State Estimation in a Document Image and Its Application in Text Block Identification and Text Line Extraction

Hyung Il Koo and Nam Ik Cho

INMC, Dept. of EECS, Seoul National University
hikoo@ispl.snu.ac.kr, nicho@snu.ac.kr

Abstract. This paper proposes a new approach to the estimation of document states such as interline spacing and text line orientation, which facilitates a number of tasks in document image processing. The proposed method can be applied to spatially varying states as well as invariant ones, so that general cases including images of complex layout, camera-captured images, and handwritten ones can also be handled. Specifically, we find CCs (Connected Components) in a document image and assign a state to each of them. Then the states of CCs are estimated using an energy minimization framework, where the cost function is designed based on frequency domain analysis and minimized via graph-cuts. Using the estimated states, we also develop a new algorithm that performs text block identification and text line extraction. Roughly speaking, we can segment an image into text blocks by cutting the distant connections among the CCs (compared to the estimated interline spacing), and we can group the CCs into text lines using a bottom-up grouping along the estimated text line orientation. Experimental results on a variety of document images show that our method is efficient and provides promising results in several document image processing tasks.

Keywords: document image processing, state estimation, graph cuts, text block identification, text line extraction.

1 Introduction

Text block identification and text line extraction are fundamentally important steps for OCR (Optical Character Recognition), and they are also essential for the rectification of camera-captured document images [1,2,3,4,5,6]. However, most research in this area has assumed scanned documents [1,7,8,9,10] and the applications to camera-captured images were limited to relatively simple layout and text-abundant cases [2,4,11]. In order to widen the area of valuable document processing tools (such as OCR and TTS for visually impaired, automatic translation of books and street signs, etc) to the camera-captured inputs, we propose a novel document state estimation algorithm and present its application in text block identification and text line extraction, where the state means line spacing, orientation, and other parameters describing the local properties of

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 421–434, 2010.

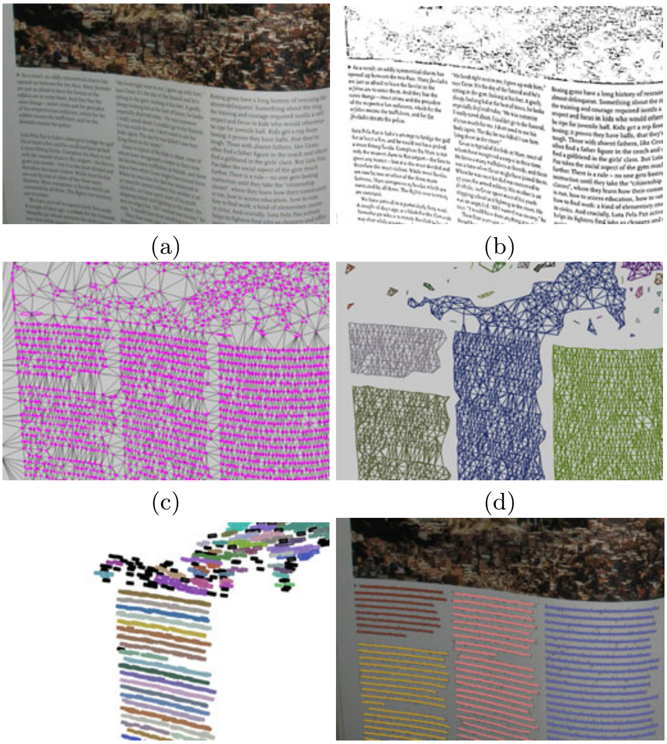
text region. Examples of input and output of our algorithm are shown in Fig. 1-(a) and (f). As can be seen, camera-captured images suffer from perspective distortion, geometric distortion, uneven illumination, motion blur, un-focussed blur, non-textual objects, and possibly cluttered background.

1.1 Our Method

Our method consists of two parts. In the former part, we estimate interline spacing and text line orientation for each Connected Component (CC), where we call two properties as the state of a CC. This step may correspond to a scale selection step in feature detection methods [12]. As the scale selection is important in detecting features from unknown measurement data, the state estimation is essential for unconstrained document image processing. For example, a simple problem to determine whether two adjacent CCs are in a same word or not may be ambiguous unless we know their states. Nevertheless, there is little research on this problem in camera based methods. It is probably because appropriate states for analysis may be known a priori in controlled situations [3,4,5]. However, we believe that the state estimation is an essential step for camera-captured image processing not only for the theoretical aspects but also for a practical system that can be demonstrated in uncontrolled environments (unknown character size, page curl, shot angle, and distance). In the latter part of our method, we develop a method that identifies text blocks and extracts text lines using the estimated states. Especially, the text line extraction method is based on a bottom-up grouping as commonly used in other related works [1,3,5,13]. However, unlike the other works, our method is largely free from conventional drawbacks due to the estimated states.

