OCR-Free Transcript Alignment

Tal Hassner
Dept. of Mathematics and Computer Science
The Open University
Israel
Email: hassner@openu.ac.il

Lior Wolf
School of Computer Science
Tel Aviv University
Tel-Aviv, Israel
Email: wolf@cs.tau.ac.il

Nachum Dershowitz
School of Computer Science
Tel Aviv University
Tel-Aviv, Israel
Email: nachumd@tau.ac.il

***Abstract*—Recent large-scale digitization and preservation efforts have made images of original manuscripts, accompanied by transcripts, commonly available. An important challenge, for which no practical system exists, is that of aligning transcript letters to their coordinates in manuscript images. Here we propose a system that directly matches the image of a historical text with a synthetic image created from the transcript for the purpose. This, rather than attempting to recognize individual letters in the manuscript image using optical character recognition (OCR). Our method matches the pixels of the two images by employing a dedicated dense flow mechanism coupled with novel local image descriptors designed to spatially integrate local patch similarities. Matching these pixel representations is performed using a message passing algorithm. The various stages of our method make it robust with respect to document degradation, to variations between script styles and to non-linear image transformations. Robustness, as well as practicality of the system, are verified by comprehensive empirical experiments.**

I. Introduction

While high-quality images are 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 Unfortunately, optical character recognition (OCR) in historical documents is notoriously difficult, for which reason transcription is currently performed manually for documents of significant historical value.

A very common scenario is line-by-line transcription. This is the case for many of the most valuable collections recently digitized and made available online. Examples include the Dead Sea Scrolls (http://www.deadseascrolls.org.il), some of the Cairo Genizah (http://www.genizah.org), the Early English Laws collection (http://www.earlyenglishlaws.ac.uk), Codex Sinaiticus (http://codexsinaiticus.org), the George Washington Papers (http://rotunda.upress.virginia.edu/founders/GEWN.html), and much of the Tibetan Buddhist Canon (http://www.tbrc.org; http://idp.bl.uk). In all these cases, explicit information of what text is written on each line of each page of the manuscript is available.

Our goal is to solve a very specific task for which no practical system currently exists, namely, determining letter-by-letter mappings between transcription texts and the matching image patches in digitized copies of scanned manuscripts. We assume that the text is divided into pages and lines. We also assume that the problem of line finding in the image is solvable. From our experience with the Cairo Genizah collection [1], and from the experience of the larger community, the latter presumption is reasonable. Lastly, we assume that we have access to a computer font that is somewhat similar to the script style used in the original manuscript. For future work on relaxing these assumptions, please refer to Section VIII.

The solution we propose is a general one in the sense that we do not try to learn to identify graphemes from the matching manuscript and text. Instead, we take a "synthesis, dense-flow, transfer" approach, previously used for single-view depth-estimation [2] and image segmentation [3]. Specifically, we suggest a robust "optical-flow" technique to directly match the historical image with a synthetic image created from the text. This matching, is performed at the *pixel level* allowing transfer of the (known) letters from pixels in the synthetic image to those in the historical document image.

II. Previous Work

The problem of matching text with images of the (printed) text was discussed in the past in [4], [5], 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., [6], [7] and more recently [8]). But OCR for handwritten text, with which we are dealing here, is notoriously difficult. In [9], [10], 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 [11], [12], [13], 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 [14] and for Chinese in [15]). 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, per-pixel matching techniques.

Next, we summarize some of the techniques that have been incorporated into our method.

SIFT flow. Given two images of approximately the same scene, query image I_Q and reference image I_R, the SIFT flow method [16] has been demonstrated to be an effective way for computing the dense, per-pixel flow between them. Generally speaking, the local properties of each image are captured by a local, appearance-based descriptor. Let the local representation at pixel p of one image be $f(I_Q, p)$ and similarly for the second image, $f(I_R, p')$. The function f represents a feature transform, applied to pixel p (p'). The original SIFT flow method uses the SIFT descriptor [17], extracted at a constant scale and orientation (i.e., Dense-SIFT [18]). Here, however, we show that this representation may not be ideal for manuscript images.

