Word Segmentation Method for Handwritten Documents based on Structured Learning

Jewoong Ryu, Hyung Il Koo, *Member, IEEE*, and Nam Ik Cho, *Senior Member, IEEE*

***Abstract*—Segmentation of handwritten document images into text-lines and words is an essential task for optical character recognition. However, since the features of handwritten document are irregular and diverse depending on the person, it is considered a challenging problem. In order to address the problem, we formulate the word segmentation problem as a binary quadratic assignment problem that considers pairwise correlations between the gaps as well as the likelihoods of individual gaps. Even though many parameters are involved in our formulation, we estimate all parameters based on the Structured SVM framework so that the proposed method works well regardless of writing styles and written languages without user-defined parameters. Experimental results on ICDAR 2009/2013 handwriting segmentation databases show that proposed method achieves the state-of-the-art performance on Latin-based and Indian languages.**

***Index Terms*—Handwritten documents, structured SVM, word segmentation.**

I. Introduction

SEGMENTATION of document images into text-lines and words is an important step for the document understanding [1]. However, unlike machine-printed documents, the segmentation of handwritten documents is still considered a challenging problem due to (i) irregular spacings between words and (ii) variations of writing styles depending on the person. According to ICDAR 2009 and 2013 handwriting segmentation contest results [2], [3], the text-line segmentation algorithms have been matured to some extent, however, there is still much room for improvements in the case of word segmentation methods.

A. Conventional Approaches for the Word Segmentation

For the word segmentation [4]–[14], document images are first segmented into text-lines [15]–[17]. Then, the word segmentation algorithm (for a single text-line) is applied to individual text-lines. Given a single text-line, the conventional word segmentation algorithms consist of two steps: the first step is to extract candidates for inter-word gaps (word-separator) and the next step is to classify the candidates into intra/inter-word gaps. For the candidate generation, a given text-line is represented with a set of super-pixels (where a super-pixel usually corresponds to a letter or a group of letters) and their gaps are considered candidates to be classified. This is a binary classification problem that assigns a label $\{0, 1\}$, where 0 means that the gap is an intra-word gap and 1 indicates it is an inter-word gap. For this classification, many algorithms have been developed: global/adaptive thresholding was used in [6]–[9], the unsupervised learning techniques such as clustering and Gaussian Mixture Model (GMM) were adopted in [10], [11] and the scale space selection approach was employed in [12]. Also, there have been researches using supervised-learning techniques such as neural networks [13], [14]. However, they only considered the local properties of individual gaps (without the considerations on correlations between the gaps).

B. Our Approach

Although the characteristics of inter-word gaps are changing across (and even in) documents, there are strong correlations (e.g., scale) between them in a text-line. However, it has been difficult to exploit these correlations in the conventional approaches, where the classification is made independently based on the properties of each gap. In order to alleviate these problems, we develop a novel framework that considers these correlations as well as local observations (i.e., the properties of each gap). To be precise, we formulate the word segmentation as an optimization problem that maximizes the similarity between inter-word gaps and the dissimilarity between inter-word and intra-word gaps, in addition to the likelihoods. Since this problem is a binary classification problem and the singleton and pairwise terms are only considered, it can be formulated as a binary quadratic problem, which can be efficiently solved with the Mixed-Integer Quadratic Programming (MIQP) solvers. Also, we estimate all parameters by adopting the structured learning framework [18], so that the proposed method can deal with a variety of inputs without user-defined parameters.

