A Segmentation-free Handwritten Word Spotting Approach by Relaxed Feature Matching

Anders Hast
Department of Information Technology
Uppsala University
Uppsala, Sweden
Email: anders.hast@it.uu.se

Alicia Fornés
Computer Vision Center - Computer Science Department
Universitat Autònoma de Barcelona
Bellaterra, Spain
Email: afornes@cvc.uab.es

***Abstract*—The automatic recognition of historical handwritten documents is still considered a challenging task. For this reason, word spotting emerges as a good alternative for making the information contained in these documents available to the user. Word spotting is defined as the task of retrieving all instances of the query word in a document collection, becoming a useful tool for information retrieval. In this paper we propose a segmentation-free word spotting approach able to deal with large document collections. Our method is inspired on feature matching algorithms that have been applied to image matching and retrieval. Since handwritten words have different shape, there is no exact transformation to be obtained. However, the sufficient degree of relaxation is achieved by using a Fourier based descriptor and an alternative approach to RANSAC called PUMA. The proposed approach is evaluated on historical marriage records, achieving promising results.**

I. Introduction

In the last years, there has been a growing interest in the digitization of historical document collections in archives, libraries and museums towards the preservation of cultural heritage. Once this first goal has been achieved, the next step consists in the recognition of these documents, in order to make the information that these documents contain available to scholars and citizen in general. In the case of printed books, the automatic transcription of these documents could be feasible, and therefore the user could directly search for information in the transcribed textual data. However, the automatic recognition of historical manuscripts is still an open problem, mainly because of paper degradation, different handwriting styles, old languages, etc. Consequently, the user is forced to carefully read each one of these manuscripts in order to find the desired information.

In this scenario, handwritten word spotting [1] arises as a good alternative for making the information available to the user. Word spotting can be considered as a special case of image retrieval, whose goal is to find all instances of the query word in the document collection. In the literature, most existing techniques are based on Dynamic Time Warping [2], Hidden Markov Models [3], [4], Recurrent Neural Networks [5] and Support Vector Machines (SVM) [6]. However, the aforementioned techniques need to segment the documents into text lines or even into words. Therefore, the performance of these techniques highly depend on the accuracy of the line or word segmentation algorithms.

For this reason, some researchers have focused on the proposal of completely segmentation-free approaches, and even some competitions in international conferences have been organized [7] to foster the development of such approaches. For example, Leydier et al. [8] propose a segmentation-free method based on zones of interest. First, for each one of the detected zones of interest (high-curvature locations and extrema points in the strokes), the method computes the gradient angles, and constructs a graph of zones of interest. The final matching is performed by searching through the tree. Although there is a mechanism for pruning, this method is quite slow. Howe [9] proposes a segmentation-free approach that creates a flexible ink-ball model, allowing for gaussian random-walk deformation of the ink trace. Thus, the query model is deformed in order to match the candidate regions and vice-versa. This method has also demonstrated good performance but it is also rather slow.

Some other approaches divide the document page into cells, compute a feature vector for each cell, and then search the query word into the document using a sliding window. This is the case of the method proposed by Rusiñol et al. [10], which consists in a patch-based framework where patches are represented by a bag-of-visual-words model powered by SIFT descriptors. Similary, Almazán et al. [11] propose a method based on the Exemplar SVM framework, where the words are described using Histogram of Oriented Gradients (HOG) and the query is found using a sliding window. Both methods are quite fast and with a good performance.

A different approach has been proposed by Kovalchuk et al. [12]. The idea is to find multiple overlapping candidate targets by analyzing the connected components, joining them if necessary. Then, for each cell in the candidate region, HOG and LBP (local binary pattern) descriptors are computed. The matching is performed using the cosine similarity operator with maximum pooling over random groups. The retrieval is performed using k-Nearest Neighbor. This method is very fast, although its performance is a bit lower than the above commented sliding-window based methods.

Some structural representations have also been proposed. For instance, Wang et al. [13] propose a coarse-to-fine segmentation-free word spotting method based on graph representations. Since structural approaches use to be slower than statistical ones, this method speeds up the matching by first locating the zones of interest from the image by using a graph embedding representation. Then, these candidate regions are compared using the graph edit distance based on the Dynamic Time Warping alignment.

