Text-Line Detection in Camera-Captured Document Images Using the State Estimation of Connected Components

Hyung Il Koo, *Member, IEEE*

***Abstract*— Camera-based text processing has attracted considerable attention and numerous methods have been proposed. However, most of these methods have focused on the scene text detection problem and relatively little work has been performed on camera-captured document images. In this paper, we present a text-line detection algorithm for camera-captured document images, which is an essential step toward document understanding. In particular, our method is developed by incorporating state estimation (an extension of scale selection) into a connected component (CC)-based framework. To be precise, we extract CCs with the maximally stable extremal region algorithm and estimate the scales and orientations of CCs from their projection profiles. Since this state estimation facilitates a merging process (bottom–up clustering) and provides a stopping criterion, our method is able to handle arbitrarily oriented text-lines and works robustly for a range of scales. Finally, a text-line/non-text-line classifier is trained and non-text candidates (e.g., background clutters) are filtered out with the classifier. Experimental results show that the proposed method outperforms conventional methods on a standard dataset and works well for a new challenging dataset.**

***Index Terms*— Document image processing, scene text detection, text-line detection, text-line segmentation.**

I. Introduction

As DIGITAL cameras and smartphones have become more widely available, camera-based text processing has attracted considerable attention, and many studies have been conducted for text-line detection in camera-captured images [1]–[8]. Text-line detection is an essential step for many document image processing tasks, such as optical character recognition (OCR), layout analysis, and pre-processing algorithms. However, many conventional methods focused on the scene text problem or constrained document cases, so that relatively little work has been performed on unconstrained camera-captured document images such as that shown in Fig. 1-(a) (see also the experimental section). For text-line detection in such inputs, we need to simultaneously address the challenges in natural (camera-captured) images and curved text-lines. However, most conventional methods focused on either case and the text-line detection in unconstrained inputs was not fully addressed. This is partially due to the lack of benchmark datasets for document cases: most benchmark datasets (for text processing in camera-captured images) consist of scene text images [1], [2], [9]. Although one dataset is available for document images [10], these images were of constrained environments. In other words, they are binary images capturing documents written in English, where the text-lines are roughly horizontal, the layouts are very similar, and there are no background clutters. Therefore, the set is not suitable for the evaluation of general text-line detection algorithms.

Considering the numerous methods that require text-line information in camera-captured images and possible OCR applications [10], [14], the need to develop a robust text-line detection method is an important challenge. In this paper, we address this problem by incorporating state estimation (an extension of scale selection [15]) into a connected component (CC)-based framework [16]. Our algorithm and dataset are publicly available on our website, http://ispl.snu.ac.kr/hikoo/curvedtext.

A. Proposed Method

The block diagram of the proposed method is shown in Fig. 1-(c). As shown, the method is based on the CC-based approach. First, in order to extract text components in natural images, we adopt the maximally stable extremal region (MSER) algorithm rather than conventional binarization algorithms [17]. This is because conventional binarization methods are designed to work for dark text on white background and have difficulties in handling multiple scales. After the CC extraction, we partition the extracted CCs into clusters, where each cluster corresponds to a text-line candidate. This step is called a text-line candidate generation [16], [18]. However, performing clustering without the knowledge of the scale and orientations is a very difficult task. In the work by O'Gorman [15], it was shown that scale selection is very beneficial in document processing. Similar to the work by O'Gorman, we estimate the states (scale and orientations) of CCs and build text-line candidates using these estimated states. The estimated states enable the candidate generation to be effectively performed for arbitrarily oriented (curved) text-lines and for a range of scales. Although our state estimation facilitates candidate generation, the bottom-up clustering usually yields a large number of false-positives on non-text regions (such as pictures in documents and background clutters). In order to filter them out, we also develop a

Manuscript received April 6, 2016; revised July 13, 2016; accepted August 26, 2016. Date of publication September 8, 2016; date of current version September 23, 2016. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea, Ministry of Science, ICT and Future Planning under Grant NRF-2014R1A1A1005698. The associate editor coordinating the review of this manuscript and approving it for publication was Prof Karsten Mueller.

The author is with the Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, South 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/TIP.2016.2607418

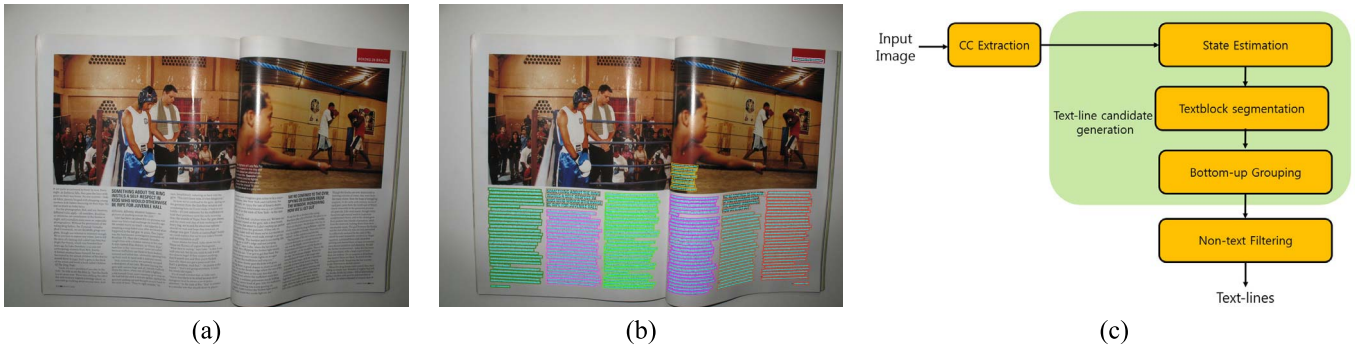


Fig. 1. (a), (b) Input and output of the proposed system (best viewed in electronic form). (c) Block diagram of the proposed system.

new text-line/non-text-line classification method based on the machine learning technique and a Markov Random Field (MRF) model. The MRF model enables us to consider the neighboring relations [16], [19].

B. Comparison to Our Previous Work

The proposed method can be considered as an extension of our previous work [20]. However, the proposed method has several improvements over the previous work. First, we extend the work by adopting a general CC-based framework so that we can deal with a variety of text/background colors and improve the invariance to scales. Second, we develop a new cost function in the state estimation. Our previous work estimated states by using the periodic patterns of text-lines; it therefore could not handle isolated text-lines. In this work, we propose a new cost function that can deal with isolated text-lines as well as textblocks (having more than one text-line). Third, we develop a systemic non-text filtering method. In the previous work, a heuristic idea using fitting errors was adopted for the classification. Since the previous method followed the assumptions that (a) dark texts are written on white background and (b) each textblock has multiple text-lines, the method did not severely suffer from false positives. However, since this simple method is not able to handle unconstrained inputs, we develop a machine-learning based method. Finally, we present a new dataset and evaluation protocol, and provide quantitative evaluation results.

II. Related Work

To the best of our knowledge, text-line detection in (unconstrained) camera-captured document images has not been discussed in the literature. However, the problem is closely related to many conventional problems. In this section, we review these related topics: curved text-line extraction, scene text detection, and handwriting segmentation.

A. Curved Text-Line Extraction

A number of methods have been developed to extract curved text-lines (segmentation) in binary images [22]–[25]. Most conventional methods were based on a bottom-up clustering approach, which builds characters, words, and text-lines progressively. In [22], clustering rules were developed by using the sizes of characters and distances between characters. In [23], the average height of characters was first estimated and text-lines were extracted with the estimated value. However, these methods were developed for roughly horizontal text-lines, while methods for arbitrarily oriented text-lines were proposed in [24] and [25]. Although they could handle more general cases, they were developed for binary images; using this approach in unconstrained inputs is therefore not straightforward. Essentially, many conventional methods were based on predefined merging rules and were prone to the over-segmentation and under-segmentation of text-lines.