Experimental results on ICDAR 2009 and 2013 handwriting segmentation contest database [2], [3] show that proposed algorithm yields the best performance on ICDAR 2013 database and comparable performance to the state-of-the-art algorithm [7] on ICDAR 2009 database. Therefore, we believe that the main contributions of this paper are: (i) the novel formulation of the word segmentation problem into a binary quadratic problem and (ii) improved performances on challenging datasets. The rest of this paper is organized as follows. In Section II, we explain our framework for the word segmentation problem and present the cost function design. Section III describes the features used in

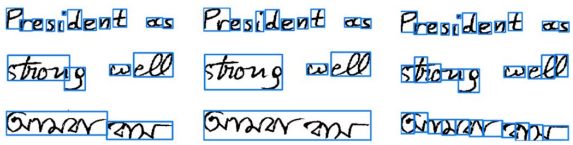
Manuscript received November 11, 2014; revised December 29, 2014; accepted January 05, 2015. Date of publication January 08, 2015; date of current version January 16, 2015. This work was supported in part by Samsung Electronics, and in part by the Ministry of Science, ICT and Future Planning, Korea, through the Information Technology Research Center support Program supervised by the National IT Industry Promotion Agency under Grant NIPA-2014-H0301-14-1019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tolga Tasdizen.

J. W. Ryu and N. I. Cho are with the INMC, Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea (e-mail: youjw@ispl.snu.ac.kr; nicho@snu.ac.kr).

H. I. Koo is with the Department of Electrical and Computer Engineering, Ajou University, Suwon, Korea (e-mail: hikoo@ajou.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2015.2389852



(a) (b) (c)

Fig. 1. Illustration of super-pixel representation methods for different scripts and writing styles: (a) results of CC-based representation, (b) results of OC-based representation, (c) results of proposed representation method [17].

the proposed method and presents the structured learning technique that yields the parameters of the cost function. Experimental results and conclusions are given in Section IV and V.

II. Our Formulation for Word Segmentation Problem

Like other conventional methods, we consider the word segmentation problem as a labeling problem that assigns a label (intra-word/inter-word gap) to each gap in a given text-line. Therefore, we first present our normalized super-pixel representation methods that extract a set of candidate gaps in each text-line. Then, we formulate the assignment problem as a binary quadratic problem, which allows us to consider pairwise relations as well as local properties.

A. Text-Line Segmentation and Super-Pixel Representation

For the text-line segmentation, we adopt the algorithm in [17], which is ranked the first in ICDAR 2013 handwriting segmentation contest [3]. After the text-line extraction, we represent each text-line with super-pixels. In the literature, two approaches for the super-pixel representation are available: (i) connected components (CCs) based representation [5]–[8], [14] and (ii) horizontally overlapping components (OCs) based representation [9]–[11]. However, we found that super-pixel representation based on the above criteria may miss some candidates in the case of the cursive writings. Moreover, the input of our formulation is a set of super-pixels in a text-line, and the size and the number of super-pixels should be consistent across its script and/or the writing style. Therefore, super-pixels from conventional methods that focused on the detection of inter-word gaps are not appropriate for our formulation. That is, even though conventional methods successfully detect inter-word gaps as shown in Fig. 1–(a) and (b), the proposed method prefers the representation in Fig. 1–(c) since our method considers intra-word gaps as well as inter-word gaps. To this end, we adopted the idea of normalized CCs in [17]: we estimate the average stroke width $\overline{W}$ in a document and split CCs so that they have normalized sizes (in terms of $\overline{W}$). With this method, we can have super-pixel representation results that reduce the effect of written languages, contents, and scanning resolutions as illustrated in Fig. 1.

B. Proposed Cost Function

Let us assume there are N gaps in a given text-line, where the i-th gap is denoted as x_i, and its label as $y_i \in \{0, 1\}$. Then, the word segmentation problem that assigns a binary label to each gap [6]–[11] can be posed as an optimization problem:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} E(\mathbf{x}, \mathbf{y}) \tag{1}$$

where $\mathbf{x} = \{x_i\}_{i=1}^{N}$ and $\mathbf{y} = \{y_i\}_{i=1}^{N}$.

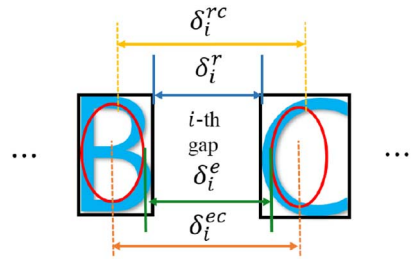


Fig. 2. Illustration of the distance features. We represent super-pixels with bounding boxes and ellipses [17], and four measures for gap-widths are employed.