State estimation of CCs. The idea of assigning states can be found in the literature [1,9,10,15]. In docstrum [1], nearest neighbor (NN) angle histogram and NN distance histogram are computed from the geometric relationship between K-nearest units. From the histograms, they estimated the orientation, interline spacing, and within-line spacing. Then, a bottom-up approach is adopted to cluster CCs into words, text lines, and blocks. Due to the state estimation, the algorithm can effectively accomplish skew estimation and page segmentation [7], however, the method cannot handle the spatially varying cases [13]. It is because the method assumes fixed states (i.e., not spatially-varying) and the same rules using the same parameters are applied to the whole image. Related works can be found in [7,9,10,15]. Since camera-captured images, handwritten ones, and documents having complex layout have spatially varying properties, we assume that each CC has its own state. Also, we formulate the state estimation problem as an energy minimization problem. In designing the energy function, we consider a neighborhood system induced by Delaunay triangulation [14] and a data term is designed to exploit the periodical property of text lines.

Text block identification and text line detection. For text block identification and text line extraction, we first segment a graph formed by Delaunay



(a) (b)

(c) (d)

(e) (f)

Fig. 1. Illustration of our algorithm, (a) Input of our algorithm, (b) Binarization result, (c) Super-pixel representation and Delaunay triangulation [14], (d) Detected text blocks. See Section 3.1 for details, (e) Our bottom-up grouping result. See Section 3.2 for details, (f) Our final result.

triangulation of CCs (Fig. 1-(c)) into subgraphs (Fig. 1-(d)) by removing long edges that connect the CCs. Then, we cluster the CCs into text lines using a bottom-up approach. It is noted that the conventional bottom-up approaches are sensitive to input variations such as language, character size, and page curl [16,11] since they required heuristic rules, artificial parameters, and training process [5,13,3]. However, our method can be robust to the variations by using the estimated scale and orientation. Compared to recently developed text line extraction methods that adopt general image segmentation techniques [17,11], our method is more efficient and detects text lines in a scale/orientation invariant manner.

However, like other methods, our method also suffers from non-textual objects as shown in Fig. 1-(e). For non-textual object rejection, a training based method was proposed in [8] for classifying each block into printed text, handwriting, or noise. Although their results are very convincing, it is not clear how to construct a training set that achieves robustness to language variation, poor image quality, and complex layout. Since the noise that smears text region [7,8] is seldom observed in camera-captured images of printed material, we assume that non-textual objects take place distant from text blocks. Then we can reject non-textual objects using the properties of clusters. Precisely, we assume that (1) a cluster in text region tends to be curvilinear, and (2) a cluster in non-textual region is isolated (represented as black rectangles) or it may be non-curvilinear as can be seen in Fig. 1-(e). Using these properties, we formulate non-textual object rejection as a labeling problem with an energy minimization approach. After the inference, we remove non-textual objects, and refine text blocks and text lines. The result is illustrated in Fig. 1-(f).

2 The State Estimation of CCs

In this section, we explain our state estimation method based on an energy minimization framework. This section consists of binarization, CC construction, energy formulation, and its minimization that gives the state of each CC.

2.1 Binarization and CC Construction

The first step of our algorithm is the binarization of a gray image I. Our binarization method is based on the retinex filtering which is efficient and robust to uneven illumination:

$$B_s = \begin{cases} 1 & I_s < \mu_1 \times G_s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where I_s is the intensity at pixel s, $G(\sigma) * I$ is a Gaussian filtered image of I, and $G_s = (G(\sigma) * I)_s$ [18]. However, since it produces a number of spurious responses on dark and homogeneous region, we introduce an additional condition that suppresses responses on homogeneous region:

$$|I_s - G_s| > \mu_2 \tag{2}$$

for $B_s = 1$. From the binary image $\{B_s\}$, we extract CCs of '1' using an 8-neighborhood system. In this process, we suppress small CCs (containing less than 10 pixels) and large CCs (containing more than 3000 pixels) for the removal of noisy ones. We denote the set of extracted CCs as $\mathcal{P}$.

