Benchmarking Keypoint Filtering Approaches for Document Image Matching

Emilien Royer, Joseph Chazalon, Marçal Mr Rusiñol, Frederic Bouchara

Benchmarking Keypoint Filtering Approaches for Document Image Matching

E. Royer[⋆], J. Chazalon[†‡], M. Rusiñol[◇] and F. Bouchara[⋆]
[⋆]: Université de Toulon, Aix Marseille Univ, CNRS, ENSAM, LSIS, Marseille, France
[†]: L3i — Univ. La Rochelle, La Rochelle, France
[‡]: LRDE — EPITA, Paris, France
[◇]: CVC — UAB, Barcelona, Spain
Email: emilien.royer@univ-tln.fr

Fig. 1. Illustration of CORE filtering with SIFT (keypoints+features) for $p = 0.01$ and $\sigma = 32.125$ on SmartDOC dataset sample (magazine002). Blue and red points are respectively kept and discarded keypoints.

***Abstract*—Reducing the amount of keypoints used to index an image is particularly interesting to control processing time and memory usage in real-time document image matching applications, like augmented documents or smartphone applications. This paper benchmarks two keypoint selection methods on a task consisting of reducing keypoint sets extracted from document images, while preserving detection and segmentation accuracy. We first study the different forms of keypoint filtering, and we introduce the use of the CORE selection method on keypoints extracted from document images. Then, we extend a previously published benchmark by including evaluations of the new method, by adding the SURF-BRISK detection/description scheme, and by reporting processing speeds. Evaluations are conducted on the publicly available dataset of ICDAR2015 SmartDOC challenge 1. Finally, we prove that reducing the original keypoint set is always feasible and can be beneficial not only to processing speed but also to accuracy.**

I. Introduction

A central pillar of computer vision fields is the design and study of local descriptors. This entry point for numerous applications such as mosaicking, super-resolution, object recognition in natural scenes, etc. has been deeply studied for the last decades and gave us several high-quality contributions like SIFT [1] or SURF [2]. Commonly based on a careful gradient analysis in the image, these kinds of methods are highly valuable in terms of results, but often require somewhat relatively serious processing capabilities which can prevent us from using them for real-time applications, especially on embedded devices with lower computing capacities. Now that smartphone devices usage prevail in everyday life, this aspect cannot be simply ignored anymore. Thus, over the last few years came the recent developments of the binary descriptors somewhat inspired by Local Binary Patterns [3]. BRIEF [4] led the way and inspired many others such as ORB [5], BRISK [6] or FREAK [7]. Most methods follow a similar pattern which consist of building the feature vector by applying successive pixel intensity comparisons in a small patch around the keypoint. Therefore, it makes sense that by doing so they are originally lighter and faster but also less accurate than gradient-histogram-based methods, the last contributions like LATCH [8], BOLD [9], BinBoost [10] or D-BRIEF [11] closed the gap with the best floating-point descriptors, results-wise.

However, such works are not easily used in document image processing. For example, ORB orders FAST keypoints [12] with a Harris corner measure and SIFT idea of saliency relies on a local contrast analysis. These ideas make plenty of sense when working with real-world images but document images are not natural. Indeed, printed text contains a plethora of corners and furthermore if it is black ink on white paper (which is the most common for printed documents). On a scanned A4 paper it is thus common to get up to more than 10.000 keypoints. This leads to several troublesome drawbacks which are the feature vector's loss of discriminability power and huge memory usage.