Ridge-based text-line detection methods were proposed in [6] and [8]. In this approach, text-lines were extracted by detecting ridges in smoothed images. Although the method can be applied to gray images as well as binary images, the filtering kernel selection also relies on the assumption about the text-line orientation (usually, horizontal text-lines are considered). An approach that addresses the text-line segmentation by minimizing a cost function is proposed in [26]. While this approach alleviates the difficulties in devising ad-hoc clustering rules, the computational complexity of this active-contour-approach seems to be very high. A method to extract virtual baselines was recently proposed in [27]. The method is based on edge information and it works for natural images. However, the method could not provide the location information of text-lines and it is only able to handle roughly horizontal cases.

These curved text-line extraction algorithms have usually been evaluated on a dataset used in the Document Dewarping Contest in the CBDAR 2007 [10]. However, as shown in the first row of Fig. 2, the dataset consists of relatively similar images. Especially, the set consists of well-binarized images having roughly horizontal text-lines.

B. Scene Text Detection

The development of scene text detection algorithms enables many practical applications (such as camera-based translator, aids for the visually impaired, and robot navigation), and numerous methods have been proposed. These scene text detection algorithms can be classified into two groups: CC-based and region-based approaches. The CC-based approach generates text-lines by grouping CCs and the region-based approach uses a classifier of local patches to find text-lines. According to the recent competition results [2], [3], CC-based methods showed superior performance compared with region-based methods. Although many methods showed

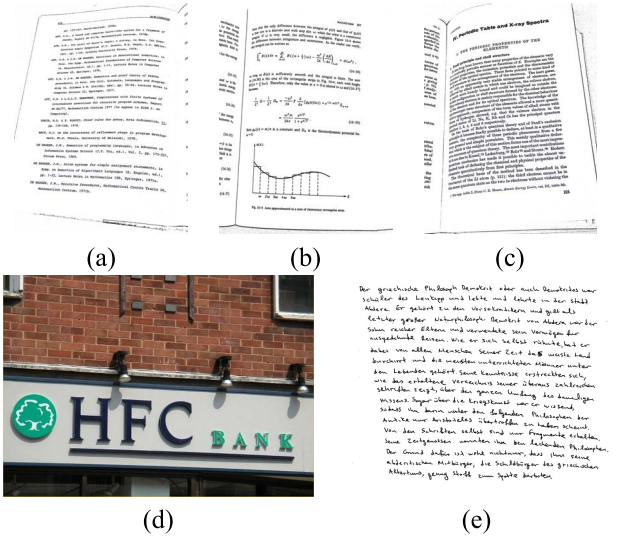


Fig. 2. (a), (b), (c) Images in curved text-line detection dataset [10], (d) Image in the scene text detection dataset [1], (e) Image in the handwriting segmentation dataset [21].

good performance in the ICDAR datasets [16], [18], they had difficulties in handling arbitrarily oriented text-lines (images in the competition sets contain roughly horizontal text-lines). In order to address these problems, more sophisticated CC-based methods were also developed [9], [28]. Although studies on scene text detection have successfully addressed many challenges in camera-captured images, they focused on the detection of short text-lines (usually words) written on planes. Therefore, conventional scene text detection algorithms cannot be applied to images that have curved text-lines such as that in Fig. 1-(a). Besides issues in text-line models (straight vs curved text-lines), the conventional methods did not exploit the distribution of CCs; their primary targets were sparse texts as shown in Fig. 2-(d) and the properties of individual CCs were considered promising features (e.g., shapes and stroke widths) [29]. However, in document images, text components are distributed regularly, and there is room for exploiting this new type of feature.

C. Handwriting Segmentation in Scanned Document Images

Handwriting segmentation is a problem to extract text-lines in scanned handwritten documents, as shown in Fig. 2-(e) [30], [31]. Although this problem involves many challenges (such as irregular character sizes, varying skews, and touching text-lines), the energy-based method in [31] showed good performance in the ICDAR 2013 text-line segmentation contest [21]. However, in this contest, the inputs were binary images consisting of only text pixels and the algorithm was able to focus on the partitioning of the pixels. Recently, gray images were used in the text-line extraction contest [32], however, there were no background clutters and non-text removal was not a critical issue.

III. Overview of the Proposed Method

As shown in Fig. 1-(c), the proposed method extracts CCs, estimates their states (scale and orientation), and uses

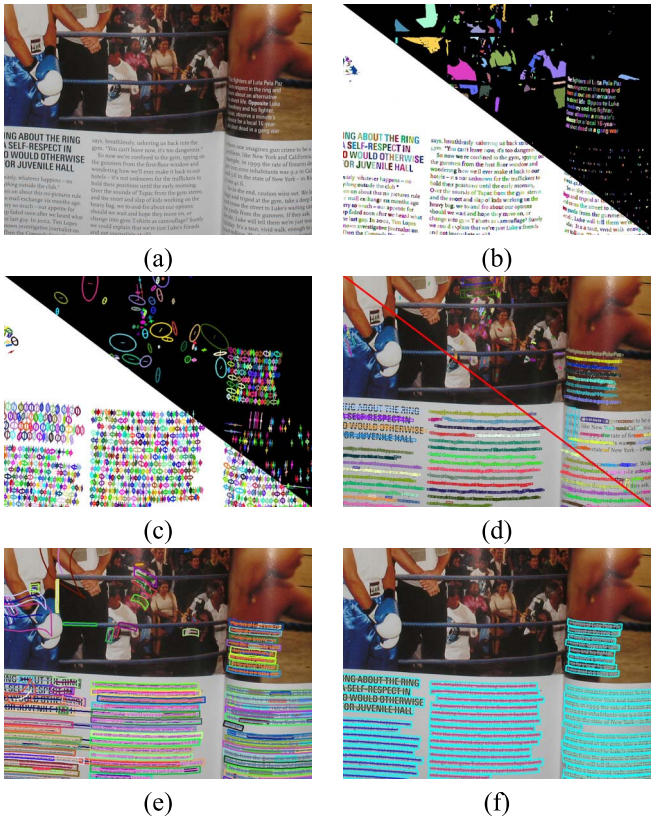


Fig. 3. Illustration of the proposed method (best viewed in electronic form): (a) Input image, (b) Extracted CCs, (c) Estimated states, (d) Bottom-up clustering process, (e) Text-line candidates, (f) Text-line results. From (b) to (d), the upper-right (resp. lower-left) part represents the processing result of bright (resp. dark) CCs.

that information in bottom-up clustering (text-line candidate generation). In particular, the estimated state provides a merging orientation/scale and the stopping criterion, and the proposed clustering method is able to handle complex cases robustly. This process (with intermediate results) is illustrated in Fig. 3. For a given image, we first extract CCs with the MSER algorithm, as shown in Fig. 3-(a) and (b). These extracted CCs are represented as superpixels (ellipses) and their state information is estimated by minimizing the proposed cost function. The cost function is designed to reflect the compactness of the projection profiles as well as the periodic patterns of text-lines. Our superpixel representation and the estimated states are illustrated in Fig. 3-(c). From the estimated states, we build text-line candidates with a bottom-up clustering method. An intermediate result for the clustering is shown in Fig. 3-(d). Since a large number of false-positives appear (especially, on non-text regions) in camera-captured images as shown in Fig. 3-(e), we apply a text-line classification method and obtain the final results. The classifier is based on machine learning techniques and an MRF model. Details of each step will be discussed in the following sections.

IV. CC Extraction and State Estimation

In this section, we present the CC extraction method and the proposed state estimation method, which is the former half part of the proposed method.

