Viral Transcript Alignment

Gil Sadeh, Lior Wolf
Tel Aviv University
Tel Aviv, Israel

Tal Hassner
The Open University
Raanana, Israel

Nachum Dershowitz
Tel Aviv University
Tel Aviv, Israel

Daniel Stökl Ben-Ezra
École Pratique des Hautes Études
Paris, France

***Abstract*—We present an end-to-end system for aligning transcript letters to their coordinates in a manuscript image. An intuitive GUI and an automatic line detection method enable the user to perform an exact alignment of parts of document pages. In order to bridge large regions in between annotation, and augment the manual effort, the system employs an optical-flow engine for directly matching at the pixel level the image of a line of a historical text with a synthetic image created from the transcript's matching line. Meanwhile, by accumulating aligned letters, and performing letter spotting, the system is able to bootstrap a rapid semi-automatic transcription of the remaining text. Thus, the amount of manual work is greatly diminished and the transcript alignment task becomes practical regardless of the corpus size.**

I. Introduction

While high-quality imaging is currently the most effective way to create digital copies of historical manuscripts, having a searchable and processable text is often equally important to scholars. Unfortunately, optical character recognition (OCR) of historical documents is notoriously difficult. On the other hand, it is a very common scenario for important texts to have been transcribed manually. This is the case for many of the most valuable collections recently digitized and made available online. Examples include the Dead Sea Scrolls, Greek papyri, Codex Sinaiticus, some of the Cairo Genizah documents, much of the Tibetan Buddhist Canon, Taiwanese deeds and court papers in the Taiwan History Digital Library, medieval Latin and English manuscripts, the Early English Laws collection, the George Washington Papers[1] , and many others.

We have designed an end-to-end interactive system which learns to align each letter of a transcript with the corresponding pixels of the letter in the digitized images of scanned manuscripts. After identifying the lines of text in the image, the system begins with a first approximate alignment using an optical-flow engine, as in [1]. The interface we propose allows the user to naturally correct any alignment errors. Thereafter, the system learns to match individual letters based on corrections introduced by the user. As the system learns to better recognize letters in the images, the amount of manual effort required decreases dramatically, from one line to the next. As a consequence, transcript alignment becomes easier and faster as the user progresses through the document.