From the above approaches, we can summarize that the methods with a better trade-off concerning performance and speed are based on gradients (e.g. HOG, SIFT), patches and sliding windows schemes. In this paper, however, we explore the use of feature descriptors from the Fourier domain and the use of key-points instead of patches. Moreover, we propose a fast deterministic and more relaxed part-based matching method to be used instead of RANSAC. Our main contribution is a segmentation-free word spotting method that describes the words in the Fourier domain, uses key-points for detection, and proposes a putative match analysis with relaxed matching.

The rest of the paper is organized as follows. Next, our word spotting approach is described, including the key-points detection, the feature computation and the matching. In Section III, the experimental results are shown and analyzed. Finally, Section IV draws the conclusions and proposes future work.

II. Methodology

The overview of the method is the following. First, the input images are binarized using Otsu, and then smoothed with a Gaussian in order to find more key points. Then, four different kind of key points are detected in the words, which basically detect lines, corners and blobs. Afterwards, the matching is performed by an improved version of RANSAC, called the Putative Match Analysis, which is able to allow a more relaxed matching among the words.

A. Key-point detection

Key point feature matching is generally done using one type of key point detector only and just to give two examples: SIFT [14] uses difference of Gaussians (DoG) and SURF [15] uses the determinant of Hessian (DoH). Both these are finding blob type of features in the images. Others find corners like the Harris detector [16] and the extended structure tensor approach [17]. The trace of matrices such as the structure tensor, the Hessian and the spinor tensor [18] can be used to find points along lines and edges.

The method proposed in this paper uses a combination of different key point detectors in order to capture the different features in hand written text, which consists of both lines, corners and blobs. Figure 1 shows an example of a binarized and smoothed query word with four different types of detected key points. Blue $*$ is a corner detector [17], green $+$ is the result of using the square of the DoH, which will thus capture both dark and bright blobs, cyan $+$ finds points along lines and red Δ is the result of a special mix of a blob and line detector[1]. One could have chosen many different combinations

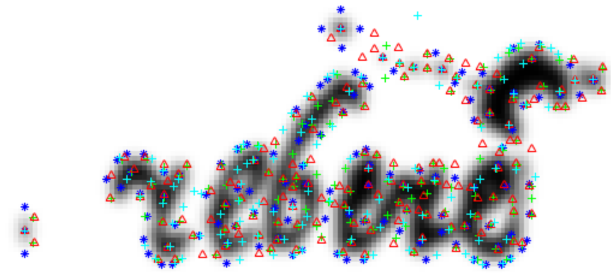

Fig. 1. An example of a query word (rebere) with four different types of key points depicted. Notice how the different key points in general are found in different positions. Blue $*$ is a corner detector, green $+$ finds dark and bright blobs, cyan $+$ finds lines and red Δ is the result of a special mix of a blob and line detector.

of any number of key point detectors [19]. However, from our experiments, we have seen that these four are the most suitable for detecting the characteristic points in handwritten words.

B. Fourier based feature descriptors

Many feature descriptors use the local image content in square areas around each key point to form a feature vector. Both scale and rotation invariance can be obtained in different ways [20]. Recently, a Fourier based feature vector has been proposed [21], [22]. This descriptor is illumination and rotation invariant and it is also invariant to scale to a certain degree. A local disc neighbourhood (with radius 16) is sampled into a 16×16 matrix, from which the amplitude of the Fast Fourier Transform (FFT) is computed. The invariance is important as these qualities will vary over the different words (e.g. the slant, size and shape of the loop of the letter *l* can vary between the query and the retrieved words). Nevertheless, the main advantage with the Fourier based descriptor is that high frequency changes, such as residuals from neighbouring words, have very little impact on the feature vector because it is based on the low frequency content in the local neighbourhood. Another advantage is that it is not sensitive to small differences in form and shape as long as they are more or less the same, i.e. the low frequencies are sufficiently similar.

C. Matching

The matching takes advantage of the fact that, since there are four different key point detectors, the nearest neighbour search (NNS) is performed within these four subgroups. The key points are computed for the whole document as well as for the query word. The sliding window used is four times larger than the query word and steps forward with the size of the query word. This ensures that the word being searched will be totally located inside the query window at least once, unless that word is substantially bigger than the query.