TABLE I

DISCRETE LEVELS OF INTERLINE SPACING ($s_p = \frac{N}{k}$). THE NUMBERS IN PARENTHESIS ARE REDUNDANT AND ARE NOT USED

	$k=5$	$k=4$	$k=3$	$k=2$
$N=64$	12.8	16.0	21.3	(32.0)
$N=128$	25.6	32.0	42.7	(64.0)
$N=256$	51.2	64.0	85.3	128.0

A. CC Extraction

To extract CCs, the MSER algorithm has been commonly used in recent scene text detection methods [17]. The MSER algorithm allows us to have text component candidates by extracting both fine and large structures at the same time. In [18], a variant of the MSER method was also proposed to resolve the overlaps between extracted CCs. We adopt this modified method and the extraction results are shown in Fig. 3-(b).

We denote a set of extracted CCs as

$$\mathcal{C} = \{c_p\}. \tag{1}$$

For each CC, we compute the center (x_p, y_p) and the covariance matrix Σ_p of the pixel positions. The covariance matrix is further represented with

$$\Sigma_p = \sigma_1 v_1 v_1^\top + \sigma_2 v_2 v_2^\top \tag{2}$$

by applying the eigenvalue decomposition [33], where σ_1 and σ_2 are eigenvalues ($\sigma_1 < \sigma_2$), and v_1 and v_2 are corresponding eigenvectors. With this decomposition, we represent $\{c_p\}$ with ellipses as shown in Fig. 3-(c): the minor and major axes are v_1 and v_2, respectively. This superpixel representation allows us to keep a set of CCs memory-efficiently.

B. Definition of States

The state of c_p is defined as a pair of two values:

$$f_p = (\theta_p, s_p) \tag{3}$$

where θ_p is the (local) orientation of a corresponding text-line and s_p is the interline spacing. For orientations, we quantize $[0, \pi]$ to $N_D = 32$ levels

$$\Theta = \left\{ \frac{k \times \pi}{N_D} \,\middle|\, k = 0, \cdots, N_D - 1 \right\}. \tag{4}$$

For scales, the proposed cost function uses the Discrete Fourier Transform (DFT) of projection profiles and 10 discrete scales are chosen in order to exploit the computational efficiency of Fast Fourier Transform (FFT) as listed in Tab. I:

$$\mathcal{S} = \{12.8, 16.0, 21.3, 25.6, 32.0, 42.7, 51.2, 64.0, 85.3, 128.0\}. \tag{5}$$

Although the above scale values are not uniformly distributed, $\mathcal{S}$ works well in practice. Rather, the problem of the above definition (i.e., s_p is an interline spacing) is that s_p is not well-defined for isolated text-lines such as that shown in Fig. 4-(c). However, fortunately, the scale selection is not critical for isolated text-lines, since the scale information is used for the following bottom-up clustering and the clustering will work for a range of scale values for isolated text-lines. Therefore, for isolated text-lines, the cost function should provide the correct orientation and the accurate scale estimation is relatively less important.

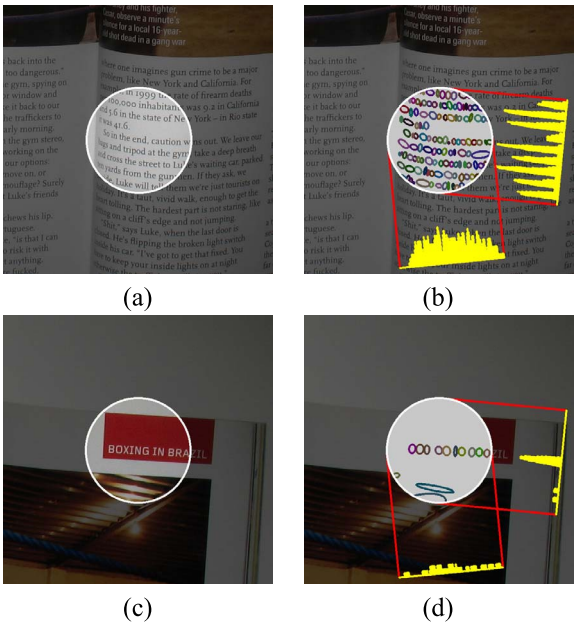


Fig. 4. Illustration of the proposed dataterm. (a), (b) Dataterm $V_p^{(1)}(\cdot)$ captures the periodic properties of multiple text-lines, (c), (d) Dataterm $V_p^{(2)}$ reflects the compactness of the projection profiles.

We also define the distance between two states as

$$\|f_p - f_q\| = \|s_p - s_q\|_{\mathcal{S}} + \|\theta_p - \theta_q\|_{\Theta} \tag{6}$$

where $\|s_p - s_q\|_{\mathcal{S}}$ is the index difference (from 0 to 9) in $\mathcal{S}$ and $\|\theta_p - \theta_q\|_{\Theta}$ is a circular index difference (from 0 to $N_D/2$). Note that the text-line orientation is invariant if we add $\pm\pi$ to the orientation.

C. Proposed Cost Function for the State Estimation

In this work, the state estimation is formulated as an optimization problem, of which the cost function is designed to (a) reflect local observations and (b) impose the smoothness constraint on states. The cost function is given by

$$E(\{f_p\}) = \sum_{c_p \in \mathcal{C}} V_p(f_p) + \sum_{(c_p, c_q) \in \mathcal{N}(\mathcal{C})} V_{p,q}(f_p, f_q) \tag{7}$$

where $\mathcal{N}(\mathcal{C})$ is a set of neighborhoods obtained by the Delaunay triangulation [34].

For the dataterm $V_p(f_p)$, we exploit the projection profile around c_p, which is defined as the number of ellipses on the projection directions as shown in Fig. 4-(b) and (d). More precisely, let us denote the profile around c_p as

$$x_{\theta,N}(n)(n = 0, \ldots, N-1), \tag{8}$$

where θ is the projection direction and N is the diameter of the corresponding region. Also, we denote the DFT of $x_{\theta,N}(n)$ as

$$X_{\theta,N}(k)(k = 0, \ldots, N-1). \tag{9}$$

From (8) and (9), we extract two features. The first feature captures the repeating patterns of text-lines. As shown in Fig. 4-(b), the projection profile shows periodicity for textblocks (having more than one text-line) when the projection direction coincides with the text-line orientation; also, this periodicity is well captured by computing the relative power of harmonic signals with a period, $s_p = \frac{N}{k}$:

$$\frac{|X_{\theta,N}(k)|^2 + |X_{\theta,N}(2k)|^2 + \cdots}{|X_{\theta,N}(0)|^2 + |X_{\theta,N}(1)|^2 + |X_{\theta,N}(2)|^2 + \cdots} \simeq \frac{|X_{\theta,N}(k)|^2}{|X_{\theta,N}(0)|^2} \quad (10)$$

as [20], and this observation is encoded into a data term

$$V_p^{(1)}(f_p) = -\log\left(\frac{|X_{\theta_p,N}(k)|^2}{|X_{\theta_p,N}(0)|^2}\right) \quad (11)$$

for $f_p = (\theta_p, \frac{N}{k})$. Although this term is effective in selecting orientation and scales, it cannot provide meaningful values for isolated text-lines (the profiles have no periodic components). For isolated text-lines as shown in Fig. 4-(c), we can exploit the compactness of the projection profiles. When the projection orientation coincides with the (local) text-line orientation, the projection profile becomes compact and

$$\frac{1}{N}|\{n|x_{\theta,N}(n) \neq 0\}| \quad (12)$$

becomes small, where $|\cdot|$ is the number of elements in the set. This observation can be represented as a new data term

$$V_p^{(2)}(f_p) = \log\left(\frac{|\{n|x_{\theta_p,N}(n) \neq 0\}|}{N}\right) \quad (13)$$

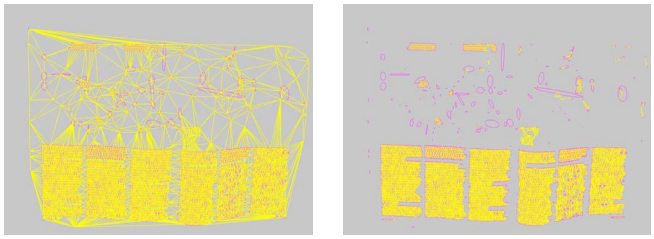
for $f_p = (\theta_p, \frac{N}{k})$. Since $V_p^{(2)}(f_p)$ does not depend on k, it will have the same value for several scales. For example, (13) has the same value for $s_p \in \{51.2, 64.0, 85.3, 128.0\}$ (the same holds for $\{12.8, 16.0, 21.3\}$ and $\{25.6, 32.0, 42.7\}$, respectively) as shown in Tab. I. However as discussed in the previous section, a range of scale values will work for isolated text-lines, provided that the orientation is correct. Finally, we use a linear combination of both terms for the dataterm in (7):

$$V_p(f_p) = \lambda V_p^{(1)}(f_p) + (1-\lambda)V_p^{(2)}(f_p) \quad (14)$$

(we set $\lambda = 0.5$). By combining these two complementary terms, our proposed dataterm can handle isolated text-lines as well as textblocks.