The optimization function of SIFT flow seeks to find correspondences between the query and the reference view; that is, to obtain for each query pixel p a vector $w(p) = [u(p), v(p)]^\top$, mapping it to a pixel p' in the reference. A good set of correspondences is expected to satisfy the following requirements:

1) *Small displacement.* There is an underlying assumption, which can be controlled using appropriate parameters, that the views are similar and therefore matching a pixel to nearby pixels is preferred. That is, $|u(p)| + |v(p)|$ is minimized for all pixels.
2) *Smoothness.* Adjacent query pixels, p_1 and p_2, should be the warping target of adjacent reference pixels p'_1 and p'_2. Therefore, the values $|u(p_1) - u(p_2)|$ and $|v(p_1) - v(p_2)|$ are minimized for all query pixels.
3) *Appearance similarity.* For every pixel p, its appearance should resemble the appearance of its matched reference pixel. That is, for all pixels, $\|f(I_Q, p) - f(I_R, w(p))\|_1$ is minimal.

Each of these three requirements has roots in the optical-flow and image-matching literature. Requirements 1 and 2 have been imposed by previous methods for optical-flow (see, e.g., [19]). The SIFT flow work [16] has extended previous methods in order to allow for the third requirement. The optimization itself uses the dual-plane representation [20] and is used to decouple the smoothness constraint for the x and y components of the flow, as well as the message-passing algorithm presented in [21], which includes the distance transform, bipartite updates and a mult-grid implementation.

LBP and its variants. Local binary patterns (LBPs) [22], [23], [24], originally a texture descriptor, have been shown to be extremely effective for face recognition [25]. Some variants have also proven useful in a wide range of computer vision domains, including object localization [26] and action recognition [27]. The simplest form of LBP is created at a particular pixel location by thresholding the 3×3 neighborhood surrounding the pixel with the central pixel's intensity value, and treating the subsequent pattern of 8 bits as a binary number. A histogram of these binary numbers in a predefined region is then used to encode the appearance of that region.

The Four-Patch LBP [28] employed here is a patch-based descriptor that has some similarities, with regard to spatial arrangements, to a variant of LBP called Center-Symmetric LBP (CSLBP) [29]. In CSLBP, eight intensities around a central point are sampled. The binary vector encoding the local appearance at the central point consists of four bits that contain the comparison of intensities to the intensities on the opposite side. Another variant, called Multi-block LBP [30], replaces intensity values in the computation of LBP with the mean intensity value of image blocks. Despite the similarity in terms and the usage of local image patches, this method is quite different from the patch-based LBPs [28] used here.

III. Method Overview

As preprocessing, we employ the line-detection method described in [1]. 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.

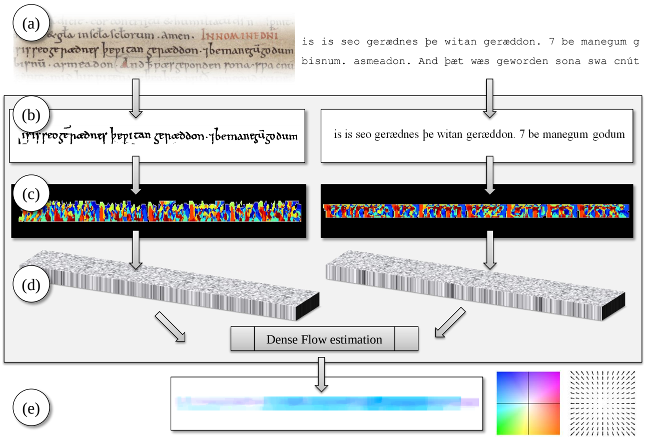

(a)
is is seo gerædnes þe witan geræddon. 7 be manegum g
bisnum. asmeadon. And þæt wæs geworden sona swa cnút
(b)
is is seo gerædnes þe witan geræddon. 7 be manegum godum
(c)
(d)
Dense Flow estimation
(e)