Nevertheless, the key point matching will be able to capture words even if it is partially outside the sliding window as shown in Figure 2. The total number of matching points from all four separate NNS's are shown as green dots. The green square is the predicted extent of the word, using the size of the query word as shown in Figure 1, which can be easily

[1] These are referred to as $Harris\ of\ Hessian$ and $Hessian^2/Harris\ of\ Hessian$ respectively in [18]

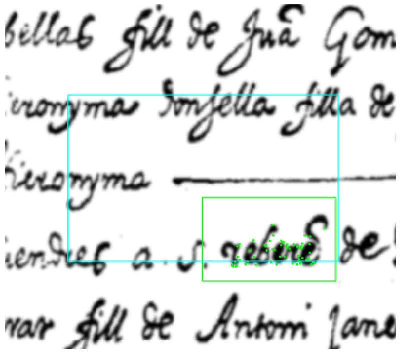

Fig. 2. The searched word in the sliding window is found and the extent of the word, depicted as a green box, which is computed using the matching key points, depicted as green dots.

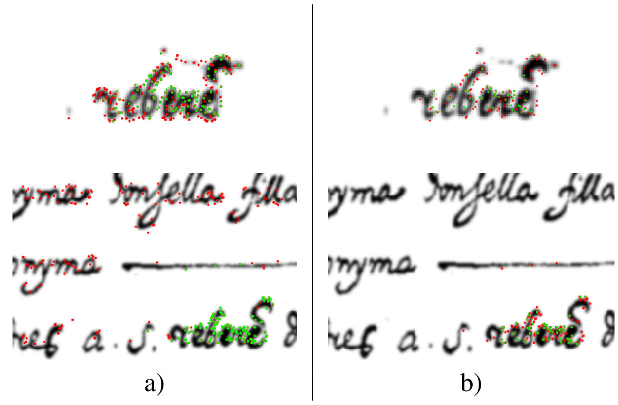

Fig. 3. The result of matching the query word (top) with the content in the sliding window (bottom). a) The preconditioner removes the most obvious false positives (red) and tentative inliers (green) are processed by PUMA. b) The true matches (in green) and the false positives (in red) after PUMA .

Fig. 4. Half of the query word is matched with half of the word found. The correspondences in the right side are depicted in green and the left side in yellow.

computed from the extent of the matching points in each word respectively. When a word is found, the matching points will be removed from the set of points so that the same word will not be found again in case the sliding window covers it again in the subsequent steps.

There are three different steps that are performed in order to obtain the result in Figure 2 and they will be explained in detail. First of all, the resulting correspondences from the matching between the query and the sliding window needs to be further processed as there are usually many false positives (outliers). RANSAC [23] is often used for this purpose and a transformation between the words (images) is obtained. Since the same word in different places in the documents can differ, exhibiting small variations in length and height, it is important that the transformation between the query and the word in the sliding window is relaxed. However, the transformation used in RANSAC cannot sufficiently capture the frequent occurring variations within handwritten words in a good way and as a result will consider some true inliers being outliers. This is probably one of the main reasons why regular patches has been used with SIFT instead of the key point approach we propose. In order to solve this problem we propose to use an alternative approach. First, a deterministic preconditioner [24], [25] removes most false matches, as shown in Figure 3 a). Next, a method called Putative Match Analysis (PUMA) [26] is used to process the tentative inliers b), which is able to be rather tolerant to variations. Another advantage of PUMA is that it is totally deterministic and will give the same result in each run, which is generally not the case with RANSAC.

The preconditioner is used just to speed up the process and works by clustering the correspondences in 2D space as positional vectors. In other words, correspondences that are inliers will have the same length and direction from one image to the other and will form a cluster in 2D while the outliers will be spread over the space. However, since corresponding words can differ in form, the cluster will be a bit scattered and the threshold must be set loosely. Nevertheless, most obvious outliers are removed in this step and the remaining tentative inliers are processed by PUMA, which aims at finding a consistent but more loosely defined Euclidean transformation within the set of correspondences. PUMA is also a clustering method, but works in a high dimensional space where for each correspondence, a cluster is formed where the other correspondences have a relative angle and length. The worst outliers are removed one by one until the true inliers are the only remaining in the clusters. Hence, both the preconditioner and PUMA do not rely on a single transformation as RANSAC does. Therefore it is less tolerant, but rather on the mutual consensus of the inliers, i,e. the formation clusters. Hence, both methods allow different parts of the word to have similar, but still a bit different transformations, which is necessary when a method based on key points is used instead of regularly spaced patches.

In some cases, a word shares many letters with some other different words, and consequently, the risk of a false positive raises. For instance, the word 'donsella' has many letters in common with both 'fill' and 'filla'. To avoid this problem, a second step was introduced where both the query word and the word in the green box is divided into two parts, so that each part is matched separately. The result is shown in Figure 4. Nevertheless, this second step was performed only if the first step was successfully passed, i.e. if the number of matches divided by the total number of points in the query word was over a certain threshold τ.