In the proposed method, the cost function $E(\mathbf{x}, \mathbf{y})$ is designed to reflect pairwise correlations as well as unary properties. To be precise, in addition to unary terms (reflecting the individual likelihood of being either word-separator or not), we encode two additional observations: (i) inter-word gaps should have similar features and (ii) the features of inter-word gap and intra-word gap should be different. Therefore the cost function is given by a pseudo-boolean function

$$\sum_i (a_i y_i + b_i(1 - y_i)) + \sum_{i<j} (c_{i,j} y_i y_j + d_{i,j}(y_i \oplus y_j)) \tag{2}$$

where $\oplus$ is XOR operator. In (2), a_i/b_i is the cost for x_i being either a word-separator or not, $c_{i,j}$ is the cost when both x_i and x_j are word-separators, and $d_{i,j}$ is the cost when either x_i or x_j is a word-separator (the other is not a word-separator).

The cost function can be represented in a more compact way by assuming that the coefficients are linear functions of feature maps [18], [19]:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}) \rangle \tag{3}$$

where

$$\Psi(\mathbf{x}, \mathbf{y}) = \left[\sum_i \psi_u(x_i) y_i, \sum_{i<j} \psi_p(x_i, x_j) y_i y_j, -\sum_{i<j} \psi_p(x_i, x_j)(y_i \oplus y_j) \right]. \tag{4}$$

Here, $\psi_u(x_i)$ is the unary feature of x_i, and $\psi_p(x_i, x_j)$ is pairwise feature. Since $\psi_p(x_i, x_j)$ reflects similarity between x_i and x_j, it becomes large as the properties of two gaps are similar, and vice versa.

The optimization of the function (3) is a binary quadratic assignment problem, which is an NP-hard problem [20]. However, since the number of gaps is usually small (e.g., $N < 100$), the optimization can be considered a Mixed-Integer Quadratic Programming problem (MIQP)[21] and we can get an approximate solution with well-developed techniques such as the branch-and-bound method [22].

III. Structured Learning for Word Segmentation

For the word segmentation, the parameter $\mathbf{w}$ in (3) and feature maps should be determined. In this section, we discuss feature map selection and explain the adopted structured learning techniques [23].

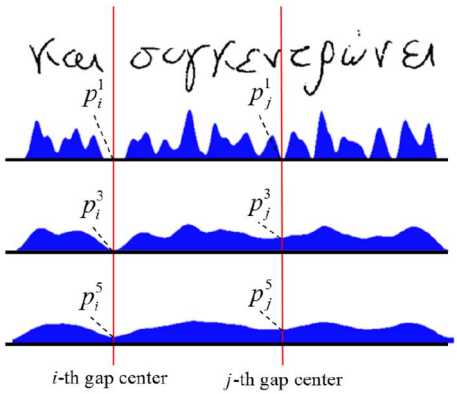


Fig. 3. Illustration of the smoothed projection profile features. The first row is a part of handwritten text-line and the other rows are Gaussian filtered projection profiles with different kernel sizes ($\overline{W}$, $3\overline{W}$, and $5\overline{W}$ respectively).