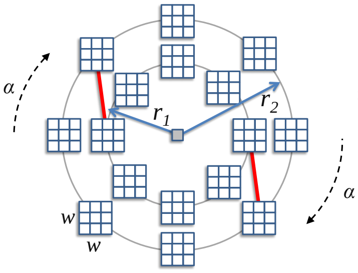
Fig. 1. The stages of the proposed dense-flow method. (a) Our input is a historical manuscript image, along with a line-by-line transcript of the text. (b, left) Each manuscript line is binarized, horizontally projected and trimmed; (b, right) the matching transcript line is rendered using an appropriate font to produce a matching reference image. (c) FPLBP codes [28] are produced for the two images (shown color coded). (d) Each pixel is represented by a 16-valued, weighted histogram of the FPLBP codes in its elongated neighborhood. A SIFT flow [16] variant is then applied to form dense correspondences between these representations in the two views. (e) The output flow (color coded) assigns a matching pixel from the rendered image, along with its known character label, to each pixel in the manuscript line image.

Our problem thus reduces to that of aligning a cropped image of a line of text and its matching transcript. First, we synthesize a reference image of the transcript text line using a suitable reference font. The reference text is synthesized in a manner in which the provenance of every pixel of the resulting image is kept; that is, we know for every pixel which is the corresponding letter. As we show in our experiments (Section VII), the font used for synthesizing these references need only approximately match the handwriting in the actual document. This is evident both in the synthetic experiments (where images of different fonts are matched) and when applied to real manuscript photos, where modern fonts are aligned with ancient handwritings.

Our method for representing the cropped line and the matching generated image and for computing the flow between these images is illustrated in Fig. 1. 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 $[0..15]$ (Section IV). Next, local histograms of FPLBP codes are pulled at each image location. Since most of the ambiguity is in the horizontal direction, these histograms are gathered from elliptical domains. Lastly, to compute optical flow between two images, the SIFT flow method [16] is applied by replacing the dense SIFT with the values of these histograms per pixel.

IV. Four-Patch LBP Encoding

The LBP descriptor family (e.g., [25], [31], [30]) encodes local texture by constructing short binary strings from the values around each image pixel. The Three-Patch LBP



(a)

$$\texttt{FPLBP}_{r1,r2,8,3,1}(p) = \\ f(d(C_{10},C_{21}) - d(C_{14},C_{25}))2^0 + \\ f(d(C_{11},C_{22}) - d(C_{15},C_{26}))2^1 + \\ f(d(C_{12},C_{23}) - d(C_{16},C_{27}))2^2 + \\ f(d(C_{13},C_{24}) - d(C_{17},C_{20}))2^3$$

(b)

Fig. 2. (a) The Four-Patch LBP code. Two patch pairs involved in computing a single bit value with parameter $\alpha = 1$ are connected by (red) lines. (b) The FPLBP code computed for parameters $S = 8$, $w = 3$, and $\alpha = 1$. The C_{ij} denote the various patches; the first subscript indicates the ring (inner or outer) and the second denotes their location on the ring, starting from index 0 at twelve o'clock.

(TPLBP) and the Four-Patch LBP [28] take after the Self-Similarity Descriptor [32] by using these short bit strings to represent similarity relationships among neighboring patches of pixels. Both TPLBP and FPLBP have been shown to capture local image information complementary to the information reflected by pixel-based descriptors; utilization of patch-based descriptors along with pixel-based descriptors has been shown to considerably enhance classification accuracy [33].

Encoding FBLBP is done by considering two rings around each pixel. Given the radii of the inner and outer rings, S evenly-distributed patches are sampled along each ring. To encode a single pixel, all $S/2$ pairs of opposing patches in the inner ring are considered, and one compares the similarity between the opposing inner patches to α-distant patches on the outer ring (see Fig. 2). Hence, the length of the binary code per single pixel is $S/2$ bits. The FBLBP codes are very compact, typically having only 16 values, while remaining highly descriptive. In [34], these codes were used to encode face images, and proven to perform almost as well as the significantly more expensive collections of SIFT [17] descriptors.