Thus if we consider the fact cited above regarding the smartphone industry, we can understand why this issue has been well acknowledged in the document field in the last few years. When looking at the literature, we see that two different approaches exist.
The first one is designing entirely new keypoints detectors and feature extractors. An early work on this matter is the SITT algorithm [13] for image document mosaicking which detects keypoints by looking for distinctive patterns like

punctuation symbols. In the context of document matching framework, which is the goal of our contribution, but also more robust, we can cite DTMSER [14] that applies the MSER algorithm [15] on distance transform images for document retrieval. Last, also for document retrieval an impressive and high compelling work in terms of speed and accuracy is the LLAH method [16], [17] which extracts points from word centroids and compute features as geometric invariant combinations.
The second one keeps the classical detectors and extractors for their welcomed properties, but tries to integrate them better in document processing pipelines. A recent work on the subject is a filtering of ORB and SIFT features [18] through histogram analysis (in an offline training stage) of keypoints usage (inliers) when matching images in order to keep only the relevant ones. Another recent work, although with a broader scope than document image processing is the CORE algorithm [19], [20], a generic filtering method for reducing the confusion of feature vectors during the matching step based on a probability approach. (See Fig. 1.)

This paper builds on those previous works to propose the following contributions: first we introduce the use of the recent CORE filtering method [19], [20] on a document image processing application for the first time (Sec. II); second we refine the keypoint filtering pipeline introduced in [18] by considering an extra stage in the process (Sec. III); then we extend the evaluation protocol previously presented in [18] with the addition of a new keypoint detector and descriptor pair, as well as a new reporting method based on the relative reduction measure of the keypoint set to enable the comparison of the CORE method with other ones (Sec. IV); and finally we prove that the CORE method allows a significant reduction of the amount of keypoints to be indexed with several advantages: no need for a supervised training (unlike [18]) and very competitive results both in terms of speed and quality (Sec. V).

II. CORE Algorithm

Earlier we discussed how classical keypoints detectors are not suitable for printed document images. The multitude of responses returned is a real issue for the loss of discriminability but there is also what is called the feature confusion issue. With similar patterns repeated regularly in the scene (in our case that would be letters and words repetitions for example), the matching step tends to be troublesome. As an illustration, let us consider an image I with two feature vectors u_i, u_j that present high proximity in the feature space. Then, another observation I' of the same scene with slight variations such as perspective or lightning changes can also contain the corresponding u'_i and u'_j, but u_i might be closer to u'_j than its rightful match u'_i, thus leading to a mismatch. Regarding what happens in the feature space with floating-point descriptors, we can consider that each vector *may* move for each dimension around a σ distance. As for binary features, since it involves binary digits, the notion of movement is irrelevant here and instead we have a bit-flip probability μ.

The CORE algorithm tries to tackle this issue by removing keypoints with a high confusion risk before any matching step. The main idea is to set for each keypoint i a numerical value C_i tied to its confusion risk. This is done with a Parzen-Rosenblatt kernel density estimator (KDE) [21] and can be used with floating-point descriptors and binary descriptors alike. Regarding the former, the authors use a gaussian kernel which gives the following formulation:

$$C_i = \frac{1}{(N-1)\left(\sigma\sqrt{2\pi}\right)^D} \sum_{j \neq i} \exp\left(-\frac{\mathtt{d_E}(u_i, u_j)^2}{2\sigma^2}\right) \quad (1)$$

with N the total number of feature vectors, D the feature dimension size and $\mathtt{d_E}(u_i, u_j) = \sqrt{\|u_i - u_j\|}$, the euclidean distance between features u_i and u_j.

Since the binary feature vectors consist of a string of binary digits, the approach used here does not rely on Gaussian kernel but on a Bernoulli scheme.

$$C_i = \frac{1}{(N-1)} \sum_{j \neq i} \mu^{\mathtt{d_H}(u_i, u_j)} (1-\mu)^{D - \mathtt{d_H}(u_i, u_j)} \quad (2)$$

with $\mathtt{d_H}(u_i, u_j)$ the Hamming distance between features u_i and u_j.

Last, with a numerical value C_i tied to the confusion risk for each keypoint i we sort them from less to more confusing. But better than simply choosing a keypoints subset of arbitrary size, the authors propose a way of linking a confusion probability p value (from 0 to 1) to a C_{th} value. This way, the subset of kept keypoints varies accordingly to the confusion risk attached to the image. This gives the rather straightforward algorithm 1 where v should be replaced by σ for floatting-point feature vectors (average variance of said vectors) and by μ for the binary ones (bit-flip probability).