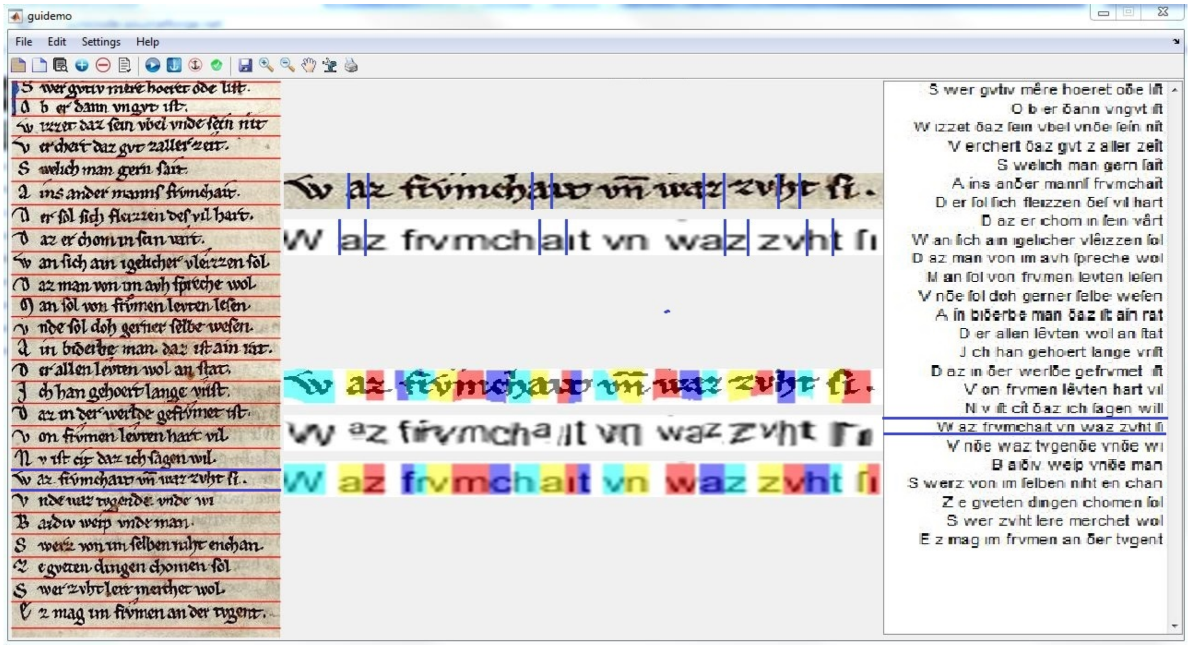
II. Previous Work

The problem of matching text with images of the text was raised in [2], but only a limited amount of research has been devoted to the issue. A straightforward approach to alignment is to perform OCR on the image and then find the best string-match between the OCR text and the transcription. A word-level recognizer is another possibility (see, e.g., [3], [4], [5]); it can also be used in training semiautomatic transcription [6], [7], [8]. But OCR for handwritten text, with which we are dealing here, is difficult. In [9], [10], [11] and others, the sequence of word images and the transcript are viewed as time series, and dynamic time warping (DTW) is used to align them. Hidden Markov models (HMM) have been used in [12], [13], [14], [15], [7], for example. Geometric models of characters and punctuation (including such features as character size and inter-character gaps) have recently been used to reduce segmentation errors (e.g., for Japanese in [16] and for Chinese in [17]). In [18] six visually significant elements that combine to signatures for the different characters of pre-Gothic Latin written by quills were selected for detection through gradient and connected components. The work in [5] uses a minimized cost function based on the relative length of words and the best combination of spaces. Alignment on the word level with a classification technique constrained by the number of the words is performed in [19]. The method in [7] also involves an application of the Viterbi algorithm. In contrast to the above-mentioned methods, we employ a rendered image of the text as the main representation of the transcript, and then use direct image-to-image matching techniques [1].

Our system incorporates a letter-spotting engine within it, and is based on techniques previously used for word spotting. Example of word spotting efforts include [20], [21], [22], [23]. Two approaches are possible in searching for occurrences of words in documents: one can first segment the text into words [24] and then compare each target word with the query, or one can search for a match to the query using a sliding window of some sort. An example of word spotting among segmented images is [25].

In our scenario, since letter segmentation is even harder than word segmentation, we cannot assume prior segmentation. Among the works that do not require segmentation are [26], [27], [28]. An in-between approach is to work with multiple overlapping target regions, as in [29]. Our letter-spotting engine also employs multiple overlapping regions and is based on a previous word spotting work [30]. The architecture is a hierarchical one and is inspired by the work of Liao et al. [31] in the domain of face recognition.

[1] Available online, respectively: http://www.deadseascrolls.org.il, http://www.papyrology.ox.ac.uk/Ancient_Lives, http://codexsinaiticus.org, http://www.genizah.org, http://www.tbrc.org, http://idp.bl.uk, http://thdl.ntu.edu.tw,http://scriptorium.english.cam.ac.uk/manuscripts, http://www.earlyenglishlaws.ac.uk, http://rotunda.upress.virginia.edu/founders/GEWN.html.