2.2 Energy Formulation

We assign a state to every site $p \in \mathcal{P}$, and denote the state as $f_p = (s_p, \theta_p)$, where s_p is the interline spacing between neighboring text lines and θ_p is the orientation of a text line where p belongs. The estimation problem is formulated as an energy minimization problem whose energy function is given by

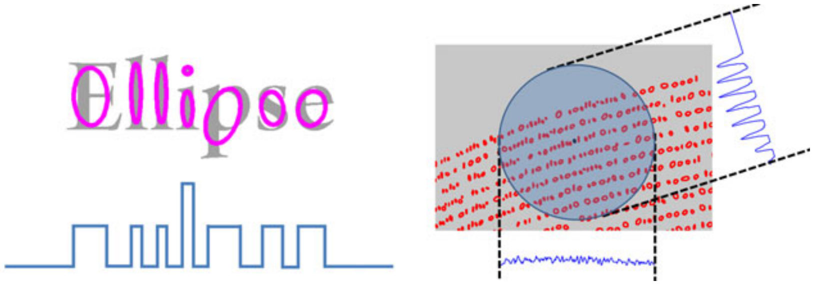
$$E(\{f_p\}) = \sum_{p \in \mathcal{P}} V_p(f_p) + \sum_{(p,q) \in \mathcal{E}} V_{p,q}(f_p, f_q) \tag{3}$$

where $V_p(f_p)$ is a data term reflecting local observation, $V_{p,q}(f_p, f_q)$ is a pairwise potential reflecting label smoothness, and $\mathcal{E}$ is a set of edges.

2.3 The Design of $V_p(f_p)$

For the design of $V_p(f_p)$, we first explain our projection method, and frequency domain analysis will be followed.

Super-pixel approximation. Since pixel based approaches are computationally demanding in analyzing local patterns, we reduce complexity by using super-pixels. Precisely, we compute the mean vector (x_p, y_p) and the covariance matrix Σ_p of pixels in the p-th CC. The covariance matrix is decomposed into $\Sigma_p = \sigma_1 v_1 v_1^T + \sigma_2 v_2 v_2^T$ where $\sigma_1 > \sigma_2$ are eigenvalues, v_1 and v_2 are eigenvectors. Using the decomposition, the CC is approximated to an ellipse (whose minor and major axes are v_1 and v_2 respectively) as illustrated in Fig. 2-(a). In this process, ellipses showing large eccentricity ($\frac{\sigma_1}{\sigma_2} > 15$) are also removed. Then we define a projected signal, which is the number of ellipse on the line of projection as illustrated in Fig. 2-(a). Fig. 2-(b) shows an example of projecting the super-pixels in a circle into some directions.



(a)

(b)

Fig. 2. (a) Ellipse approximation of CCs and its projection, (b) Ellipse approximation of CCs and its projected signals into two directions

Data term based on frequency domain analysis. As illustrated in Fig. 2-(b), when CCs around a site p are projected to the normal direction to a text line, a periodic pattern (whose period is the interline spacing s_p) is observed. From the observation, we design $V_p(f_p)$ so that it decreases as the periodicity of projected signal is increasing. For this, we first obtain a projected signal $x(n)$ by projecting CCs to the orientation of θ_p, and its DFT $X_N(k)$ is computed: $X_N(k) = \sum_{n=0}^{N-1} x(n) \exp\left(-j\frac{2\pi kn}{N}\right)$. The normalized energy of a signal of period $\frac{N}{k}$ is given by

$$\frac{|X_N(k)|^2 + |X_N(2k)|^2 + \cdots}{|X_N(0)|^2 + |X_N(1)|^2 + |X_N(2)|^2 + \cdots} \simeq \frac{|X_N(k)|^2}{|X_N(0)|^2} \tag{4}$$

where the numerator of left hand side is the energy of repeating component ($T = \frac{N}{k}$) and the denominator is the overall energy of $x(n)$ [19]. Moreover, we verified through experiment that this can be replaced as the magnitude of first harmonic over the DC term as shown in the right-hand side of (4). Based on this measure of periodicity, $V_p(f_p)$ is defined as

$$V_p(f_p) = -\log \frac{|X_N(k)|^2}{|X_N(0)|^2}. \tag{5}$$

Finally, we have to choose (s_p, N_p, k_p) satisfying $\frac{N_p}{k_p} = s_p$. There are several factors to be considered. For the good localization in frequency domain, a large N is desirable. On the other hand, a large N is not good at handling spatially varying states. We also have to consider computational complexity. Considering these factors, we select 10 scales from $12.8 \leq s_p \leq 128$ and they are summarized in Table 1. We also quantize orientations into $D = 32$ steps:

$$\theta_p \in \left\{ i \times \frac{\pi}{D} \middle| i = 0, 1, \ldots, D-1 \right\}. \tag{6}$$

In summary, $V_p(f_p)$ for label $f_p = (s_p, \theta_p)$ is computed as follows.