Data: I : image input
Data: p : accepted confusion probability
Data: v : Kernel density estimator window parameter
Data: $C_{th} \leftarrow$ findThreshold(p, v)
Result: χ : keypoint subset returned
$K \leftarrow$ keypoint set detected on I
$U \leftarrow$ associated feature vectors
for $u_i \in U$ **do**
  $c_i \leftarrow$ KDE(u_i, U, v)
end
for $k_i \in K$ **do**
  if $c_i < C_{th}$ **then**
    Add k_i to χ
  end
end
return χ

Algorithm 1: CORE algorithm.

An example of the CORE algorithm is illustrated with Figure 1. We can observe interesting trends regarding the keypoints localization: those kept tend to be located on special spots such as images, titles, subtitles, etc. whereas discarded keypoints are mostly inside text blocks which contain the most repetitive patterns, visually speaking.

III. Improving Document Image Matching With Filtering

Image matching using local descriptors follows a simple pipeline, where stable keypoints or regions are first detected before their local image neighborhood in the image is summarized into a single descriptor. For a given image, would it be the model image or a video frame, we can detect and compute a set of keypoints and their associated descriptors. In the image matching scheme we consider in this paper, we restrict ourselves to a one-to-one matching between a model image and each of the frames of a video recording. Locating the precise position of the instance of the model image within each frame of the video is made possible by first matching each local descriptor extracted from the video frame against the descriptors previously extracted from the model image and stored in an indexing structure for fast nearest neighbor search. In order to avoid ambiguous matches, a ratio-test strategy [1] is used. Finally, given a set of putative matches, a final RANSAC [22] step estimates the perspective transform between the recognized model document and its instance appearing in the scene, discarding outliers.

This image matching process progressively discards more and more information from each of the original images to finally select only a consistent subset of inliers which support the estimated homography. Filtering the relevant parts of the images is a costly process which can be improved by discarding bad candidates as early as possible, saving both memory and computation time. Filtering elements is particularly interesting before the indexation of the model image, as the resulting keypoints and descriptors set will be used at each iteration of the subsequent process. Early filtering of elements from video frames is harder in a real-time environment when filtering as to be added to the regular processing time: only very simple techniques can be applied here.

The first possible filtering happens at the core of the keypoint detection methods: their goal is to select points or regions which will exhibit the best invariance to illumination changes, blur, perspective and other distortions. Given the limited amount of context those methods can use, they can only rank the keypoints according to some basic response heuristic. Such filtering is suitable for limiting the amount of keypoints considered at run-time in a video frame, but for the model image such decision can be postponed until more context is available.

A second filtering stage can happen once descriptors are computed, taking into account the distribution of descriptors in the feature space. The CORE method [19], [20] works at this stage by discarding local descriptors based on their probability of confusion during the matching stage.

A third filtering stage can be performed one step further in the process, during descriptor matching and perspective transform estimation. This requires the use of training frames for each model to optimize. Filtering is implemented at this stage by discarding local descriptors which are rarely used as supports for estimating the perspective transform. This makes use of much broader context but comes at the cost of obtaining training data. It is possible to simulate the transformation of the image [23], or to use training data [18]. The contribution of each descriptor can be weighted by the quality of the segmentation found, but in practice this adds little information over the redundancy of training examples.

IV. Evaluation Protocol

In this paper we extend the evaluation protocol presented in [18]. We introduce results for the recent CORE method, evaluate the performance on one more keypoint detection and description scheme: SURF-BRISK, and we report the results based on the relative reduction of the keypoints set.

A. Methods under evaluation

We evaluate the following three methods.

1) Baseline: Our baseline approach is a filtering based on the keypoints' responses. Each algorithm has its own method. For example, as said earlier, ORB orders FAST keypoints with a Harris corner measure and SIFT relies on a contrast analysis. This allows us to compute reduced keypoint subsets with fixed size. We can then evaluate the matching quality by reducing progressively the keypoints subsets sizes, from 100% size to 10% with 10% decrements for each step.