Nevertheless, in case of long words, the same problem may occur again within the parts. We noted that the reliability of the algorithm was increased by introducing a third and final step that was only entered if the second step was successfully

Fig. 5. The corresponding points after matching within the green box only.

passed. This time the matching is done on the whole word within green box. The final result is shown in Figure 5.

Each step produces its own ranked list according to their confidence. So, the three lists are concatenated to form the final list of retrieved words.

D. Summary of the method and its benefits

In summary, the method consists of three different steps. First a sliding window technique is used to match the feature vectors in the query word with feature vectors in the sliding window. A preconditioner is used together with PUMA to find inliers and also, for allowing a rather relaxed transformation between the words. Another advantage is that they are both deterministic and the randomness of RANSAC is avoided. Second, if the number of inliers is high enough, then it is considered a candidate and the size of the word is estimated and extracted from the document so that neighboring text will not affect the matching. Finally, the word is split into two equal size parts and the matching process is repeated for each part. If both parts contain enough matching points, then the process is repeated for the whole word as a final check.

The Fourier based descriptor with its log polar sampling scheme was used since it does not contain the time consuming framework for obtaining rotation and scale invariance that SIFT makes use of. SIFT needs to compute the main orientation using gradient histograms, while instead rotation invariance is obtained inherently in the sampling process. The log in the log polar sampling also assures enough scale invariance for our purposes and there is no need to find the maximum over scales as SIFT does. However, more importantly, only the low frequencies are used to construct the feature vector and high frequency differences will not have an impact on the result, as would be the case for SIFT. In other words, only the main characteristics of the strokes are considered and not small individual variations.

By using several key point detectors instead of just one blob detector that SIFT makes use of, it is possible to capture different characteristics of the strokes, i.e. both corners, blobs and edges. Another advantage is that the total amount of points to work with is also increased and this will assure a better matching. The exhaustive search in the matching process is $O(n^2)$, but this does not mean that the comparisons are increased quadratically since the matching is done individually for each type of key point and it is therefore divided into four smaller parts. As an example, if $n = 4k$ where k is the size of each part, the exhaustive search for using a single key point detector of size n would be $n^2 = (4k)^2 = 16(k^2)$, while using four different key points turns into four times faster: $4(k^2)$.

III. Experimental results

The proposed method has been evaluated using a subset of the Barcelona Historical Handwritten Marriages (BH2M) database [27]. The whole collection comprises 244 books with information on 550,000 marriages held between the 15th-19th century. For the experiments, we have selected 50 pages from the 17th century, which have been written by one single author. The ground-truth contains the bounding-boxes for each word plus its corresponding transcription.

For comparison purposes, we show the performance of several methods in the literature that have also been tested using the same set of images. The first method [28] uses an indexation scheme for selecting the candidate regions, a statistical descriptor based on gradients (HOG) and the SVM classifier. The second method [29] uses a pseudo-structural descriptor based on LOCI features. The third method [13] corresponds to a structural descriptor based on graph-representation and matching.

For performance evaluation, we combine the retrieved regions of all the pages and rerank them according to their score. A region is classified as positive if it overlaps more than 50% the area of the annotated bounding box in the ground truth, and negative otherwise. We compute the Precision-Recall curve. Precision is the number of relevant (true positive) objects found divided by the number of all retrieved objects, and Recall is the number of relevant objects found divided by the number of all relevant objects in the set.

$$Precision = \frac{|retrieved \cap relevant|}{|retrieved|} \tag{1}$$

$$Recall = \frac{|retrieved \cap relevant|}{|relevant|} \tag{2}$$

Since one final single value is easier for comparing different approaches, we provide the mean Average Precision (mAP), which corresponds to the area below the Precision-Recall curve, and is computed using the following equation:

$$\mathrm{mAP} = \frac{\sum_{n=1}^{|\mathrm{retrieved}|} P@n \times r(n)}{|\mathrm{relevant}|} \tag{3}$$

where $P@n$ is the precision at the n top-most returned results, and $r(n)$ is a binary function indicating whether the n-th item in the returned ranked list is a relevant object (true positive).

Table I shows the Average Precision for each one of the query classes and the mean Average Precision (mAP). From the experimental results, one can see that our approach significantly improves the other approaches (the mAP of 77% is almost 20 points above the mAP of other approaches).