A. Feature Vector

In order to construct the feature map $\Psi(\mathbf{x}, \mathbf{y})$, we adopt the following features:

1) *Normalized distances between neighboring super-pixels ($\delta_i^r, \delta_i^{rc}, \delta_i^e, \delta_i^{ec}$)*: The most salient property for word-separators is its large width (compared with intra-word gaps), and conventional methods already exploited this feature [5], [7], [9], [10]. We also exploit this property, however, we employ 4 measures to represent the width of gaps. As shown in Fig. 2, they are boundary distances between rectangles/ellipses (denoted as δ_i^r and δ_i^e) and center-to-center distances of them (denoted as δ_i^{rc} and δ_i^{ec}). In order to achieve invariance to the capturing resolution, we normalize these distances with the estimated mean stroke width $\overline{W}$ as [17].
2) *Projection profile features ($p_i^n, n = 1, \ldots, 5$)*: The projection profile of a text-line is a one-dimensional array that shows the number of pixels for each horizontal position. Thus, the zero-run (the length of consecutive zeros) of projection profile has been exploited for the word segmentation of machine-printed documents [24]. However, in handwritten documents, zero-run features become less salient because letters in different words are likely to touch each other and the skew (or curve) of a text-line may corrupt the zero-run in the projection profile. In order to address these difficulties, we apply multiple Gaussian filters to projection profiles, where the kernel sizes are set to be proportional to the stroke width $\overline{W}$ (in order to achieve the invariance to scales, e.g., scanning resolution). Then, the value of filtered projection profile at the gap center is used as a feature, which is denoted as $p_i^n (n = 1, 2, \ldots, 5)$ respectively. We illustrate them in Fig. 3. Also, we normalize projected values with $\overline{W}$.
3) *Width ratio between current gap and the largest gap in a given text-line (r_i)*: Since the width ratio between a word-separator and the largest gap:

$$r_i = \frac{\delta_i^r}{\max_i \delta_i^r}. \tag{5}$$

is likely to be much larger than that of intra-word gaps, we also employ the ratio as a feature.

Based on above features, we define 10-dimensional unary feature $\psi_u(x_i)$ as

$$\psi_u(x_i) = [\delta_i^r, \delta_i^e, \delta_i^{rc}, \delta_i^{ec}, p_i^1, p_i^2, p_i^3, p_i^4, p_i^5, r_i] \in \Re^{10}. \tag{6}$$

For the pairwise feature map (that should reflect the similarity between two gaps), we adopt a term-wise squared difference as in [20]:

$$\psi_p(x_i, x_j) = -|\psi_u(x_i) - \psi_u(x_j)|^2 \in \Re^{10}. \tag{7}$$

Thus, the dimension of the proposed feature map $\Psi(\mathbf{x}, \mathbf{y})$ in (4) is 30.

Algorithm 1 Cutting Plane Algorithm [23]

Input: M training samples $\{(\mathbf{x}^n, \mathbf{y}^n)\}_{n=1}^M, C, \epsilon$
$\mathcal{S}_n \leftarrow \phi, \zeta_n \leftarrow 0, \forall n$
repeat
  for $n = 1$ to M **do**
    $\hat{\mathbf{y}} \leftarrow \arg\max_{\mathbf{y}} (\Delta(\mathbf{y}^n, \mathbf{y}) + \langle \mathbf{w}, \Psi(\mathbf{x}^n, \mathbf{y}) \rangle)$
    if $\Delta(\mathbf{y}^n, \hat{\mathbf{y}}) - \langle \mathbf{w}, \Psi(\mathbf{x}^n, \mathbf{y}^n) - \Psi(\mathbf{x}^n, \hat{\mathbf{y}}) \rangle > \zeta_n + \epsilon$ **then**
      $\mathcal{S}_n \leftarrow \mathcal{S}_n \cup \{\hat{\mathbf{y}}\}$
    end if
  end for
  $(\mathbf{w}^*, \zeta^*) \leftarrow \arg\min_{\mathbf{w}, \zeta'} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{n=1}^M \zeta_n'$
  subject to
  $\langle \mathbf{w}, \Psi(\mathbf{x}^n, \mathbf{y}^n) - \Psi(\mathbf{x}^n, \mathbf{y}') \rangle \geq \Delta(\mathbf{y}^n, \mathbf{y}') - \zeta_n', \mathbf{y}' \in \mathcal{S}_n, \zeta_n' \geq 0, \forall n$
  $\mathbf{w} \leftarrow \mathbf{w}^*, \zeta \leftarrow \zeta^*$
until no $\mathcal{S}_n$ has changed during iteration