V. Flow Computation

For each FPLBP image, we compute at each pixel a local histogram, in a manner reminiscent of the Distribution Fields representation, recently used for object tracking in [35]. To account for the added ambiguity in the horizontal direction, the histograms are pulled using an elliptical domain. Furthermore, the histograms are weighed such that pixels near the center of the window contribute more. A 2D Gaussian filter, with sigmas 2.5 (horizontally) and 1 (vertically), is used for this purpose.

To efficiently smooth these dense histograms, we employ the fast approximation method of [36], which uses integral images and multiple averaging passes in order to approximate Gaussian filtering. Here, for each FPLBP value, we create a binary map and then perform Gaussian smoothing so as to propagate the values to the nearby histograms. Following this step, each pixel is represented by a 16-bit vector.

Given two histogram fields as described above, we employ a modified SIFT flow, based on its original implementation [16], to compute the dense matches themselves. The dense SIFT descriptors originally used by SIFT flow are replaced by the histogram for each pixel, smoothed as described above; the code was left otherwise unchanged.

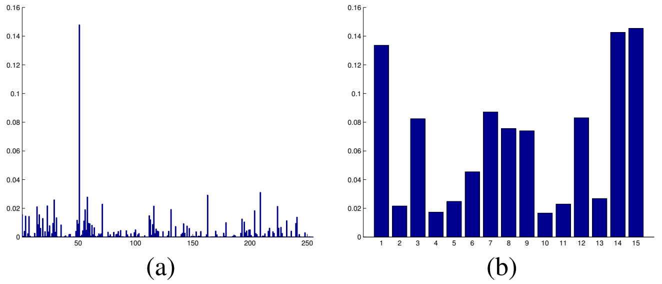


Fig. 3. (a) The histogram of TPLBP values as computed on a set of document images. (b) The histogram of FPLBP codes, which is much more uniform.

VI. Why Four-Patch LBP?

The Three-Patch LBP (TPLBP) is a patch-based Local Binary Pattern descriptor in which a central patch is compared to a pair of patches. A bit value is set to 1 if the central patch is more similar to the first patch of the pair. A ring of eight patch locations is placed around the central patch, and all pairs of one patch and the patch located two further locations clockwise are used. The result is an eight bit code or, equivalently, values in the range $[0..255]$.

Previous contributions have demonstrated that FPLBP is inferior to TPLBP and LBP in face recognition [28] and texture recognition [37]. However, our initial experiments have demonstrated that TPLBP-based flow performs only slightly better than SIFT flow. (Unsurprisingly, the basic LBP descriptor is even less suitable for this task, since the usage of patches is crucial for such images.)

Several properties of FPLBP make it suitable for the task at hand. First it is much more compact than TPLBP and even LBP. Since the run time of our method is dominated by the run time of the SIFT flow method, which in turn depends on the underlying descriptor length, this is a very desirable property of the FPLBP codes. Second, and more importantly, out of the range of codes produced by TPLBP (and also LBP), only a small fraction is common in typical document images. The underlying reason is that while, in intensity images, there is little inherent restriction on scene patterns, the local structure of documents is very restrictive. Thus, for documents, the simplicity of FPLBP, which essentially compares the variability in the local appearance between left/right, top/down, and both diagonals, is an advantage. Lastly, the compact range of values ensures that similar patterns are clumped together. Fig. 3 demonstrates the difference, with this regard, between TPLBP and FPLBP; the histogram of TPLBP codes is much sparser than the FPLBP counterpart, and hence more efficiently uses different values to capture different appearances.

VII. Experiments

We ran two sets of extensive experiments: synthetic experiments, using computer generated images, and real-data experiments, using a varied set of historical documents.