As expected, the performance is higher when searching longer words (e.g. 'habitant' and 'rebere' have a mAP over 90%) because the probability of finding the query word as part of other similar words is much lower. It is indeed a very typical problem of segmentation-free approaches: if the method does not know when each word starts and ends, then the query word

TABLE I
EXPERIMENTAL RESULTS. AVERAGE PRECISION (IN %) FOR EACH ONE OF THE QUERY CLASSES, COMPUTED OVER 50 PAGES, AND THE FINAL MEAN AVERAGE PRECISION (MAP).

Query class	Statistical (HOG) [28]	Pseudo-structural (LOCI) [29]	Structural (Graphs) [13]	**Proposed** (Fourier)
barna	40.34	8.34	55.56	65.98
fill	42.09	73.31	56.66	60.80
filla	57.39	44.34	48.21	74.23
habitant	74.10	10.87	55.83	96.74
pages	76.55	67.37	70.20	84.07
viuda	69.47	39.25	49.67	77.24
viudo	33.13	34.89	40.92	64.35
rebere	74.78	57.50	75.67	94.10
mAP	**58.48**	**41.98**	**56.59**	**77.18**

can be found as part of a longer word. For instance, Figure 6(a) shows that 'filla' (daughter in English) is retrieved when searching the word 'fill' (son in English). Similarly, a query word can be very similar to the ending plus the beginning of the next word (see Figure 6(b)). Consequently, in such cases, the mAP is usually lower than the ones obtained through segmentation-based approaches.

One drawback of our key point matching is that some parts of the retrieved words may be very similar to some small parts of the query word, and hence give many inliers. This is the case of very similar words, such as 'viuda' and 'viudo' (widow and widower in English). In order to minimize this issue, the words are divided into two parts, and then we check whether each corresponding part of the word matches as well. We believe that the Average Precision for query words that share many letters can be increased by dividing the word into more parts than just two, whenever the word is long enough.

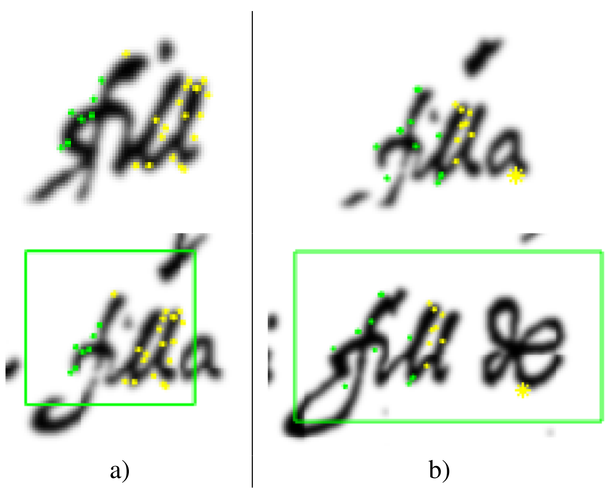

Fig. 6. Typical failure cases with query word in the top and the retrieved word below. a) The word 'fill' is found inside the longer word 'filla'. b) The word 'filla' is found by taking parts of two consecutive words ('fill'+'de'). Note the point denoted by a yellow * that is mistakenly taken for a part of the ending 'a' in 'filla'.

In summary, these results demonstrate that a descriptor in the Fourier domain with a relaxed putative matching is able to characterize and discriminate handwritten words. However, there are several ad-hoc parameters set in this approach. A tentative word is discarded if the aspect ratio does not correspond to the query word or when the number of inliers is two low, indicating a bad match. These parameters have been manually set by trying to find the best threshold while running some small experiments in several samples. Obviously, there is room for machine learning approaches to find the best parameters for different types of documents.

IV. CONCLUSION

In this paper we have presented a word spotting method based on key-points, a Fourier-based descriptor, and the Putative Match Analysis. The approach is segmentation-free, in other words, there is no need to segment the document into lines nor words. The experimental results are encouraging, demonstrating that the proposed methodology is suitable for dealing with the inherent difficulties in handwriting.

For future work we plan to divide words into more parts in order to avoid confusions between similar words. Moreover, we think that machine learning can help to set the parameters that are now manually optimised for the test data. Finally, we plan to test our method with multi-writer document collections in order to analyse its robustness in front of different handwriting styles.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish project TIN2012-37475-C02-02, the European project ERC-2010-AdG-20100407-269796, the Swedish Research Council (Vetenskapsradet, Dnr 2012-5743) and the Riksbankens jubileumsfond (NHS14-2068:1). The authors acknowledge the support of the Ramon y Cajal Fellowship RYC-2014-16831.