- CCs around the p-th CC are projected to the line whose orientation is θ_p, resulting $x(n)$. In the projection, the size of window is determined according to the Table 1.

Table 1. Discrete levels of interline spacing (s_p) used in our algorithm

Discrete levels (s_p)	12.8	16.0	21.3	25.6	32.0	42.7	51.2	64.0	85.3	128.0
N_p	64	64	64	128	128	128	256	256	256	256
k_p	5	4	3	5	4	3	5	4	3	2

- When there are only small number of CCs (i.e., $x(n) \leq 3$ for all n) or $|X_{N_p}(k_p)|$ is not a local maximum, we set $V_p(f_p) = \epsilon$.
- Otherwise, the data cost is given by

$$V_p(f_p) = -\log \frac{|X_{N_p}(k_p)|^2}{|X_{N_p}(0)|^2}. \tag{7}$$

2.4 Pairwise Potential

For a neighborhood system, we adopt Delaunay triangulation [14] and our pairwise potential is given by

$$V_{p,q}(f_p, f_q) = \mu(f_p, f_q) \times \exp\left(-\frac{k \times d_{pq}^2}{(s_p^2 + s_q^2)} \right) \tag{8}$$

where d_{pq} is the Euclidean distance between the site p and q. Since $d_{pq}/\sqrt{s_p^2 + s_q^2}$ can be considered as an intrinsic distance (i.e., invariant to camera settings, distance between documents and camera, and so on) between two CCs, the cost function allows label discontinuities between distant sites while imposing smoothness constraints on nearby ones. Moreover, in order to allow small amount of label discontinuities (which is common in camera-captured documents), $\mu(f_p, f_q)$ is defined as

$$\mu(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ \lambda_1 & |f_p - f_q| \leq 3 \\ \lambda_2 & \text{otherwise} \end{cases} \tag{9}$$

where $|f_p - f_q|$ is the label distance defined as the sum of orientation difference and scale difference ($\lambda_1 < \lambda_2$).

2.5 Optimization

In (3), the number of sites ($|\mathcal{P}|$) is usually up to $20,000$ and the number of labels is 32×10. We optimize the cost function using *Expansion move* algorithm [20].

3 Text Block Identification and Text Line Extraction

In this section, we explain the latter part of our algorithm. From the estimate states, (1) we segment a document image into blocks by removing long edges, (2) each block is decomposed into clusters, and (3) we reject non-textual clusters by considering the curvilinearity of each cluster and neighboring relations. Finally, (4) we refine text blocks and text lines.

3.1 Page Segmentation

For page segmentation, we remove perpendicular edges (which are perpendicular to text lines) satisfying $d_{pq} \geq \epsilon_1 \times \min(s_p, s_q)$, and we remove parallel edges satisfying $d_{pq} \geq \epsilon_2 \times \min(s_p, s_q)$. Since the edges connecting two vertically adjacent regions are usually longer than edges connecting horizontally adjacent regions, we can achieve more accurate segmentation by considering orientation as well as interline spacing. Two constants are determined according to conventional layout: $(\epsilon_1, \epsilon_2) = (1.2, 0.9)$.

However, this method may suffer from perspective contraction, coarse quantization of interline spacing, and noise. That is, a CC on text region and another CC on a picture region may be linked as Fig. 1-(d). Therefore, non-textual object rejection should be applied. Since non-textual object rejection is closely related with our bottom-up grouping method, we explain the method in the next section and the explanation on non-textual object rejection will be followed.

Skew correction. After page segmentation, we first find the dominant angle of a text block by using a voting method and compensate the skew in order to represent a text line as a form of $y = f(x)$ without numerical instability.