B. Structured Learning

We find the parameter $\mathbf{w}$ with a structured learning technique. Given M training samples $\{(\mathbf{x}^n, \mathbf{y}^n)\}_{n=1}^M$ (M text-lines), we formulate n-slack Structured SVM [18] to estimate the optimal $\mathbf{w}$:

$$\min_{\mathbf{w}, \zeta} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{n=1}^M \zeta_n, \tag{8}$$

subject to

$$\langle \mathbf{w}, \Psi(\mathbf{x}^n, \mathbf{y}^n) - \Psi(\mathbf{x}^n, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}^n, \mathbf{y}) - \zeta_n, \forall n$$
$$\zeta_n \geq 0, \forall n. \tag{9}$$

For the loss function $\Delta(\mathbf{y}^n, \mathbf{y})$, we adopt Hamming distance defined as

$$\Delta(\mathbf{y}^n, \mathbf{y}) = \sum_i (y_i^n + y_i - 2y_i^n y_i). \tag{10}$$

Since the quadratic minimization problem (8) has an exponentially large number of constraints (9), we adopt the cutting plane algorithm [23] to solve (8) efficiently: the algorithm finds the most violated constraints by optimizing

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} (\Delta(\mathbf{y}^n, \mathbf{y}) + \langle \mathbf{w}, \Psi(\mathbf{x}^n, \mathbf{y}) \rangle). \tag{11}$$

TABLE I
EXPERIMENTAL RESULTS ON THE ICDAR 2009/2013 HANDWRITING SEGMENTATION CONTEST EVALUATION SET [2], [3]. SOME RESULTS ARE FROM [2], [3]. NUMBERS IN PARENTHESES ARE DIFFERENCES WITH THE TOP METHOD. †: 1ST IN 2013 COMPETITION, ‡: 1ST IN 2009 COMPETITION

	ICDAR 2013 [3]			ICDAR 2009 [2]			Average
	DR	RA	F-Measure	DR	RA	F-Measure	
GOLESTAN-a†	89.66% (0.84)	90.44% (1.11)	90.05% (0.98)	N/A	N/A	N/A	N/A
GOLESTAN-b	89.59% (0.91)	90.07% (1.48)	89.83% (1.20)	N/A	N/A	N/A	N/A
NCSR [11]	88.31% (2.19)	90.98% (0.57)	89.62% (1.41)	N/A	N/A	N/A	N/A
PAIS	N/A	N/A	N/A	91.83% (3.33)	89.29% (5.69)	90.54% (4.23)	N/A
ILSP [7]‡	87.93% (2.57)	88.37% (3.18)	88.15% (2.88)	**95.16%**	94.38% (0.60)	**94.77%**	91.46% (1.36)
LRDE	86.75% (3.75)	86.94% (4.61)	86.84% (4.19)	88.56% (6.60)	79.74% (15.24)	83.92% (10.85)	85.38% (7.44)
CUBS [25]	87.86% (2.64)	86.91% (4.64)	87.38% (3.65)	89.62% (5.54)	84.45% (10.53)	86.96% (7.81)	87.17% (5.65)
Proposed	**90.50%**	**91.55%**	**91.03%**	94.25% (0.91)	**94.98%**	94.61% (0.16)	**92.82%**

and tries to find the optimal $\mathbf{w}$. Note that the sub-problem that finds the most violated constraints can also be formulated as a binary quadratic assignment problem, since the Hamming loss function can be decomposed into the feature map [26]. The cutting plane algorithm is summarized in Algorithm 1.