S wer gvtiv mêre hoeret oðe liſt
O b er ðann vngvt iſt
W izzet ðaz ſein vbel vnðe ſeín niſt
V erchert ðaz gvt z aller zeit
S welch man gern ſait
A ins anðer mannſ frvmchait
D er ſol ſich fleizzen ðeſ vil hart
D az er chom in ſein vârt
W an ſich ain igelicher vlêizzen ſol
D az man von im avh ſpreche wol
M an ſol von frvmen levten leſen
V nðe ſol doh gerner ſelbe weſen
A ín biðerbe man ðaz iſt aín rat
D er allen lêvten wol an ſtat
J ch han gehoert lange vriſt
D az in ðer werlðe gefrvmet iſt
V on frvmen lêvten hart vil
N v iſt cît ðaz ich ſagen will
W az frvmchait vn waz zvht ſi
V nðe waz tvgenðe vnðe wi
B aiðiv weip vnðe man
S werz von im ſelben niht en chan
Z e gveten dingen chomen ſol
S wer zvht lere merchet wol
E z mag im frvmen an ðer tvgent

Fig. 1. A screen shot of our system. The left pane is the image pane; the right pane contains the transcribed text. The middle pane is used to zoom in and interactively align a single line at a time. In this middle pane, the first two lines are used to edit anchor points. Hovering over one of the anchor lines highlights the matching anchor in the other line. Anchors can be added, dragged, or removed. The other three lines in the middle pane depict the interpolated alignment: a line from the image, a line of warped synthesized text, and the line of synthesized text. Manuscript: Thomasin Circlaere, Welscher Gast, Cod. Pal. germ. 389, fol. 1r, courtesy of the Universitätsbibliothek Heidelberg at http://digi.ub.uni-heidelberg.de/diglit/cpg389/0013 transcription by Jakub Šimek, Heidelberg, SFB 933, Teilprojekt B06.

III. Baseline interactive system

The goal of the transcript alignment system is to produce accurate results with minimal user intervention. The user interface, depicted in Figure 1, comprises a manuscript image pane, a document text pane, and an alignment pane, which is used to align one line of text at a time (though not addressed in this paper, the extension to vertical scripts is straightforward.)

The process starts by automatically detecting the lines of the manuscript using a line detection algorithm, here the line-detection method described in [32]. This method is based on binarization followed by a horizontal projection and a peak-finding algorithm. Once lines are found, they are individually trimmed at their left and right boundaries, as detected through a vertical projection. Our GUI allows the user to then add lines which were missed, remove spurious lines, and adjust the location of the separators between the lines.

Then, individual manuscript lines are aligned with the corresponding text line by line. This alignment is performed in a semi-automatic manner. Anchor points along the horizontal lines are used to describe the location of text letters in the image. Each anchor point divides a segment in the manuscript to a left part and to a right part; it also divides a matching text segment to two parts. Such anchor points can be inserted by the user or set automatically by the system (see Section IV). The insertion of such points, or the adjustment and possibly removal of misplaced automatic points, constitutes the main forms of interaction for the proposed alignment process.

In order to interpolate between the anchor points, we employ a solution previously proposed by the authors [1]. Since this is a general solution, which does not rely on the existence of OCR capabilities, it can be used to align text and images in a wide variety of scripts with minimal effort required for adaptation. Instead of detecting individual graphemes, it robustly employs an optical-flow based technique to directly match the historical image with a synthetic image created from the text at the pixel level. The ability to synthesize such images assumes that, given a manuscript, a similarly looking font is available. The method, however, was shown to perform well even when the manuscript font is vastly different from the one used in the original text.

The method of [1] is independently applied to every segment—every stretch of pixels lying between two adjacent anchor points. This local text-to-image alignment problem thus reduces to that of aligning a cropped image out of a line of text and its matching, synthetic transcript image. For completeness, we briefly describe the alignment method below.

Text to image alignment

The process starts by synthesizing a reference image of the transcript segment using a suitable font. This synthesis is performed in a manner in which the provenance of every pixel of the resulting image is kept; i.e., we know for every pixel which is its corresponding letter.