For the pairwise term in (7), we obtain a neighborhood system $\mathcal{N}(\mathcal{C})$ by applying the Delaunay triangulation to $\mathcal{C}$ and impose smoothness constraints for all neighboring pairs [34]. The neighborhood system $\mathcal{N}(\mathcal{C})$ is illustrated in Fig. 5-(a) and we adopt a smoothness term proposed in [20] for a neighboring pair $(c_p, c_q) \in \mathcal{N}(\mathcal{C})$:

$$V_{p,q}(f_p, f_q) = \mu(f_p, f_q) \times \exp\left(-\frac{\beta \times d_{pq}^2}{(s_p^2 + s_q^2)}\right) \quad (15)$$



(a) (b)

Fig. 5. Illustration of textblock segmentation: only dark CCs in Fig. 1-(a) are shown. (a) Delaunay triangulation results (neighborhood system), (b) Results after removing relatively long edges.

where $d_{pq} = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$ and

$$\mu(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ 0.4 & \|f_p - f_q\| \leq 3 \\ 5 & \text{otherwise.} \end{cases} \quad (16)$$

Since d_{pq} is the distance between two CCs, $d_{pq}^2/(s_p^2 + s_q^2)$ can be considered a normalized distance between two CCs (i.e., the value is invariant to the resolution of input images). Intuitively, the pairwise term (15) imposes smoothness constraints on adjacent sites, while allowing discontinuities for distant sites.

D. Inference

In order to minimize the proposed cost function (7), we adopt a standard combinatorial optimization method, i.e., the graph-cut method (the *expansion-move* algorithm) [19], [35]. Although there are a relatively large number of states (i.e., 32×10) for each CC, the algorithm works well. The estimated states are illustrated in Fig. 3-(c) with line segments, the lengths of which represent the scales $\{s_p\}$, and orientations are $\{\theta_p\}$. For an effective inference, the optimization was independently performed on the subsets of $\mathcal{C}$ (e.g., dark and bright CCs).

V. Text-Line Extraction

From the estimated states, text-line candidates are generated and the non-text-lines in the candidate set are filtered out using a trained classifier. Since text-lines can be more effectively extracted by applying a bottom-up clustering method to individual textblocks, we perform the textblock segmentation prior to the text-line candidate generation.

A. Textblock Segmentation

We can extract textblocks by removing the edges that are long when compared with the estimated scales:

$$d_{pq} \geq \epsilon \times \min(s_p, s_q), \quad (17)$$

and build clusters by partitioning $\mathcal{C}$ into

$$\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_L\} \quad (18)$$

as shown in Fig. 5 (we set $\epsilon = 2.0$). This idea works well in many cases; however, we find that this criterion has difficulties in handling close textblocks and unfolded book surfaces (due to perspective contraction)

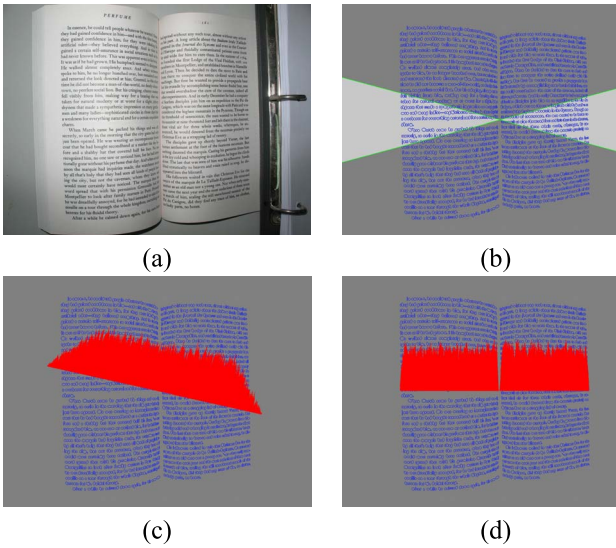


Fig. 6. Textblock segmentation: (a) The segmentation method using (17) has difficulties in handling unfolded book surfaces, (b) Search range $[\theta_{\mathcal{C}_i} - \Delta, \theta_{\mathcal{C}_i} + \Delta]$, (c), (d) Projection profiles for two orientations.

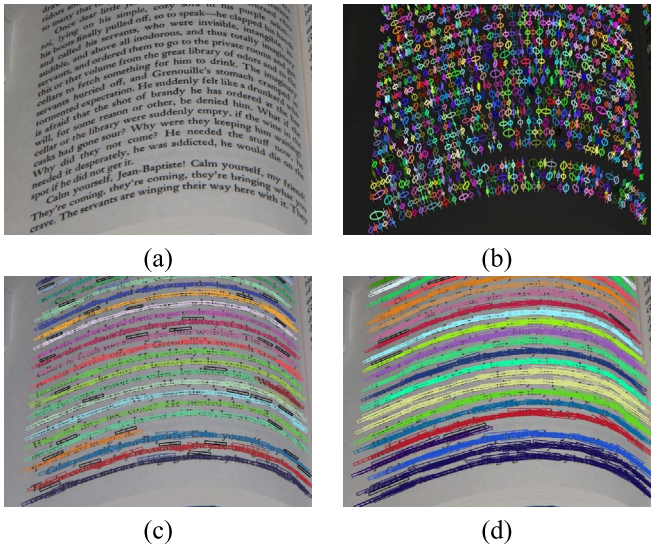


Fig. 7. Text-line candidate generation: (a) Zoom-in of Fig. 6-(a), (b) Estimated states, (c) Clustering with small scale (over-segmentation), (d) Clustering with large scale (under-segmentation). Each cluster is coded with different colors.

as shown in Fig. 5-(b) and Fig. 6-(a) respectively. In order to address this problem, a projection-profile-based method is also adopted [36]. That is, for each cluster $\mathcal{C}_i$, we first find the most frequent orientation of CCs in the cluster:

$$\theta_{\mathcal{C}_i} = \arg\max_{\theta} \left| \{ c_p \in \mathcal{C}_i | \theta_p = \theta \} \right|. \tag{19}$$

Given the most frequent orientation $\theta_{\mathcal{C}_i}$, we compute projection profiles in $\left[\theta_{\mathcal{C}_i} - \Delta, \theta_{\mathcal{C}_i} + \Delta\right]$ as shown in Fig. 6-(b). When there are zero-runs in the projection profiles as shown in Fig. 6-(d), the cluster $\mathcal{C}_i$ is partitioned into sub-clusters according to the zero positions. This segmentation method is recursively applied. However, in order to prevent the over-segmentation, we only apply this block segmentation step to large clusters. We set Δ to 15° for all experiments.