For the purpose of the synthetic experiments, we took a page from *A Tale of Two Cities* by Charles Dickens. Then

TABLE I. RESULTS OF THE SYNTHETIC EXPERIMENT. FOR EACH QUERY FONT AND FOR EACH OF THE THREE TESTED METHODS, THE AVERAGE DISPLACEMENT ERROR WAS COMPUTED. THE TABLE SHOWS THE MEAN VALUE (±SD) PER METHOD OVER ALL 269 QUERY FONTS. ALSO SHOWN ARE THE MEDIAN DISPLACEMENT ERROR AND THE PERCENT OF FONTS ON WHICH EACH METHOD PERFORMS BEST OUT OF THE THREE METHODS. THE BASELINE METHOD REFERS TO A LINEAR STRETCHING OR SHRINKING OF THE REFERENCE TEXT TO MATCH THE DIMENSIONS OF THE TEST TEXT.

Method:	Baseline	SIFT-flow	Proposed
Mean error ± SD	10.21 ± 4.2	8.38 ± 3.8	6.18 ± 3.1
Median error	11.23	7.42	5.27
Percent best error	6%	17%	77%

we created, using the 274 standard fonts installed on our PC, image documents containing the text, where the location, in pixels, of each character is known. This large number of fonts is used in order to test the robustness of our method over a large number of query-reference font combinations. We scaled all fonts to the height of 19 pixels and the resulting image was at a resolution of 1140×716 pixels. The average width of a character was about nine pixels.

As a reference image, we used the document created using the Times New Roman font. Non-English Graphemes such as "Webdings", "Wingdings", and "Euclid Math" were removed, and a total of 269 query images were tested. Each image contained 50 lines that were detected automatically (see Section III). For testing purposes, each character was associated with its center (the mean value of its pixel coordinates).

For each character, we computed the distance of its center in the reference image, warped by the tested method to the query image, to its center in the reference image. Because both reference and test images are synthetic, we have per-pixel character labels for both images. The centroid of each character, in each image, can therefore be simply computed by considering all the pixels that share the same character label. We note that this error measure would not be zero even for a perfect mapping since the center of mass of each character varies depending on the font. Still, it is expected to be low when the mapping is accurate. The distances are then aggregated by computing the mean distance per document.

The results are summarized in Table I. Shown are the mean and standard deviation of the average per-document distances, as well as the median distance and the percent of query fonts on which each method gave the best results. The methods compared are a baseline method, in which the line is stretched or shrunk linearly between the query and the reference, the result of using the original SIFT-flow, and the result of the proposed method. Also tested were an LBP-based method and a TPBLP-based method, but neither was competitive and so they are omitted.

For the real-data experiment, we used pages from a wide variety of manuscripts of various sources and languages, including Tibetan, English, Greek, and Hebrew. Some results are presented in Fig. 4, where, for brevity, only one typical line per document is shown.

VIII. CONCLUSION

The problem of transcript alignment, when formulated as a pixel mapping problem, has unique characteristics that challenge optical-flow methods originally designed for conventional photos. Here, we combine three powerful methods so as to create an effective representation for the purpose of establishing dense correspondences: SIFT flow provides a modern optimization framework that is flexible with regard to the underlying descriptor; the spatial histogram representation adds smoothness that allows for better matching of uncertain image parts; Four-Patch LBP codes provide a succinct, self-similarity based descriptor that is both symmetric and robust in the face of measurement noise.

Our results show a significant improvement over baseline methods on a wide variety of synthetic and authentic transcript and image pairs. Our method forms a strong foundation from which more elaborate systems can be constructed. One obvious direction is to use the method in an iterative manner in which, after each application, letter appearances would be learned and employed to resolve ambiguities. Note that the identification of such ambiguities is straightforward given the local nature of the suggested cost function.

In addition, some of our underlying assumptions can be relaxed. When transcripts are provided only per paragraph or per page, one could estimate the number of characters in a manuscript line based on the line's length and density, and use multiple competing hypotheses to find the best match for the line break. Finally, incorporating both a self-learning mechanism and a line-splitting mechanism, one can imagine an elaborate method that would enable searching within an existing corpus for the text depicted in a historical image.