Then, the cropped image line and the matching generated image are both represented by the same image encoding pipeline. First, each image I is converted to a Four Patch LBP (FPLBP) code image C, where each pixel is assigned an integer value in the range $[0..15]$ [33]. Next, local histograms of FPLBP codes are pulled at each image location. Since most of the ambiguity is in the horizontal direction for horizontal scripts, these histograms are gathered from elliptical domains; a 2D Gaussian filter, with sigmas 2.5 (horizontally) and 1 (vertically), is used for this purpose. Lastly, to compute optical flow between two images, the SIFT-flow method of [34] is applied by replacing its original Dense SIFT representation [35] with the values of these histograms per each pixel.

Once the method is applied, each pixel in the manuscript image segment is mapped to a pixel of the synthetic image, which is, in turn, mapped to the original text letters. Each manuscript image pixel is therefore aligned to the text. This very efficient method can be applied interactively to each image segment once the anchor points are placed or modified. However, it is often the case that in order to obtain very accurate results, the anchor points cannot be spread far apart; that is, larger image sections raise the likelihood that local matching errors will occur.

In order to reduce the manual effort of adding many anchor points by hand, we attempt to learn from parts of the document which have already been accurately aligned. One avenue for improvement would be to synthesize more accurate images using letter samples that were already aligned. However, the method of [1] is robust to specific letter forms and the resulting improvement would be limited. Instead, we choose to use the letter samples in order to automatically place anchor points, as is described next.

IV. Going Viral

The main motivation for transcript alignment is that it provides letter samples that are accessible for both human analysis and machine learning. In our system the automatic analysis occurs already as part of the transcript alignment system in order to reduce the required effort. This is done through the use of letter spotting, using a method that is adapted from the word spotting method we previously proposed [30] for efficient word spotting.

We scale each manuscript line to have a height of 21 pixels and consider letters of a width of 9 pixels. Previously aligned text provides letter samples of varying widths. Letter samples that are too narrow are discarded. For letter samples that are wider, only the central part of the specified width is used. The letter samples are stored in a dataset from which they are retrieved during search.

For the insertion of automatic anchor points to a line of text, the manuscript segment of the entire line is divided into overlapping windows (stride of 1) of the specified width. These are used as queries in the letter spotting search. Each dataset or query image patch is represented as a vector as follows: First the patch is divided to 3×7 non-overlapping cells of size 3×3 that are each encoded by a HOG descriptor [36] of length 31. All descriptors are concatenated and $L2$ normalized. The HOG descriptor r is therefore of dimensionality $31 \times 3 \times 7 = 651$.

A matrix $M \in \mathbb{R}^{5250 \times 651}$, which consists of the vector representations (same as r) for 5250 random 9×21 bounding boxes from all manuscript lines, is then considered. The vector r is transformed to a vector $s \in \mathbb{R}^{5250}$ by means of a linear projection: $s = Mr$. In other words, the normalized descriptor vector is represented by its cosine similarities to a predetermined set of exemplars.

Then, a max-pooling step takes place. The set of indices $[1..5250]$ is randomly split into fixed groups I_i of size 15. Given a vector s, this max-pooling is performed simply by considering one measurement per such triplet I_i that is the maximal value among the three indices of I_i in the vector v. Put differently, let t be the vector of length 350 that results from the max-pooling process as applied to the vector s. Then $t_i = \max_{j \in I_i} s_j$.

Queries are performed based on the Euclidean distance between the vectors t associated with the patches extracted from the image line and the vectors t associated with the already stored letter samples. Only the top-1 retrieval is considered. Among overlapping image regions non-maximal suppression is employed, i.e., from the group of overlapping scan windows in the new line to transcribe, only the scan window with the lowest Euclidean distance to the top-1 search result is considered as a possible match.

Eliminating matching conflicts: The process above is based on visual similarity alone and contains many false matches. Next, we select an optimal set of anchor points by adding the information arising from the alphabet letters associated with the letter samples and the actual text to be aligned. This is done using binary linear programming, which is computed very efficiently using standard solvers for problems of the scale at hand.