B. Text-Line Candidate Generation

For each textblock $\mathcal{C}_i \subset \mathcal{C}$, text-line candidates are generated by using a bottom-up grouping method [37]. The key idea is to draw a rectangle $ws_p \times hs_p$ for each CC, whose center is (x_p, y_p) and orientation is θ_p, and each connected region is considered to be a text-line candidate (to be precise, a candidate is a set of CCs). However, a single (w, h) cannot work for all cases; a small scale value results in over-segmentation as shown in Fig. 7-(c) and a large scale value yields under-segmentation as shown in Fig. 7-(d). In order to address this problem, we adopt a sequence of w in the clustering and also propose a stopping criterion.

To be specific, we first build initial candidates with $w_1 = 0.3$ and we iteratively refine this text-line candidate set by merging the elements (candidates) in the set. The merging condition is given as follows:

- Two candidates are connected with a new w_i value,
- A new candidate (the union of two candidates) is curvilinear.

We say a candidate $\mathcal{T}$ ($\subset \mathcal{C}$) is curvilinear when the centers of the CCs in $\mathcal{T}$ are well approximated with a k-th order polynomial:

$$\sqrt{\min_{f} \sum_{c_p \in \mathcal{T}} \frac{1}{|\mathcal{T}|} \left(y'_p - f(x'_p) \right)^2} \leq \frac{\bar{s}}{4} \tag{20}$$

where $\bar{s} = \frac{1}{|\mathcal{T}|} \sum s_p$ is the average scale and (x'_p, y'_p) is a rotated point of (x_p, y_p) so that points are spread across the x-axis. In experiments, we use an increasing sequence for w:

$$w \in \{0.3, 0.4, 0.5, 0.8, 1.0, 1.5, 2.0\} \tag{21}$$

and set $h = 0.15$. However, we find that the choice of w-sequence is not critical. The candidate generation results are shown in Fig. 3-(e). We use a fourth order polynomial in (20) and the estimated polynomial is called a text-line curve.

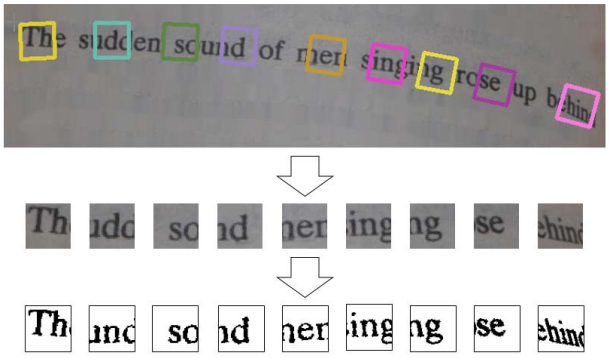


Fig. 8. Geometric and photometric normalization for the text-line confidence computation. For clear presentation, only a part of blocks are shown.

C. Computation of Text-Line Confidence

We compute the text confidence for each text-line and use the confidence values in the non-text-line filtering. Specifically, given a text-line candidate, we extract block-patches along the estimated text-line curve as shown in Fig. 8. For the geometric

and photometric normalization of these blocks, we transform each square block into a 64×64 block and the patch is binarized. The binarization method is the same as that in [16]; we consider the average color of CCs as the text color and the mean color of the whole patch as the background color. This procedure is illustrated in Fig. 8.

Given the normalized blocks, the text-line confidence was computed using the method in [16]; we train a text/non-text classifier for square patch inputs, so that the network yields $+1$ for text patches and -1 for non-text patches. We use 50-dimensional feature vectors. Specifically, for a given 64×64 block, the block is partitioned into four 64×16 horizontal slits. For each horizontal slit, we calculate (a) the number of white pixels, (b) the number of horizontal white-black transitions, (c) the number of horizontal black-white transitions, (d) the number of vertical white-black transitions, and (e) the number of vertical black-white transitions. We also compute their variances. A similar feature extraction is performed on four 16×64 vertical slits, so that we have 50 $(= 2 \times (4 \times 5 + 5))$ dimension feature vectors. For the training, we adopt artificial neural networks consisting of three layers (50-30-20-2), trained with the back-propagation algorithm [38]. We also tested the well-known Histogram of Oriented Gradients (HOG) feature [39] and the concatenation of the feature vector in [16] and HOG. However, the feature in [16] was simple and showed better performance. Finally, the text-line confidence for a candidate $\mathcal{T}$ is given by the average of the network outputs:

$$\psi(\mathcal{T}) = \frac{1}{|P|} \sum_{k \in P} (\text{the network output of the } k\text{-th patch}) \tag{22}$$

where P is a square patch set of the candidate $\mathcal{T}$ (as shown in the bottom of Fig. 8). The number of patches is chosen so that it is proportional to the number of CCs in $\mathcal{T}$.

D. Non-Text-Line Filtering

Given the text-line candidates of a textblock $\mathcal{C}_k$

$$\{\mathcal{T}_1, \mathcal{T}_2, \dots\}, \tag{23}$$

the non-text-line filtering can be formulated as a labeling problem that assigns labels to these text-line candidates. Let us denote the label of the i-th candidate as $l_i \in \{+1, -1\}$, where $+1$ implies that the candidate is a true text-line and vice versa. When the smoothness constraint (neighboring relations) is not imposed, the filtering can be performed with a threshold τ:

$$l_i = \begin{cases} +1 & \psi(\mathcal{T}_i) > \tau \\ -1 & \text{otherwise.} \end{cases} \tag{24}$$

Although this works (as shown in the experimental section), we can improve the performance by imposing the smoothness constraints on labels. In particular, we formulate this problem with MRF modeling

$$E(\{l_i\}) = \sum_i V_i(l_i) + \alpha \sum_{i,j} e_{i,j} \delta(l_i, l_j) \tag{25}$$

where $\delta(l_i, l_j) = 1$ when $l_i \neq l_j$, and $\delta(l_i, l_j) = 0$ otherwise. The dataterm $V_i(l_i)$ is based on the text-line confidence

$$V_i(l_i) = |P_i| \times \begin{cases} (\tau - \psi(\mathcal{T}_i)) & l_i = +1 \\ (\psi(\mathcal{T}_i) - \tau) & l_i = -1, \end{cases} \tag{26}$$

and the pairwise term is designed to reflect the sum of the edge strengths between two candidates

$$e_{i,j} = \sum_{a \in \mathcal{T}_i, b \in \mathcal{T}_j, (a,b) \in \mathcal{N}(\mathcal{C})} \exp\left(-\frac{\beta \times d_{pq}^2}{(s_p^2 + s_q^2)}\right) \tag{27}$$

(large values for adjacent text-line pairs). Since this is a binary labeling problem and the cost function is sub-modular, we can find a global optimum using the graph-cut method [19]. The choice of α and τ will influence the overall performance and will be discussed in the experimental section.

E. Non-Overlap Constraint

Finally, we impose the non-overlap constrain to resolve the overlaps between detected text-lines. In particular, $\mathcal{T}_i$ is filtered out when

$$\psi(\mathcal{T}_j) > \psi(\mathcal{T}_i) \tag{28}$$

$$|R(\mathcal{T}_i) \cap R(\mathcal{T}_j)| > 0.4 \times |R(\mathcal{T}_i)| \tag{29}$$

for some $\mathcal{T}_j$. Intuitively, this condition means that when there are significant overlaps between two text-lines, we remove the text-line that has a smaller confidence value.

VI. Experimental Results

In order to demonstrate the performance of the proposed method, we conducted extensive experiments. First, we evaluated the proposed method for the conventional dataset [10] and compared the performance with the existing methods. We then built a new dataset consisting of challenging camera-captured images and evaluated the performance on the set. In all experiments, we set $\beta = 0.125$. For the training of a text-line/non-text-line classifier, we used training images in [1], [2], and [9]. However, in order to improve the classification performance, we augmented the training set with several document images that had Asian scripts. Our executable file and the dataset are available on our website http://ispl.snu.ac.kr/hikoo/curvedtext.