2) Histograms: The histograms optimization was introduced in [18]. This filter relies on an off-line training step based on how many times a keypoint was successfully used by the RANSAC algorithm to estimate the homography between the model image and each video frame of a training set. This method requires a training video for each model image but can evaluate naturally the stability of a keypoint and the discriminative power of its descriptor. Just as the baseline approach, it allows us to select a proportion of keypoints from a given keypoint set, from 100% to 10% with 10% decrements for each step.

3) CORE: The CORE algorithm is a recent contribution with a filtering based on a probability approach relying on the feature vectors analysis. We test it for the first time in a document image matching application. Contrary to the histograms optimization it does not require a training step. However, it was designed to return varying keypoints subsets sizes depending on the inner confusion within the image, not by computing fixed subsets size. We could order the keypoints by their C_i value but we prefer to stay true to the algorithm's philosophy. Thus, we vary the p parameter from 0.15 (15% confusion tolerated) to 0.005 (0.5%) in order to return reduced keypoints subsets.

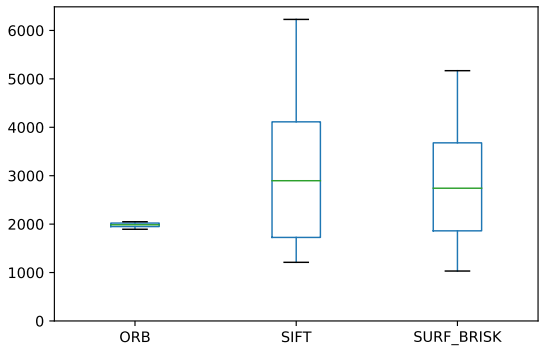

Fig. 2. Distribution of keypoints originally extracted from model images for each detection scheme. Lines indicate 5 and 95 percentiles, first and third quartiles, and median.

B. Detection and Description Schemes

We evaluate each method with the following three classical algorithms to test the efficiency of the keypoints selection methods. Figure 2 shows the distribution of keypoints originally extracted from model images for each detection scheme.

1) ORB: We set the ORB algorithm to return initial sets of 2000 keypoints per model image. Said set sizes are relatively faithful to what was asked since in practice we only observed a 10% variation in extreme cases.

2) SIFT: We employ regular parameters for the SIFT detector. We do not set restriction to set the number of keypoints for initial extraction: we noticed that the number of keypoints detected for each model was between 1000 and 6000.

3) SURF-BRISK: Following the evaluation in [24], we choose to couple the BRISK feature descriptor with a SURF keypoints detector since it is an interesting combination. Furthermore, the SURF detector returns numerous keypoints with document images (usually much more than SIFT). It is thus a good candidate for a filtering evaluation. No restriction to the number of keypoints to be detected was given to SURF but we noticed sizes between 1000 and 5000 keypoints for each model.

C. Dataset

The testing dataset is the SmartDOC database for document capture (challenge 1) [25], consisting of six different document types coming from public databases and five document images per class. An example of each of those six different document types is shown in Figure 3. Small video clips for each document in different backgrounds were recorded totaling near 25 000 frames with its corresponding ground-truth of the document position.

As the dataset ensures documents are always fully visible in each frame (i.e. not "zoomed at"), we had to reduce the size of the model images to match the maximum size documents could appear in frames. Having bigger images would result in a lot of keypoints detected at high resolution never being matched, and add a strong bias in favor of keypoint reduction as discarding high-resolution keypoints would always improve