This process takes into account the visual distance d_i for a potential letter sample p_i to the retrieving query window q_i based on the corresponding vectors t. It also considers the expected distance d_{ij}, along the horizontal axis, of each text letter l_j from the position of q_i. In order to compute this distance, l_j is first associated with an image location based on the optical flow method described in Section III.

We formulate the problem as a bipartite matching problem. Each matched dataset letter sample p_i is a potential candidate to a text letter l_j in the new line's transcript if they represent the same alphabet letter. In this case, we define $W_{ij} = exp(-d_i - d_{ij}/\sigma)$, for some parameter σ. If the letter associated with p_i and l_j is not the same, $W_{ij} = 0$. The value of σ is set in all our experiments to 50 to allow a very relaxed consideration of the predicted location along the horizontal axis.

The following optimization problem is solved in order to recover the selected matches, which are represented by the binary variables x_{ij}.

$$\begin{aligned} \max_{x_{ij} \in \{0,1\}} & \sum_{i,j} W_{ij} x_{ij} \\ \text{subject to} & \sum_{i} x_{ij} \leq 1, \ \forall j \\ & \sum_{j} x_{ij} \leq 1, \ \forall i \\ & x_{i_1,j_1} + x_{i_2,j_2} \leq 1, \ \forall i_1 > i_2, j_1 < j_2 \end{aligned}$$

The first two constraints state that no query window q_i can be associated with multiple text letters l_i and vice versa. Note that it is possible that the same database sample appears twice, i.e., $p_i = p_{i'}$ for $i \neq i'$. The third constraint enforces the order of matches. It cannot happen that two image patches

q_{i_1} and q_{i_2} appear in a different order in the line than that of the matched l_{j_1} and l_{j_2}.

Finally, we note that some letters tend to create more false matches than others. For each alphabet letter we hold usage counters. Once an automatic anchor created for a letter is rejected more than 30% of the time, no more suggestions are made based on this letter.

V. Experiments

The complete system was implemented and evaluated on a variety of historical documents in various scripts. In each experiment, the system was run line by line. In each line, the transcript alignment was performed to the highest possible accuracy with as many manual interventions as needed. The amount of manual intervention in each line was recorded. In order to eliminate ambiguities in the evaluation, dragging of anchor points was not permitted. Only insertion and deletion were used. If dragging was allowed, it could have been used to created anchors at locations that are unrelated to the letter spotting results, hence masking the success and failures of the automatic anchor placement.

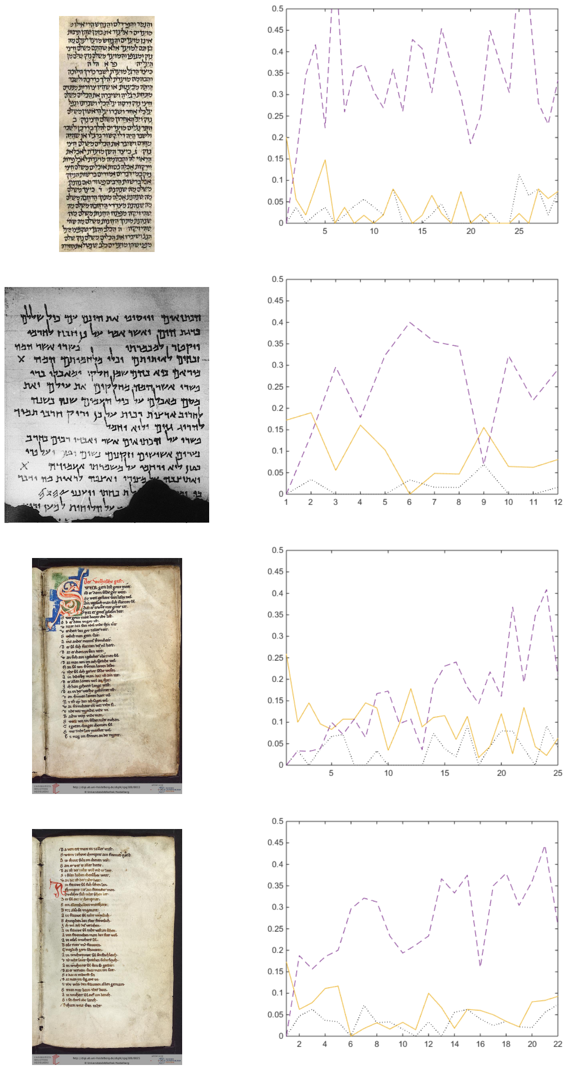
Figures 2 and 3 depicts the alignment effort on sample documents. The number of detected anchor points, which is twice the number of spotted letters, is shown, as well as the number of anchor points removed or added. Overall, as the alignment process progresses, the number of spotted letters increases and the manual effort decreases. In addition, it should be noted that removing a false anchor point is much easier than adding a new one. The latter requires twice the number of mouse clicks and a much higher accuracy.