A. Evaluation on a Conventional Set

We evaluated the proposed method on a conventional dataset [10], which consists of 102 images having $N_G = 3,091$ text-lines. For the evaluation, we adopt the pixel-based evaluation metric proposed in [40]. A weighted bipartite graph $G = (U, V, E)$ is constructed, where $U = \{G_i\}_{i=1}^N$ is a set of ground truth text-lines, $V = \{\mathcal{T}_j\}_{j=1}^M$ is a set of the detected text-lines, and E is a set of edges. The edge weight w_{ij} between G_i and $\mathcal{T}_j$ is given by the number of (text) pixels in the intersection area. After constructing the bipartite graph, an edge incident to each node is considered significant when

$$w_{ij} \geq \max(t_r \times P, t_a) \tag{30}$$

where P is the number of (text) pixels of a node, t_r is the relative threshold and t_a is the absolute threshold

TABLE II
EXPERIMENTAL RESULTS ON THE DATASET [10]. THE SYMBOL † REFERS TO THE BEST SETTING FOR THE DATASET PRESENTED IN SEC. VI-B

Algorithm	N_s	N_{fa}	N_{useg}	N_{oseg}	P_{ucomp}	P_{ocomp}	P_{mcomp}	P_{o2o}
Docstrum [15]	6852	6066	2096	4383	51.50 %	66.90 %	**0.00**(1) %	21.26 %
Neighborhood distance [23]	3256	4215	102	208	3.17(2) %	6.05 %	0.03 %	89.93 %
Rule-based method [22]	2924	**785**(1)	57	682	1.81 %	21.71 %	4.43 %	91.10 %
Ridge-based method [6]	3115	2183	110	144	3.30 %	4.40 %	0.29 %	89.65 %
Extended Coupled-Snakelets [26]	3106	3328	**51**(2)	**61**(2)	**1.58**(2) %	**1.84**(2) %	**0.00**(1) %	**95.12**(2) %
Proposed ($\tau = 0.60, \alpha = 0.06$)	3032	**1529**(2)	**17**(1)	**0**(1)	**0.550**(1) %	**0.00**(1) %	1.20 %	**97.70**(1) %
†Proposed ($\tau = 0.27, \alpha = 0.06$)	3002	**1405**(2)	**17**(1)	**1**(1)	**0.550**(1) %	**0.032**(1) %	2.17 %	**96.70**(1) %

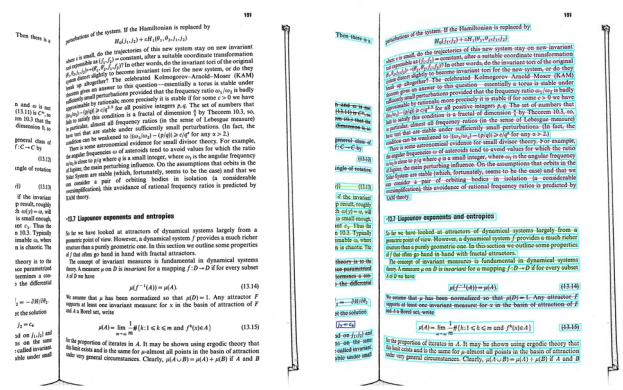
(we set $t_r = 0.1$ and $t_a = 100$ as [41]). Then, the vectorial score in [40] consists of the following quantities:

- N_s: the number of segments,
- N_{fa}: the number of false-positives,
- N_{useg}: the number of under-segmented components (the number of significant edges that $\{\mathcal{T}_j\}_{j=1}^{M}$ has minus M),
- N_{oseg}: the number of over-segmented components (the number of significant edges that $\{G_i\}_{i=1}^{N}$ has minus N),
- P_{ucomp}: $\frac{N_{ucomp}}{N_g}$ where N_{ucomp} is the number of detected text-lines having more than one significant edge,
- P_{ocomp}: $\frac{N_{ocomp}}{N_g}$ where N_{ocomp} is the number of ground truth text-lines having more than one significant edge,
- P_{mcomp}: $\frac{N_{mcomp}}{N_g}$ where N_{mcomp} is the number of missed ground truth text-lines,
- P_{o2o}: $\frac{N_{o2o}}{N_g}$ where N_{o2o} is the number of one-to-one matches.

Experimental results are summarized in Tab. II. As shown, the proposed method demonstrates good performance for several scores. Especially, the proposed method outperforms conventional methods in terms of P_{ucomp} and P_{ocomp}. That is, our method exploits the state information in the bottom-up clustering and generates more complete text-lines than those generated by conventional methods (which use predefined merging rules). As can be seen in Tab. II, many methods (including the proposed method) yield a large number of false-positives. It is mainly because the ground truth annotations only contain text-lines in the main page and text-line detection results on the other page are considered as false-positives. As shown in Fig. 9-(b), many correctly detected lines are classified as false-positives, and we believe that it is not suitable to compare N_{fa} directly. Compared with conventional methods, the proposed method shows relatively worse P_{mcomp} values (many missing text-lines), since the proposed method detects texts based on their distribution properties and the method sometimes misses pages numbers (usually consisting of 3 digits).

B. Building a New Dataset

Evaluations on the conventional dataset show that the proposed method compares favorably with conventional methods. However, the proposed method is developed to handle unconstrained camera-captured document images. In order to demonstrate the robustness of our method, we built a new dataset. The dataset consists of 86 images having



(a) (b)

Fig. 9. Input and output of the proposed method for the conventional set proposed in [10]. As shown, the proposed method detects text-lines on both pages (although text-lines on a left-hand-side page are classified as false positives).

Fig. 10. Example images in the proposed dataset. Recognizable text-lines are represented with polygons.

5,053 text-lines. In building the set, we tried to collect images of targets we typically encounter in our daily lives. As shown in Fig. 10, we also build ground truth annotation (corners of their bounding polygons) for each text-line and these annotations are stored in corresponding XML files. In building the ground truth, we instructed annotators to only draw boundaries for readable text-lines. However, for some ambiguous cases (usually, unfocused text on the background), we basically tried to annotate as many text-lines as possible.

C. Evaluation Metric

In addition to the dataset building, an evaluation metric should be developed. In conventional cases such as [10] and [21], binary images are given and the evaluation was performed with a pixel-based metric. However, the text-line detection in natural images is rather similar to the scene text detection problem (where the algorithm needs to find regions corresponding to texts) and we adopt the region-based

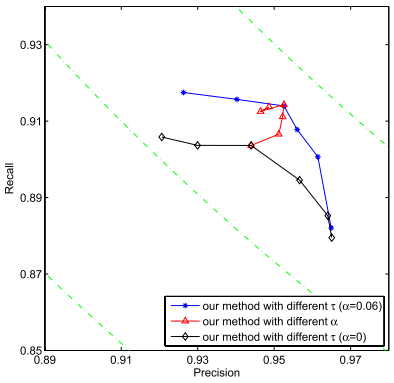


Fig. 11. Precision-recall curve. The parameter τ controls the precision and recall and the smoothness term ($\alpha > 0$) improves the performance (compared with text-line-wise decisions). However, α does not control the precision and recall in the same way as τ.

metric (intersection-over-union metric) [9], [42], [43]. Specifically, when a ground truth text-line G_i and a detected line $\mathcal{T}_j$ satisfies

$$\frac{|R(G_i) \cap R(\mathcal{T}_j)|}{|R(G_i) \cup R(\mathcal{T}_j)|} \geq \kappa, \tag{31}$$

the pair is considered to be a one-to-one match. Here, $R(G_i)$ is an annotated region for G_i (as shown in Fig. 10), and $R(\mathcal{T}_j)$ is a corresponding region for $\mathcal{T}_j$, which can be obtained by shifting the fitted polynomials in (20). We set a threshold κ to 0.6 (0.5 is used in [9]). Finally, the performance is measured with three quantities (precision, recall, and F-measure):

$$p = \frac{o2o}{M}, \quad r = \frac{o2o}{N}, \quad F = \frac{2pr}{p+r} \tag{32}$$

where $o2o$ is the number of one-to-one matches, N is the number of ground truth text-lines and M is the number of detected text-lines.