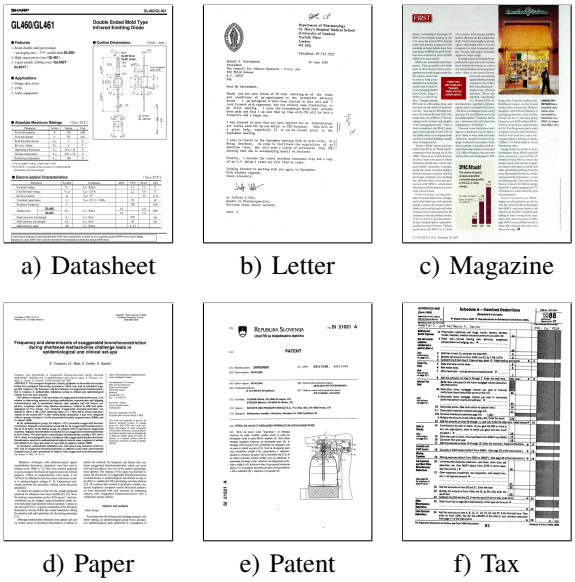

Fig. 3. Sample documents used in our dataset. a) Data-sheet from Ghega, b) letter from Tobacco800, c) magazine from PRIMA, d) paper from MARG, e) patent from Ghega and f) tax form from NIST.

results. Doing so prevents us from deliberately adding keypoints from useless levels in models.

D. Performance Evaluation

Performance evaluation is performed by averaging quality (segmentation accuracy) and speed (frame processing time) indicators computed at a frame level. Such indicators are averaged progressively to cope with normalization issues due to the variability of:

1) the number of keypoints which can be detected on each model image (see Figure 2);
2) the number of frames per video among documents and backgrounds.

Quality and speed indicators are computed against the reduction factor applied by each keypoint selection method to the original set of keypoint extracted by the baseline method. This reduction factor is normalized to $[0, 1]$ using the original size of the set of keypoints detected for each document model, and expressed in percentage. This is particularly useful to cope with the variability in the number of keypoints filtered by the CORE method for a given threshold. A small reduction factor indicates a small (strongly reduced) keypoint set, whereas a factor of 1 indicates the complete original set.

1) Segmentation Accuracy: In order to assess the ability of the different methods at providing accurate matchings, we will use the Jaccard index measure [26], as proposed in [18], [25] that measures the goodness of overlapping of the resulting S and ground-truth G quadrilaterals for a given frame f, after projecting the coordinates in the plane of the document (each pixel in the target referential covers the same physical surface of the document):

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

where $G \cap S$ and $G \cup S$ are the polygon intersection and union respectively.

Values are comprised between 0 (worst case) and 1 (best case). Results below 0.6 are not reliable for any use.

Good keypoints selection methods are expected to maintain or even augment the quality score when the reduction factor decreases. They must remain over the baseline to prove their interest.

2) Frame Processing Time: As the processing speed is subject to the influence of the time required to find a suitable homography given all candidate matches, we measured the total processing time for each frame of the dataset, for all the keypoint selection variants. Processing time is measured using the standard Python profiler module "cProfile", and all the computations were performed on similar hardware. Times reported include the complete time required to process a frame using a Python implementation and exclude any marginal computation like environment setup, training times, model loading, etc.

Good keypoints selection methods are expected to reduce the processing time when reducing a keypoint set since it implies a faster RANSAC convergence.

V. Evaluation Results

We analyze here the results obtained by the three keypoint reduction methods for three detection and description schemes. Figure 4 summarizes those results.

We can see that the original hypothesis of this work, that processing speed actually improves when the size of the model is reduced, hold in all cases (despite some spikes due to imprecision in the measure) and thus legitimates using the keypoint reduction methods benchmarked here.

For each approach, past a certain level of reduction the matching quality starts to drop significantly. However, the corresponding threshold is not the same with every approach.

For the histograms (HIST) method [18], the important amount of context used to select keypoints is a clear advantage regarding the reliability of the results: it enables very strong reductions of the original set of keypoints while keeping the segmentation accuracy stable, and even improving it for ORB and SURF-BRISK. Processing time is also steadily decreasing along the size of the model keypoint set, in a way similar to the baseline, showing no loss of processing speed.