IV. EXPERIMENTAL RESULTS

For the evaluation of the proposed algorithm, we conducted experiments on two publicly available database: ICDAR 2009 and ICDAR 2013 handwriting segmentation contest databases [2], [3]. ICDAR 2009 database is composed of 100 training images and 200 test images. Even though they were written by various writers, all images are Latin-based scripts. On the other hand, ICDAR 2013 database (consisting of 200 training images and 150 test images) seems to be a more challenging set as it includes Latin-based and Indian scripts. For the training of Structured SVM [18], we used each training set and the parameter C in (9) is set to 0.1. In order to solve binary quadratic assignment problems (3), (11), we adopt the MIQP solver by ILOG CPLEX [27].

For the objective evaluation, we adopt the measures defined in [2], [3]. To be precise, we use the `MatchScore` [28] is defined as

$$\texttt{MatchScore}(i,j) = \frac{|G_j \cap R_i|}{|G_j \cup R_i|}, \tag{12}$$

where G_j and R_i are two sets of pixels labeled as the i-th word by the algorithm and the j-th word by ground truth respectively, and $|\cdot|$ denotes the number of pixels in a set. A pair is considered a one-to-one match when the `MatchScore` is higher than 0.9. The performance metric F-measure (FM) is defined as

$$DR = \frac{o2o}{N}, RA = \frac{o2o}{M}, FM = \frac{2 \cdot DR \cdot RA}{DR + RA}, \tag{13}$$

where $o2o$ is the number of one-to-one matches, N is the number of words in the ground truth, and M is the number from the proposed algorithm.

Experimental results on both databases are summarized in Table I. As shown in the table, our method yields the largest F-measure on ICDAR 2013 database, while showing comparable performance to [7] on ICDAR 2009 database. We illustrate some examples in Fig. 4. As shown, our structured-learning based framework works well for a variety of inputs. We also evaluate the contribution of each feature by using a partial set of features. According to the experiments,

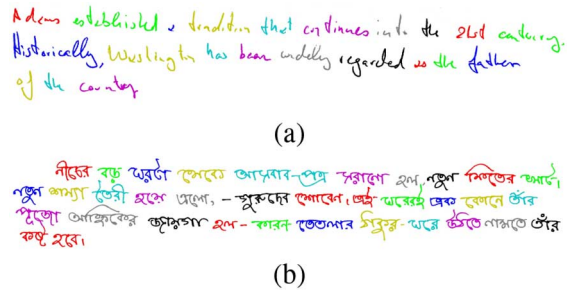


Fig. 4. Examples of proposed word segmentation results. (a) English script. (b) Traditional Indian(Bangla) script.

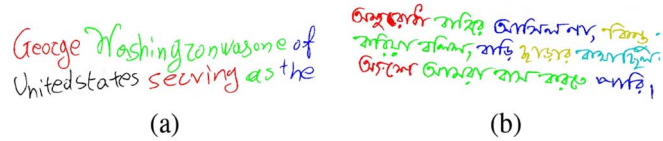


Fig. 5. Failure cases. (a) English script. (b) Traditional Indian(Bangla) script.

the performance decreases from 0.28% (without r_i) to 6.98% (without p_i^n), which shows that all the features have their discrimination powers. Some failure cases are illustrated in Fig. 5. As shown, our method does not work when intra/inter word gaps have similar properties. The proposed word segmentation method takes $1 \sim 2$ seconds to process a handwritten document having $10 \sim 20$ text-lines with an Intel Core i5 PC with our un-optimized C++ implementation. Including the text-line extraction algorithm [17], the total processing time is about $3 \sim 4$ seconds.

V. CONCLUSION

In this paper, we have proposed a word segmentation algorithm for handwritten document images. We formulated the segmentation problem as a binary quadratic programming and estimated the parameters with the structured learning method. Due to the proposed formulation, we could take into account the pairwise similarities between word-separators as well as unary properties in the word segmentation. Also, due to the Structured SVM, all parameters are estimated in a principled way and it is believed that our method can be easily extended to other databases. Experimental results on the ICDAR 2009/2013 handwriting segmentation databases have shown that the proposed word segmentation algorithm yields the state-of-the-art performances.

REFERENCES

[1] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1162–1173, Nov. 1993.