Figure 4 is a typical example of the amount of manual labor required. The automatic alignment, obtained by the anchors arising from the spotted letters combined with the optical flow solution, is very descent. However, in order to obtain very accurate results, some adjustments are necessary.

VI. Conclusion

The combination of letter spotting and a robust interpolation mechanism allows for accurate and rapid transcript alignment. While the idea that document analysis should be bootstrapped with some manual labor, becoming increasingly automatic, is intuitive, very few document analysis systems to date have demonstrated this property. We believe that the success of the system described here comes from the realization that instead of using the obtained knowledge in order to improve the underlying alignment engine through machine learning, an optimization mechanism can select from an over-complete set of hints that are derived from the growing set of annotated results.

The entire code of the system will be made publicly available as an open source project.

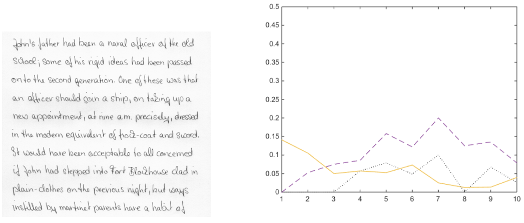


(a) (b)

Fig. 2. Sample historical documents and plots depicting the amount of user intervention required at every line. (a) Samples in various languages and script types. (b) For each line (x-axis), the y-axis counts the number of spotted anchors (dashed purple line), the number of manual added anchor points (solid brown line), and the number of automatically placed anchor points which were removed (dotted black line). These have been normalized to twice the number of letters per line, since one spotted letter provides two anchors. As can be seen, the number of automatically detected letters increases as more aligned lines become available, reducing the amount of manual work required. From top to bottom: (i) ms Kaufmann MS A50, fol. 128r col 2. By courtesy of the Oriental Collection of the Library and Information Centre of the Hungarian Academy of Sciences transcription by Daniel Stökl Ben Ezra and Michael Krupp. (ii) The Commentary on Habakkuk Scroll (1QpHab); Qumran, Cave 1; Accession number: 95.57/28. Courtesy Israel Museum and Shrine of the Book. (iii) See Fig. 1. (iv)See Fig. 1 (fol. 7r).

Acknowledgments

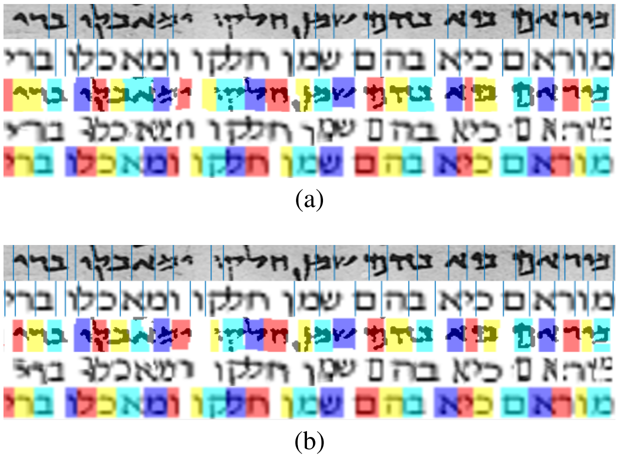
This tool is the result of a cooperation between the EPHE (École Pratique des Hautes Études), Paris, Tel Aviv University, and the Open University of Israel. The TAU authors would like to thank the Ministry of Science, Technology and Space (Israel-Taiwan grant number 3-10341) and the German-Israeli Foundation (grant I-145-101.3-2013) for their support. This research was also supported in part by the Israel Science Foundation (grant 1330/14). This research was initiated at the Dagstuhl Perspectives Workshop 12382, “Computation and Palaeography: Potentials and Limits”.



