	0.888 ± 0.002
3000	1.718	0.888 ± 0.004	0.891 ± 0.002
4000	1.781	0.892 ± 0.004	0.892 ± 0.002



a) ORB b) SIFT

Fig. 5. Influence of training set over result quality

VI. Conclusions

Inspired by the work by Kurz et al. [11], we studied how a filtering step of local descriptors could influence the performance of a document matching framework. The evaluation was conducted using the ICDAR2015 SmartDOC dataset containing near 25 000 images from documents to be captured by a mobile device, and showed that the selection of local descriptors based on their response to a training sequence could lead to important performance improvements in the case of ORB, and slow the performance drop when reducing the number of descriptors in the case of SIFT. This experiments proved that, as long as the training is performed on sequences with well isolated document models, the framework can benefit from the filtering stage we proposed as it will both reduce processing time and memory requirements.

Acknowledgments

This work is partially supported by the People Programme (Marie Curie Actions) of the Seventh Framework Program of the European Union (FP7/2007-2013) under REA grant agreement no. 600388, and by the Agency of Competitiveness for Companies of the Government of Catalonia, ACCIÓ, and the Spanish project TIN2014-52072-P.

References

[1] M. Rusiñol, J. Chazalon, J. Ogier, and J. Lladós, "A comparative study of local detectors and descriptors for mobile document classification," in *Proceedings of the 13th International Conference on Document Analysis and Recognition*, 2015.

[2] T. Nakai, K. Kise, and M. Iwamura, "Hashing with local combinations of feature points and its application to camera-based document image retrieval–retrieval in 0.14 second from 10,000 pages–," in *Proceedings of the 1st International Workshop on Camera-Based Document Analysis and Recongition*, 2005, pp. 87–94.

[3] M. Luqman, P. Gomez-Krämer, and J. Ogier, "Mobile phone camera-based video scanning of paper documents," in *Proceedings of the 5th International Workshop on Camera-Based Document Analysis and Recongition*, 2013, pp. 77–82.

[4] T. Nakai, K. Kise, and M. Iwamura, "Camera-based document image retrieval as voting for partial signatures of projective invariants," in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, 2005, pp. 379–383.

[5] J. Moraleda and J. Hull, "Toward massive scalability in image matching," in *Proceedings of the International Conference on Pattern Recongition*, 2010, pp. 3424–3427.

[6] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, July 2012.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 2564–2571.

[8] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM – 50th anniversary issue: 1958 – 2008*, vol. 51, no. 1, pp. 117–122, January 2008.

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

[10] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[11] D. Kurz, T. Olszamowski, and S. Benhimane, "Representative feature descriptor sets for robust handheld camera localization," in *Proceedings of the International Symposium on Mixed and Augmented Reality*, 2012, pp. 65–70.

[12] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.

[13] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[14] J. Burie, J. Chazalon, M. Coustaty, S. Eskenazi, M. Luqman, M. Mehri, N. Nayef, J. Ogier, S. Prum, and M. R. nol, "ICDAR2015 competition on smartphone document capture and OCR (smartdoc)," in *Proceedings of the 13th International Conference on Document Analysis and Recognition*, 2015.

[15] E. Medvet, A. Bartoli, and G. Davanzo, "A probabilistic approach to printed document understanding," *International Journal of Document Analysis and Recognition*, vol. 14, no. 4, pp. 335–347, December 2011.

[16] G. Ford and G. Thoma, "Ground truth data for document image analysis," in *Proceedings of the Symposium on Document Image Understanding and Technology*, 2003, pp. 199–205.

[17] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher, "A realistic dataset for performance evaluation of document layout analysis," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, 2009, pp. 296–300.

[18] D. Dimmick, M. Garris, and C. L. Wilson, "Structured forms database," National Institutte of Standards and Technology, Tech. Rep., 1991.

[19] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard, "Building a test collection for complex document information processing," in *Proc. Int. ACM SIGIR Conf.*, 2006, pp. 665–666.

[20] J. Chazalon, M. Rusiñol, J. Ogier, and J. Lladós, "A semi-automatic groundtruthing tool for mobile-captured document segmentation," in *Proceedings of the 13th International Conference on Document Analysis and Recognition*, 2015.

[21] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.