ACKNOWLEDGMENTS

The authors would like to thank Yaacov Choueka, Roni Shweka, Peter Stokes, Orna Almogi, and Dorji Wangchuk for help collecting relevant data. LW and ND are supported through a research grant from Google, Inc. This research was initiated at the Dagstuhl Perspectives Workshop 12382, "Computation and Palaeography: Potentials and Limits".

REFERENCES

[1] L. Wolf, R. Littman, N. Mayer, T. German, N. Dershowitz, R. Shweka, and Y. Choueka, "Identifying join candidates in the Cairo Genizah," *Int. J. Comput. Vision*, vol. 94, no. 1, pp. 118–135, Aug. 2011.

[2] T. Hassner and R. Basri, "Example based 3D reconstruction from single 2D images," in *Proc. Conf. Comput. Vision Pattern Recognition workshops*, 2006.

[3] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2009, pp. 1972–1979.

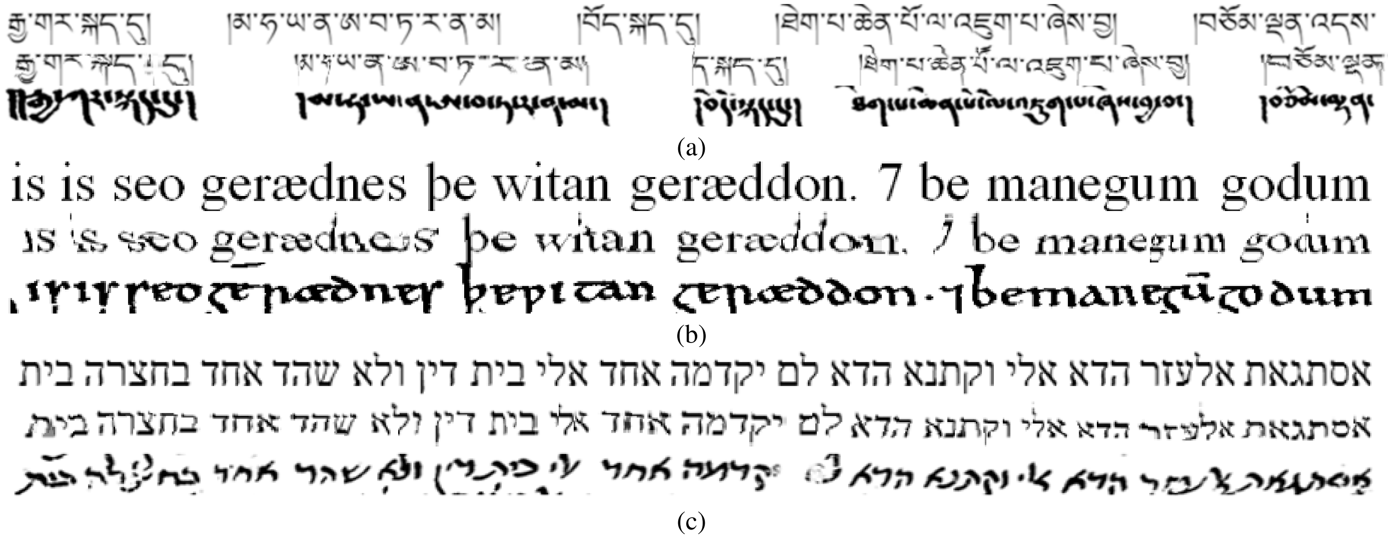
[4] J. D. Hobby, "Matching document images with ground truth," *IJDAR*, vol. 1, no. 1, pp. 52–61, 1998.

[5] C. I. Tomai, B. Zhang, and V. Govindaraju, "Transcript mapping for historic handwritten document images," in *Frontiers in Handwriting Recognition*, 2002, pp. 413–418.

[6] V. Lavrenko, T. M. Rath, and R. Manmatha, "Holistic word recognition for handwritten historical documents," in *DIAL*. IEEE Computer Society, Jan. 2004, pp. 278–287.

[7] C. Huang and S. N. Srihari, "Mapping transcripts to handwritten text," in *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition*. IEEE Computer Society, 2006, pp. 15–20.