For the CORE method [19], [20], Figure 2.a) clearly indicates that this method is not reliable after a value of 0.15 (15%) below which the quality measure drops below the baseline. However, results with SIFT and SURF-BRISK descriptors exhibit a very interesting quality and speed performance for moderate reductions of the original keypoint set: segmentation quality is on par with the histograms method while being unsupervised. Even if the quality drops suddenly after a certain stage, it remains better than the baseline, making CORE a simple and reliable solution for keypoint set reduction for those local descriptors. Processing time, finally, is also steadily decreasing along the size of the model keypoint set, confirming the usefulness of such approach. Regarding the rather poor results with ORB features, an explanation could come from the way its detector (*Oriented FAST*) selects the keypoints. By combining the FAST algorithm with a Harris corner measure, it brings a strong emphasis on cornerness which isn't well suited for document images whereas SURF and BRISK rely more on scale-space analysis. Therefore, Oriented FAST sets of points might be more random here which penalizes the CORE algorithm with its probability approach, failing to select a relevant set of keypoints by analyzing the features distribution.

VI. Conclusions

We introduce the use of the recent CORE filtering method [19], [20] for improving the processing of document images. We benchmarked this method against previous work using a reproducible protocol supported by a public dataset and continued the first experiments of Chazalon et al. [18] by adding more evaluations. The CORE method adds an extra stage to the keypoint filtering pipeline, making use of statistical properties of the descriptors extracted. This method exhibits very interesting properties making it suitable for improving SIFT or SURF-BRISK sets of local descriptors for which it can be as efficient as a supervised method, and also ORB to a lesser extent. The actual computations for keypoint filtering is also very fast. The histogram method remains a suitable choice because of its reliability both in terms of accuracy (which it can even improve while reducing the original set) and resulting processing speed. This stability comes, however, at the cost of manually creating or generating training samples for each model to optimize, and an extra training phase. Both methods can, of course, be combined.

Acknowledgment

This work was supported by the French region Provence-Alpes-Côte d'Azur, by the Spanish project TIN2014-52072-P, by the CERCA Programme / Generalitat de Catalunya, and by the MOBIDEM project, part of the "Systematic Paris-Region" and "Images & Network" Clusters, funded by the French Government and its economic development agencies. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

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

[2] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[3] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of the IAPR International Conference on Pattern Recognition*, 1994, pp. 582–585.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proceedings of the European Conference on Computer Vision*, 2010, pp. 778–792.

[5] 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.