D. Evaluation on the New Dataset

We evaluated the proposed method on the new dataset and the precision-recall curve is shown in Fig. 11. The curve was plotted by changing α in (25) and τ in (26) respectively. Because parameter τ is a threshold used to determine whether or not a given candidate is a text-line, τ controls the precision and recall, while α controls the weight of the smoothness term. In other words, $\alpha = 0$ means that the classification is performed for each text-line independently, as in (24). As shown in Fig. 11, the smoothness term improves both the precision and recall ($\alpha > 0$). In terms of F, the proposed method shows the best performance when $\alpha = 0.06$: $p = 0.9526$, $r = 0.9139$, and $F = 0.9329$.

Four input and output pairs are shown in Fig. 12. The first and second columns show that the proposed method is able to handle relatively complex layouts and non-text regions, and it works for a range of scales. In practice, our method is able to handle $10 \sim 100$ pixel (character) heights. The performance degrades when handling blurred and/or sparse text-lines. As shown in the third column, textblocks located around the bookbinding suffer from out-of-focus blurs, and the proposed method fails to detect four textblocks around the bookbinding. Also, the proposed method exploits the distribution patterns of text components, and it shows difficulties in detecting text-lines that only have a small number of characters. We believe this means that the proposed method is rather complementary to the scene text detection problem [16], [18] that mainly focused on sparse (straight) text-lines. The last column in Fig. 12 shows the orientation-robustness of the proposed method. Although $(p, r) = (0.915, 0.901)$ is relatively low, a careful examination reveals that most of the focused text-lines are successfully detected.

E. Limitations of the Proposed Method

Although the proposed method is developed to handle a variety of inputs, it has several limitations. As shown in Fig. 13-(a) and (b), the method fails to handle tables in technical documents because periodic signals appear along both rows and columns in the tables, and the proposed method fails to estimate states [44]. Also, some text-lines are not well represented with low-order polynomials (e.g., 4-th order polynomials) as shown in Fig. 13-(c) and (d). Although the detection results in Fig. 13-(d) seem reasonably good, the quantitative result is $(p, r) = (0.25, 0.22)$ (only 2 one-to-one matches among 9 ground truth lines). In order to deal with such text-lines, we need to adopt more sophisticated models in the text-line representation. However, it should be noted that our polynomial models can cover most practical cases. In addition to the above problems, the method is based on the MSER algorithm and it also suffers from the limitations of MSERs: it has difficulties in handling cursive scripts and blurred texts.

We applied our method to other datasets (handwriting and page segmentation dataset [21], [45]), and the proposed method showed poor performance for some cases (such as Fig. 13-(e) and (f)): the method has difficulties in handling touching characters and complex layouts. Since these datasets consist of challenging images, we believe dataset-specific knowledge should be exploited in order to show the state-of-the-art performances (e.g., no skews in images from [45] and no non-text components in images from [21]). Also, we applied our method to the scene text detection dataset [2]. However, as discussed in the previous section, the proposed method did not show good performance for detecting sparse and short text-lines (please note that the proposed method was developed to handle camera-captured document images). Result images can be found in our project page.

F. Discussions on Scales

In the proposed method, we selected the scale level set $\mathcal{S}$ in (5) to exploit the FFT algorithm. However, the set $\mathcal{S}$ (roughly) covers all possible scales in [12.8, 128]. For example, the Viola/Jones detector [46] scans an image at 11 scales, where each scale is 1.25 larger than the previous scale, and this is similar to our scale level set $\mathcal{S}$. Moreover, the method works well for unconstrained camera-captured document images (showing diversities in resolutions, contents, fonts, and the number of columns).

The proposed method can be extended by using an image pyramid (we detect text-lines in multiple scales and integrate

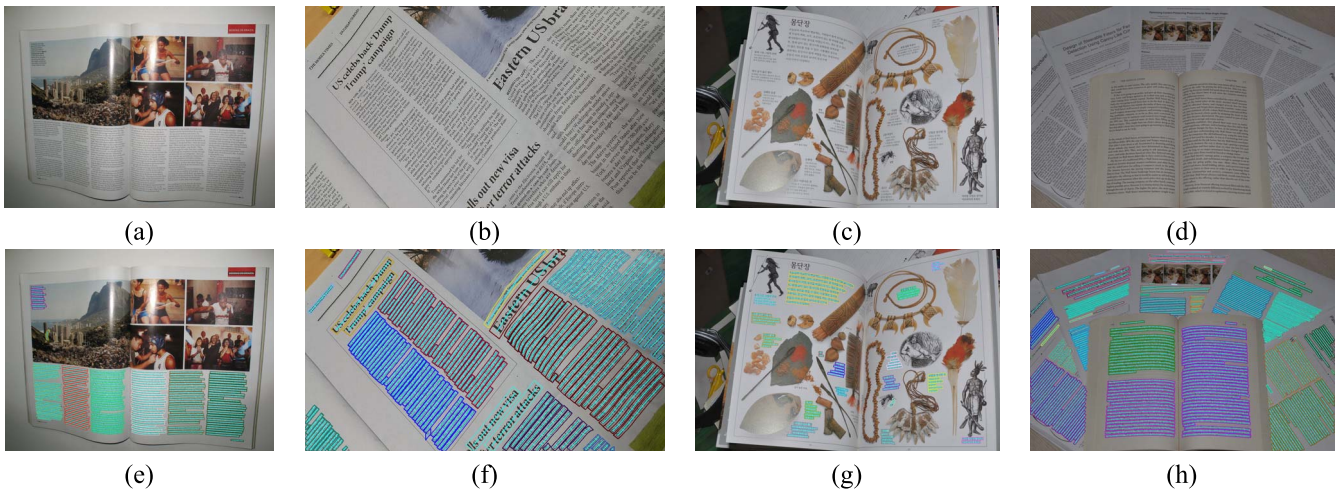


Fig. 12. Examples of input and output images (best viewed in electronic form). First column: $(p, r) = (0.995, 0.995)$, second column: $(p, r) = (0.939, 0.922)$, third column: $(p, r) = (0.885, 0.750)$, fourth column: $(p, r) = (0.915, 0.901)$.

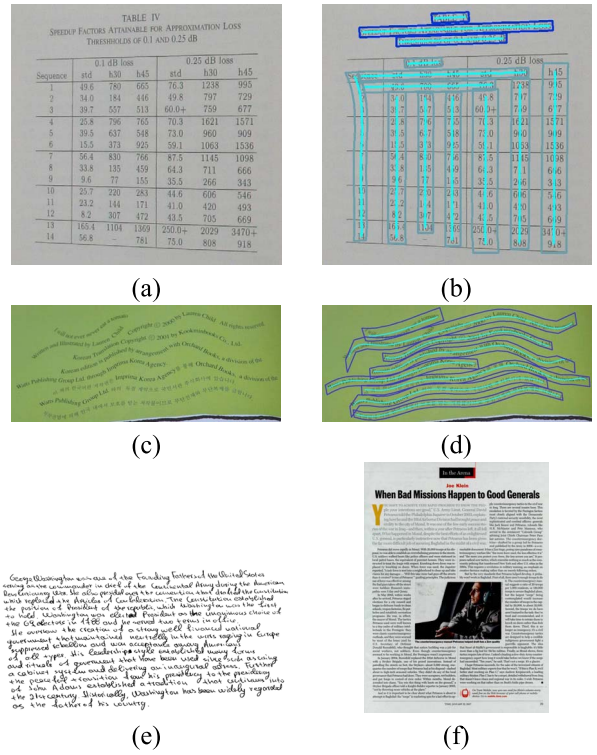


Fig. 13. Failure cases: the proposed method has difficulties in handling (a) tables in technical documents and (c) artificially curved text-lines. Also, the method fails in handling (e) handwritten documents having many touching components and (f) complex layouts.

them by using the confidence measure). As shown in Fig. 14, this idea allows us to detect headlines whose height is larger than 100 pixels (the headline height in Fig. 14 is about 200 pixels). However, the proposed method still has difficulties in detecting headlines having a small number of characters.