REFERENCES

[1] J. Llados, M. Rusinol, A. Fornes, D. Fernandez, and A. Dutta, "On the influence of word representations for handwritten word spotting in historical documents," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 05, 2012.

[2] T. Rath and R. Manmatha, "Word spotting for historical documents," *IJDAR*, pp. 139–152, 2007.

[3] J. A. Rodríguez-Serrano and F. Perronnin, "Handwritten Word-Spotting using Hidden Markov Models and Universal Vocabularies," *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, 2009.

[4] ——, "A model-based sequence similarity with application to handwritten word spotting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2108–2120, 2012.

[5] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 211–224, 2012.

[6] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[7] I. Pratikakis, K. Zagoris, B. Gatos, G. Louloudis, and N. Stamatopoulos, "Icfhr 2014 competition on handwritten keyword spotting (h-kws 2014)," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Sept 2014, pp. 814–819.

[8] Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz, "Towards an omnilingual word retrieval system for ancient manuscripts," *Pattern Recognition*, vol. 42, no. 9, pp. 2089–2105, 2009.

[9] N. R. Howe, “Part-structured inkball models for one-shot handwritten word spotting,” in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 582–586.

[10] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, “Efficient segmentation-free keyword spotting in historical document collections,” *Pattern Recognition*, vol. 48, no. 2, pp. 545–555, 2015.

[11] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, “Segmentation-free word spotting with exemplar svms,” *Pattern Recognition*, vol. 47, no. 12, pp. 3967–3978, 2014.

[12] A. Kovalchuk, L. Wolf, and N. Dershowitz, “A simple and fast word spotting method,” in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 3–8.

[13] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Lladós, and A. Fornés, “A coarse-to-fine word spotting approach for historical handwritten documents based on graph embedding and graph edit distance,” in *International Conference on Pattern Recognition*, 2014, pp. 3074–3079.

[14] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014

[16] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proceedings of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[17] A. Hast, “Interest point detection based on the extended structure tensor with a scale space parameter,” in *VISAPP*, 2015, pp. 1–8, poster with Paper.

[18] A. Hast and A. Marchetti, “Invariant interest point detection based on variations of the spinor tensor,” in *WSCG 2014*, 2014, pp. 49–56, short Paper.

[19] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: A survey,” *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2007.

[20] S. Gauglitz, T. Höllerer, and M. Turk, “Evaluation of interest point detectors and feature descriptors for visual tracking,” *Int. J. Comput. Vision*, vol. 94, no. 3, pp. 335–360, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11263-011-0431-5

[21] A. Hast, “Robust and invariant phase based local feature matching,” in *ICPR 2014*, 2014, pp. 809–814, poster with Paper.

[22] A. Hast and A. Marchetti, “Rotation invariant feature matching - based on gaussian filtered log polar transform and phase correlation,” in *ISPA 2013*, 2013, pp. 100–105.

[23] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[24] A. Hast and A. Marchetti, “An efficient preconditioner and a modified ransac for fast and robust feature matching,” in *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2012)*, 2012, pp. 11–18, short Paper.

[25] A. Hast and G. Kylberg, “Clustering in 2d as a fast deterministic alternative to ransac,” in *The second workshop on Features and Structures (FEAST), colocated with ICML*, 2015, p. 1, poster with Paper.

[26] A. Hast and A. Marchetti, “Putative match analysis - a repeatable alternative to ransac for matching of aerial images,” in *International Conference on Computer Vision Theory and Applications*, 2012, pp. 1–4, poster with Paper.

[27] D. Fernández-Mota, J. Almazán, N. Cirera, A. Fornés, and J. Lladós, “Bh2m: The barcelona historical, handwritten marriages database,” in *International Conference on Pattern Recognition (ICPR)*. IEEE, 2014, pp. 256–261.

[28] J. Almazán, D. F. Mota, A. Fornés, J. Lladós, and E. Valveny, “A coarse-to-fine approach for handwritten word spotting in large scale historical documents collection.” in *ICFHR*, 2012, pp. 455–460.

[29] D. Fernández, J. Lladós, and A. Fornés, “Handwritten word spotting in old manuscript images using a pseudo-structural descriptor organized in a hash structure,” in *IbPRIA*, 2011, pp. 628–635.