[2] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR 2009 handwriting segmentation contest," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2009, pp. 1393–1397.

[3] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," in *proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2013, pp. 1402–1406.

[4] R. Bozinovic and S. Srihari, "Off-line cursive script word recognition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 11, no. 1, pp. 68–83, Jan. 1989.

[5] U. Mahadevan and R. Nagabushnam, "Gap metrics for word separation in handwritten lines," in *proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 1995, pp. 124–127.

[6] G. Seni and E. Cohen, "External word segmentation of off-line handwritten text lines," *Patt. Recognit.*, vol. 27, no. 1, pp. 41–52, Jan. 1994.

[7] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis, "Handwritten document image segmentation into text lines and words," *Patt. Recognit.*, vol. 43, no. 1, pp. 369–377, Jan. 2010.

[8] T. Stafylakis, V. Papavassiliou, V. Katsouros, and G. Carayannis, "Robust text-line and word segmentation for handwritten documents images," in *proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 3393–3396.

[9] T. Varga and H. Bunke, "Tree structure for word extraction from handwritten text lines," in *proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2005, pp. 352–356.

[10] S. H. Kim, S. Jeong, G. S. Lee, and C. Y. Suen, "Word segmentation in handwritten Korean text lines based on gap clustering techniques," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2001, pp. 189–193.

[11] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," *Patt. Recognit.*, vol. 42, no. 12, pp. 3169–3183, Dec. 2009.

[12] R. Manmatha and J. L. Rothfeder, "A scale space approach for automatically segmenting words from historical handwritten documents," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1212–1225, 2005.

[13] G. Kim, V. Govindaraju, and S. Srihari, "A segmentation and recognition strategy for handwritten phrases," in *Proc. Int. Conf. Pattern Recognition*, 1996, pp. 510–514.

[14] S. Srihari, H. Srinivasan, P. Babu, and C. Bhole, "Handwritten Arabic word spotting using the cedarabic document analysis system," in *Proc. Symp. Document Image Understanding Technology*, 2005, pp. 123–132.

[15] F. Yin and C.-L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning," *Patt. Recognit.*, vol. 42, no. 12, pp. 3146–3157, Dec. 2009.

[16] H. I. Koo and N. I. Cho, "Text-line extraction in handwritten Chinese documents based on an energy minimization framework," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1169–75, Mar. 2012.

[17] J. W. Ryu, H. I. Koo, and N. Cho, "Language-independent text-line extraction algorithm for handwritten documents," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1115–1119, Sep. 2014.

[18] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, pp. 1453–1484, Sep. 2005.

[19] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proc. Int. Conf. Machine Learning*, 2004, pp. 104–111.

[20] T. Caetano, J. McAuley, L. Cheng, Q. V. Le, and A. Smola, "Learning graph matching," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048–1058, Jun. 2009.

[21] R. Lazimy, "Mixed-integer quadratic programming," *Math. Program.*, vol. 22, no. 1, pp. 332–349, Jan. 1982.

[22] B. Borchers and J. E. Mitchell, "An improved branch and bound algorithm for mixed integer nonlinear programs," *Comput. Oper. Res.*, vol. 21, no. 4, pp. 359–367, Apr. 1994.

[23] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, Oct. 2009.

[24] S. H. Kim, C. B. Jeong, H. K. Kwag, and C. Y. Suen, "Word segmentation of printed text lines based on gap clustering and special symbol detection," in *Proc. Int. Conf. Pattern Recognition*, 2002, pp. 320–323.

[25] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten Arabic text lines," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2009, pp. 176–180.

[26] S. Kim, S. Nowozin, P. Kohli, and C. Yoo, "Task-specific image partitioning," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 488–500, Feb. 2013.

[27] IBM ILOG CPLEX Optimizer [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[28] I. Phillips and A. Chhabra, "Empirical performance evaluation of graphics recognition systems," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 21, no. 9, pp. 849–870, Sep. 1999.