G. Computation Complexity

The proposed method takes an average of 10 seconds in handling 3264×2448 images that capture unfolded book surfaces as shown in Fig. 12-(a). The CC extraction takes about 1 second and the inference (the state estimation step) takes

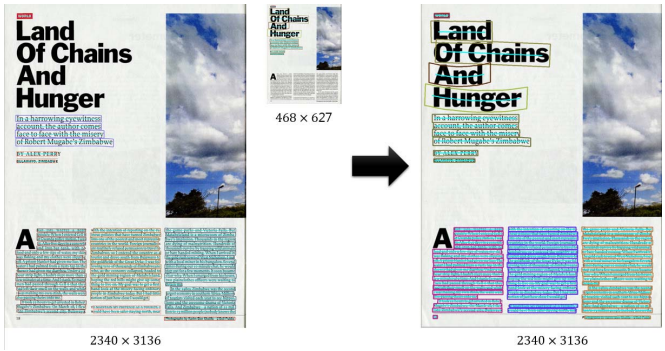


Fig. 14. By using an image pyramid, we can detect headlines as well as main texts.

$6 \sim 7$ seconds. However, the complexity of the inference step is proportional to the number of CCs, and the method takes less than 5 seconds in handling the image in Fig. 12-(c) (which has a relatively small number of CCs).

VII. Conclusion

In this paper, we proposed a text-line detection method for unconstrained camera-captured document images. To develop this method, we adopted scale-selection ideas in document processing and CC-based approach in the scene text detection problem. The proposed method extracted CCs with the MSER algorithm and built text-line candidates by using the bottom-up clustering method. Although the bottom-up clustering has been commonly used in the literature, we developed a new clustering method that can handle arbitrarily oriented text-lines in a scale-robust manner. We evaluated our method on the conventional dataset and our method compared favorably with conventional methods. Also, we built a new challenging dataset and evaluated the proposed method on the set. In addition, our dataset and ground truth are publicly available on our website.

References

[1] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, 2011, pp. 1491–1496.

[2] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2013, pp. 1484–1493.

[3] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.

[4] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Coupled snakelet model for curled textline segmentation of camera-captured document images," in *Proc. Int. Conf. Document Anal. Recognit.*, Jul. 2009, pp. 61–65.

[5] M. Diem, F. Kleber, and R. Sablatnig, "Text line detection for heterogeneous documents," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2013, pp. 743–747.

[6] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Ridges based curled textline region detection from grayscale camera-captured document images," in *Proc. 13th Int. Conf. Comput. Anal. Images Patterns*, Sep. 2009, pp. 173–180.

[7] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[8] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Towards generic text-line extraction," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2013, pp. 748–752.

[9] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1083–1090.

[10] F. Shafait and T. M. Breuel, "Document image dewarping contest," in *Proc. Int. Workshop Camera-Based Document Anal. Recognit.*, 2007, pp. 181–188.

[11] J. Liang, D. DeMenthon, and D. Doermann, "Geometric rectification of camera-captured document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 591–605, Apr. 2008.

[12] H. Cao, X. Ding, and C. Liu, "A cylindrical surface model to rectify the bound document image," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2003, pp. 228–233.

[13] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 785–792.

[14] B. S. Kim, H. I. Koo, and N. I. Cho, "Document dewarping via text-line based optimization," *Pattern Recognit.*, vol. 48, no. 11, pp. 3600–3614, 2015.

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

[16] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2296–2305, Jun. 2013.

[17] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2002, pp. 384–393.

[18] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.

[19] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[20] H. I. Koo and N. I. Cho, "State estimation in a document image and its application in text block identification and text line extraction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2010, pp. 421–434.

[21] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1402–1406.

[22] D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan, and M. Thielo, "A new method for text-line segmentation for warped documents," in *Proc. 7th Int. Conf. Image Anal. Recognit. (ICIAR)*, Póvoa de Varzim, Portugal, Jun. 2010, pp. 398–408.

[23] B. Gatos, I. Pratikakis, and K. Ntirogiannis, "Segmentation based recovery of arbitrarily warped document images," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, 2007, pp. 989–993.

[24] H. Goto and H. Aso, "Extracting curved text lines using local linearity of the text line," *Int. J. Document Anal. Recognit.*, vol. 2, no. 2, pp. 111–119, 1999.

[25] P. K. Loo and C. L. Tan, "Word and sentence extraction using irregular pyramid," in *Proc. 5th Int. Workshop Document Anal. Syst. V (DAS)*, Princeton, NJ, USA, Aug. 2002, pp. 307–318.

[26] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Coupled snakelets for curled text-line segmentation from warped document images," *Int. J. Document Anal. Recognit.*, vol. 16, no. 1, pp. 33–53, 2011.

[27] G. Meng, Z. Huang, Y. Song, S. Xiang, and C. Pan, "Extraction of virtual baselines from distorted document images using curvilinear projection," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 3925–3933.

[28] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, Sep. 2015.

[29] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2963–2970.

[30] T. Su, T. Zhang, and D. Guan, "Corpus-based hit-mw database for offline recognition of general-purpose chinese handwritten text," *Int. J. Document Anal. Recognit.*, vol. 10, no. 1, pp. 27–38, 2007.

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

[32] M. Murdock, S. Reid, B. Hamilton, and J. Reese, "ICDAR 2015 competition on text line detection in historical documents," in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 1–5.

[33] G. Strang, *Linear Algebra and Its Applications*. Orlando, FL, USA: Harcourt Brace Jovanovich, 1988.

[34] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry*. Berlin, Germany: Springer-Verlag, Feb. 2000.

[35] R. Szeliski *et al.*, "A comparative study of energy minimization methods for Markov random fields," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 16–29.

[36] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer*, vol. 25, no. 7, pp. 10–22, Jul. 1992.

[37] A. Delaye and K. Lee, "A flexible framework for online document segmentation by pairwise stroke distance learning," *Pattern Recognit.*, vol. 48, no. 4, pp. 1197–1210, 2015.

[38] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.

[39] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2005, pp. 886–893.

[40] F. Shafait, D. Keysers, and T. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 941–954, Jun. 2008.

[41] S. S. Bukhari, F. Shafait, and T. Breuel, "Performance evaluation of curled textline segmentation algorithms on CBDAR 2007 dewarping contest dataset," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 2161–2164.

[42] S. M. Lucas *et al.*, "ICDAR 2003 robust reading competitions: Entries, results, and future directions," *Int. J. Document Anal. Recognit.*, vol. 7, nos. 2–3, pp. 105–122, 2005.

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[44] W. Seo, H. I. Koo, and N. I. Cho, "Junction-based table detection in camera-captured document images," *Int. J. Document Anal. Recognit.*, vol. 18, no. 1, pp. 47–57, Mar. 2015.

[45] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos, "ICDAR 2009 page segmentation competition," in *Proc. Int. Conf. Document Anal. Recognit.*, Jul. 2009, pp. 1370–1374.

[46] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

Hyung Il Koo (S'09–M'10) received the B.S., M.S., and Ph.D. degrees from the Department of Electrical Engineering and Computer Science from Seoul National University, Seoul, South Korea, in 2002, 2004, and 2010, respectively. From 2010 to 2012, he was a Research Engineer with the Qualcomm Research Korea. He joined the Department of Electrical and Computer Engineering, Ajou University, in 2012, where he is currently an Associate Professor. His research interests include computer vision and machine learning.