John's father had been a naval officer of the old school; some of his rigid ideas had been passed onto the second generation. One of these was that an officer should join a ship, on taking up a new appointment, at nine a.m. precisely, dressed in the modern equivalent of frock-coat and sword. It would have been acceptable to all concerned if John had stepped into Fort Blockhouse clad in plain-clothes on the previous night, but ways instilled by martinet parents have a habit of

(a) (b)

Fig. 3. A cursive document from the IAM database [37]. Please refer to the caption of Fig. 2 for details.



(a)

(b)

Fig. 4. A single aligned line from the middle of the document (1QpHab col. VI line 5). Each plot includes an image line and a transcript line with the anchor points marked; an image line with marked letter locations; a warped synthesized transcript line; the synthesized transcript line with matching letters marked by matching colors. (a) The automatic alignment, which includes 22 anchor points added automatically by spotting 11 letters. (b) An improved alignment after adding five anchor points.

References

[1] T. Hassner, L. Wolf, and N. Dershowitz, “OCR-free transcript alignment,” in *ICDAR*, 2013, pp. 1310–1314.

[2] J. D. Hobby, “Matching document images with ground truth,” *IJDAR*, vol. 1, no. 1, pp. 52–61, 1998.

[3] V. Lavrenko, T. M. Rath, and R. Manmatha, “Holistic word recognition for handwritten historical documents,” in *DIAL*, 2004.

[4] C. Huang and S. N. Srihari, “Mapping transcripts to handwritten text,” in *ICFHR*, 2006.

[5] S. Zinger, J. Nerbonne, and L. Schomaker, “Text-image alignment for historical handwritten documents,” in *Document Recognition and Retrieval*, 2009, pp. 703–724.

[6] J. Puigcerver, A. H. Toselli, and E. Vidal, “Word-graph and character-lattice combination for KWS in handwritten documents,” in *ICFHR*, 2014, pp. 181–186.

[7] A. Toselli, V. Romero, and E. Vidal, “Alignment between text images and their transcripts for handwritten documents,” in *Language Technology for Cultural Heritage*, 2011, pp. 23–37.

[8] J. A. Sánchez, V. Bosch, V. Romero, K. Depuydt, and J. de Does, “Handwritten text recognition for historical documents in the transcriptorium project,” in *Proc. Int. Conf. on Digital Access to Textual Cultural Heritage*, 2014.

[9] E. M. Kornfield, R. Manmatha, and J. Allan, “Text alignment with handwritten documents,” in *DIAL*, 2004.

[10] D. Jose, A. Bhardwaj, and V. Govindaraju, “Transcript mapping for handwritten English documents,” in *DRR*, ser. SPIE Proceedings, B. A. Yanikoglu and K. Berkner, Eds., vol. 6815. SPIE, 2008, p. 68150.

[11] L. Lorigo and V. Govindaraju, “Transcript mapping for handwritten arabic documents,” in *Electronic Imaging*, 2007.

[12] M. Zimmermann and H. Bunke, “Automatic segmentation of the IAM off-line database for handwritten English text,” in *ICPR*, 2002.