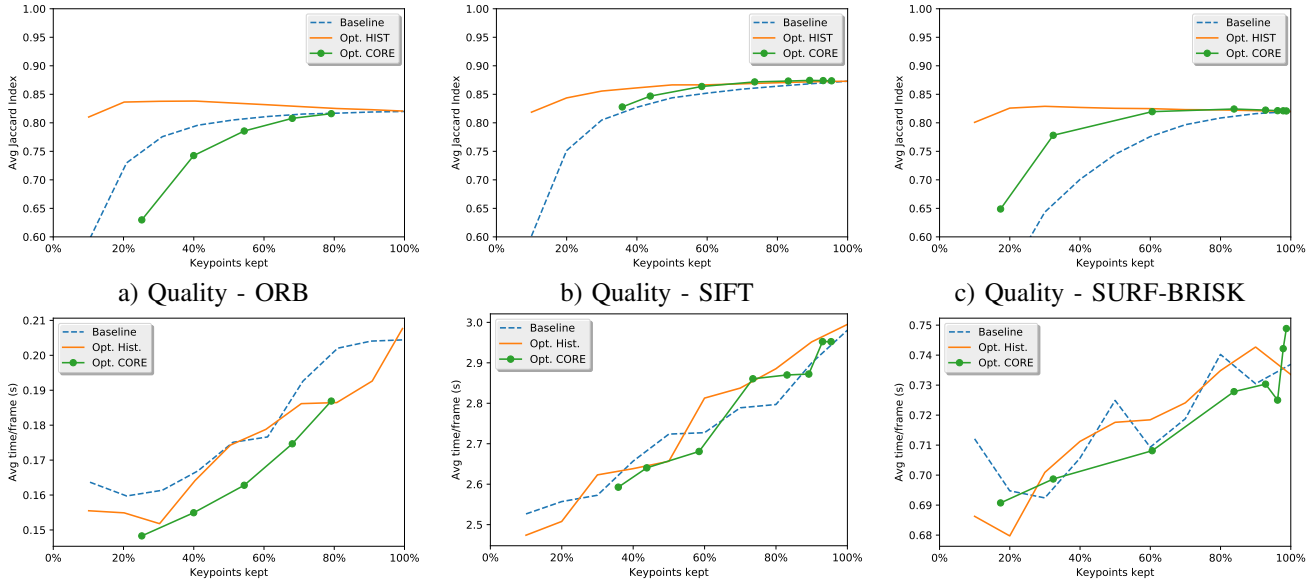

Fig. 4. Quality (top) and speed (bottom) results for each keypoint filtering method (baseline, histograms [18] and CORE [19], [20]), for three classical local descriptors: ORB (a, d), SIFT (b, e) and SURF keypoints with BRISK descriptors (c, f). Both methods under test enable to accelerate processing speed when reducing the keypoint set. The histograms methods achieves the best quality and the CORE method constantly provides better results than the baseline, except for ORB, without requiring any training. Speed curves carry some imprecision due to the non-deterministic of this measure. This explains why curves does not exactly join for $x = 100\%$.

[6] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 2548–2555.

[7] R. Ortiz, "FREAK: Fast retina keypoint," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 510–517.

[8] G. Levi and T. Hassner, "LATCH: learned arrangements of three patch codes," in *Proceedings of the Winter Conference on Applications of Computer Vision*, 2016, pp. 1–9.

[9] V. Balntas, L. Tang, and K. Mikolajczyk, "BOLD - binary online learned descriptor for efficient image matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2367–2375.

[10] M. C. T. Trzcinski and V. Lepetit, "Learning image descriptors with boosting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 597–610, March 2015.

[11] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 228–242.

[12] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European Conference on Computer Vision*, 2006, pp. 430–443.

[13] M. Block, M. R. Ortegn, A. Seibert, J. Kretzschmar, and R. Rojas, "SITT-a simple robust scaleinvariant text feature detector for document mosaicing," Free University of Berlin, Tech. Rep. B-07-02, 2007.

[14] 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 *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, 2013, pp. 230–234.

[15] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, September 2004.

[16] T. Nakai, K. Kise, and M. Iwamura, "Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval," in *Proceedings of the International Workshop on Document Analysis Systems*, 2006, pp. 541–552.

[17] M. Iwamura, T. Nakai, and K. Kise, "Improvement of retrieval speed and required amount of memory for geometric hashing by combining local invariants," in *Proceedings of the British Machine Vision Conference*, 2007, pp. 1010–1019.

[18] J. Chazalon, M. Rusiñol, and J.-M. Ogier, "Improving document matching performance by local descriptor filtering," in *Proceedings of the International Workshop on Camera Based Document Image Analysis*, 2015, pp. 1216–1220.

[19] E. Royer, T. Lelore, and F. Bouchara, "CORE: A COnfusion REduction algorithm for keypoints filtering," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2015, pp. 561–568.

[20] ——, "COnfusion REduction (CORE) algorithm for local descriptors, floating-point and binary cases," *Computer Vision and Image Understanding*, vol. 158, pp. 115 – 125, 2017.

[21] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. pp. 1065–1076, 1962.

[22] 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.

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

[24] M. Rusiñol, J. Chazalon, J.-M. Ogier, and J. Lladós, "A comparative study of local detectors and descriptors for mobile document classification," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2015, pp. 596–600.

[25] J. Burie, J. Chazalon, M. Coustaty, S. Eskenazi, M. Luqman, M. Mehri, N. Nayef, J. Ogier, S. Prum, and M. Rusiñol, "ICDAR2015 competition on smartphone document capture and OCR (SmartDoc)," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2015, pp. 1161–1165.

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