[8] M. Al Azawi, M. Liwicki, and T. M. Breuel, "WFST-based ground truth alignment for difficult historical documents with text modification and layout variations," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 865 818–865 818.



(a)

(b)

(c)

Fig. 4. Examples of alignment results on actual historical manuscripts. Each group contains a synthesized image of the transcript, a warped version of the synthesized image as mapped to the historical document, and the original line, as taken from the historical document. A good result would have the characters of the warped text in the middle row exactly above the matching characters in the third row, and would tend to deform the modern font to appear similar to the original scribal hand. (a) A line from the Tibetan bKa' gdams gsung 'bum collection. The script and the font are not entirely compatible, and the transcript is missing a syllable; however, the alignment is reasonable. (b) A line from Cnut's Oxford code. (c) A document from the Cairo Genizah.

[9] E. M. Kornfield, R. Manmatha, and J. Allan, "Text alignment with handwritten documents," in *DIAL*. IEEE Computer Society, 2004, pp. 195–211.

[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] M. Zimmermann and H. Bunke, "Automatic segmentation of the IAM off-line database for handwritten English text," in *International Conference on Pattern Recognition*, vol. 4, 2002.

[12] 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.

[13] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, "Transcription alignment of Latin manuscripts using hidden Markov models," in *Workshop on Historical Document Imaging and Processing*, ser. HIP '11. ACM, 2011, pp. 29–36.

[14] B. Zhu and M. Nakagawa, "Online handwritten Japanese text recognition by improving segmentation quality," in *Proc. 11th Intl. Conf. on Frontiers in Handwriting Recognition*, Montreal, Canada, 2008, pp. 379–384.

[15] F. Yin, Q.-F. Wang, and C.-L. Liu, "Integrating geometric context for text alignment of handwritten Chinese documents," in *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, Nov. 2010, pp. 7 –12.

[16] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *Trans. Pattern Anal. Mach. Intell.*, 2011.

[17] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[18] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proc. int. conf. on Multimedia*, 2010, pp. 1469–1472.

[19] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *ECCV*, 2004.

[20] A. Shekhovtsov, I. Kovtun, and V. Hlavac, "Efficient MRF deformation model for non-rigid image matching," in *Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1 –6.

[21] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vision*, vol. 70, no. 1, Oct. 2006.

[22] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative-study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, 1996.

[23] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, 2002.

[24] T. Ojala, M. Pietikäinen, and T. Mäenpää, "A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification," in *ICAPR*, 2001.

[25] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, 2006.

[26] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured hog-lbp for object localization," in *CVPR*, Jun. 2011, pp. 1393 –1400.

[27] V. Kellokumpu, G. Zhao, and M. Pietikainen, "Human activity recognition using a dynamic texture based method," in *BMVC*, 2008.

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

[29] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns," in *Indian Conference Computer Vision, Graphics and Image Processing*, 2006.

[30] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li, "Face detection based on multi-block LBP representation," in *IAPR/IEEE International Conference on Biometrics*, 2007.

[31] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Analysis and modeling of faces and gestures (AMFG)*, 2007.

[32] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," *Proc. Conf. Comput. Vision Pattern Recognition*, pp. 1–8, June 2007.

[33] L. Wolf, T. Hassner, and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics," *Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1978–1990, 2011.

[34] M. Guillaumin, J. Verbeek, C. Schmid, I. Lear, and L. Kuntzmann, "Is that you? Metric learning approaches for face identification," in *Proc. Int. Conf. Comput. Vision*, 2009.

[35] L. Sevilla-Lara and E. Learned-Miller., "Distribution fields for tracking," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2012.

[36] P. Kovesi, "Fast almost-Gaussian filtering," in *Proceedings of the 2010 International Conference on Digital Image Computing: Techniques and Applications*, ser. DICTA '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 121–125.

[37] S. Ebert, D. Larlus, and B. Schiele, "Extracting structures in image collections for object recognition," in *European Conf. Comput. Vision*, 2010.