[13] J. L. Rothfeder, R. Manmatha, and T. M. Rath, “Aligning transcripts to automatically segmented handwritten manuscripts,” in *Document Analysis Systems*, ser. Lecture Notes in Computer Science, H. Bunke and A. L. Spitz, Eds., vol. 3872. Springer, 2006, pp. 84–95.

[14] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, “Transcription alignment of Latin manuscripts using HMMs,” in *HIP*, 2011.

[15] A. Toselli and E. Vidal, “Fast HMM-filler approach for key word spotting in handwritten documents,” in *ICDAR*, 2013.

[16] B. Zhu and M. Nakagawa, “Online handwritten Japanese text recognition by improving segmentation quality,” in *ICFHR*, 2008.

[17] F. Yin, Q.-F. Wang, and C.-L. Liu, “Integrating geometric context for text alignment of handwritten chinese documents.”

[18] Y. Leydier, V. Eglin, S. Bres, and D. Stutzmann, “Learning-free text-image alignment for medieval manuscripts ,” in *ICFHR*, 2014.

[19] N. Stamatopoulos, G. Louloudis, and B. Gatos, “Efficient transcript mapping to ease the creation of document image segmentation ground truth with text-image alignment,” in *ICFHR*, 2010.

[20] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, “A novel word spotting method based on recurrent neural networks,” *PAMI*, 2012.

[21] B. Gatos and I. Pratikakis, “Segmentation-free word spotting in historical printed documents,” in *ICDAR*, 2009.

[22] J. A. Rodríguez-Serrano and F. Perronnin, “Local gradient histogram features for word spotting in unconstrained handwritten documents,” in *ICFHR*, 2008.

[23] A. Fischer, A. Keller, V. Frinken, and H. Bunke, “HMM-based word spotting in handwritten documents using subword models,” in *ICPR*, 2010.

[24] R. Manmatha and N. Srimal, “Scale space technique for word segmentation in handwritten documents,” in *Scale-Space Theories in Computer Vision*, 1999.

[25] J. Almazan, A. Gordo, A. Fornés, and E. Valveny, “Handwritten word spotting with corrected attributes,” in *ICCV*, 2013.

[26] L. Rothacker, M. Rusiñol, and G. Fink, “Bag-of-features HMMs for segmentation-free word spotting in handwritten documents,” in *ICDAR*, 2013.

[27] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, “Efficient exemplar word spotting,” in *BMVC*, 2012.

[28] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, “Browsing heterogeneous document collections by a segmentation-free word spotting method,” in *ICDAR*, 2011.

[29] A. J. Newell and L. D. Griffin, “Multiscale histogram of oriented gradient descriptors for robust character recognition,” in *ICDAR*, 2011.

[30] A. Kovalchuk, L. Wolf, and N. Dershowitz, “A simple and fast word spotting method,” in *ICFHR*, 2014, pp. 3–8.

[31] Q. Liao, J. Z. Leibo, Y. Mroueh, and T. Poggio, “Can a biologically-plausible hierarchy effectively replace face detection, alignment, and recognition pipelines?” *CoRR*, vol. abs/1311.4082, 2013.

[32] L. Wolf, R. Littman, N. Mayer, T. German, N. Dershowitz, R. Shweka, and Y. Choueka, “Identifying join candidates in the Cairo Genizah,” *IJCV*, 2011.

[33] L. Wolf, T. Hassner, and Y. Taigman, “Descriptor based methods in the wild,” in *Post-ECCV Faces in Real-Life Images Workshop*, 2008.

[34] C. Liu, J. Yuen, and A. Torralba, “Sift flow: Dense correspondence across scenes and its applications,” *PAMI*, 2011.

[35] A. Vedaldi and B. Fulkerson, “Vlfeat: An open and portable library of computer vision algorithms,” in *Proc. int. conf. on Multimedia*, 2010.

[36] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, 2005.

[37] U.-V. Marti and H. Bunke, “The iam-database: an english sentence database for offline handwriting recognition,” *IJDAR*, 2002.