3.2 Bottom-Up Grouping

For grouping, we draw a rectangle for each CC, whose size is $ws_p \times hs_p$, its center (x_p, y_p), and rotated by θ_p(+ text block skew). Then, each connected region corresponds to a word or a text line as can be seen in Fig. 3. However, a single choice of (w, h) is not adequate. When small (w, h) is used, a text line may be partitioned into several clusters (over-segmentation of a text line) as Fig. 3-(a). On the other hand, more than one line may be merged into a single cluster (under-segmentation of a text line) when large (w, h) is used as shown in Fig. 3-(b). Therefore, we develop a method that incrementally increases w value (fixing $h = 0.25$). First, we group CCs into clusters using $w_1 = 0.8$, resulting a set of clusters $\mathcal{W}$. Then, two clusters C_i, $C_j \in \mathcal{W}$ are merged into a new cluster (i.e., C_i and C_j in $\mathcal{W}$ are replaced with $C_i \cup C_j$) when three conditions are satisfied:

1. Two clusters are connected when a new w_i is used.
2. The overlap of two supports (x-domain) is less than 10% of their length.
3. A new cluster $(C_i \cup C_j)$ is still a curvilinear one (the detailed explanation of this condition will be followed in the next section).

Intuitively, the second and third conditions prevent the merging of neighboring text lines. We use $w_2 = 1.0$ and $w_3 = 1.2$. Fig 1-(e) shows our bottom-up grouping result.

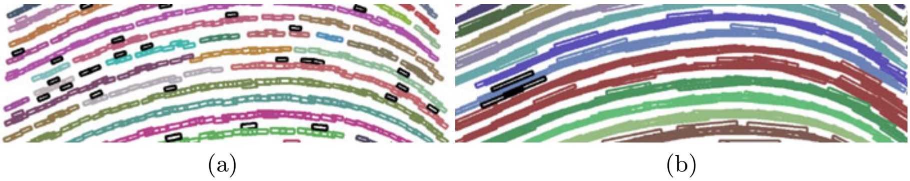

(a) (b)

Fig. 3. (a) When we use small (w, h), a text line can be segmented into several small clusters (over-segmentation), (b) If we use large (w, h), more than one line can be merged into a single cluster (under-segmentation)

3.3 Curvilinearity Measure

For the curvilinearity measure of a cluster $C \in \mathcal{W}$, we define the fitting error of C and the scale of C. The fitting error of C is defined as

$$\eta(C) = \sqrt{\frac{1}{|C|} \times \min_f \sum_{p \in C} \left| y'_p - f(x'_p) \right|^2} \tag{10}$$

where $|C|$ is the number of CCs in C, the degree of polynomial f is determined according to $|C|$ (from first to fourth order polynomials), and (x'_p, y'_p) is the rotated point of (x_p, y_p) by the text block skew. Also the scale of C is given by

$$s(C) = \frac{1}{|C|} \sum_{p \in C} s_p. \tag{11}$$

Since $\eta(C)/s(C)$ can be considered as a normalized fitting error, we can measure the curvilinearity by comparing $\eta(C)$ and $s(C)$. For example, $\eta(C) \ll s(C)$ means C is a curvilinear cluster. Experiment results show that most of text lines satisfy $\eta(C) < 0.2 \times s(C)$ and we say that C is curvilinear when it satisfies the inequality.

3.4 Textual/Non-textual Cluster Labeling

Although the proposed curvilinearity test (i.e., $\eta(C) < 0.2 \times s(C)$) provides a good rule to reject non-textual clusters, the performance can be improved by considering neighboring relations. We also formulate the problem as an energy minimization problem. From $\mathcal{W}$, we construct a new graph where each site is an element in $\mathcal{W}$, and denote its label $l_i = 0$ when $C_i \in \mathcal{W}$ is a part of text region, and $l_i = 1$ otherwise. Our energy function is given by

$$E(\{l_i\}) = \sum V_i(l_i) + \lambda_3 \sum e_{ij}\delta(l_i, l_j) \tag{12}$$

where

$$\delta(l_i, l_j) = \begin{cases} 1 \; l_i \neq l_j \\ 0 \; l_i = l_j . \end{cases} \tag{13}$$

In defining $V_i(l_i)$, we consider two properties that (1) an isolated C_i ($|C_i| \leq 5$) is likely to be a non-textual object and (2) a non-curvilinear C_i is likely to be a non-textual object. Therefore, when $|C_i| \geq 6$, our data term is given by

$$V_i(l_i) = |C_i| \times \begin{cases} \eta(C_i) & l_i = 0 \\ 0.2 \times s(C_i) & l_i = 1 . \end{cases} \tag{14}$$

The pairwise term in (12) is derived from (8), and it is given by

$$e_{ij} = \sum_{p \in C_i, q \in C_j, (p,q) \in \mathcal{E}} \exp\left(- \frac{k \times d_{pq}^2}{(s_p^2 + s_q^2)} \right). \tag{15}$$

The cost function is also minimized by graph-cuts [20].

3.5 Text Line Refinement

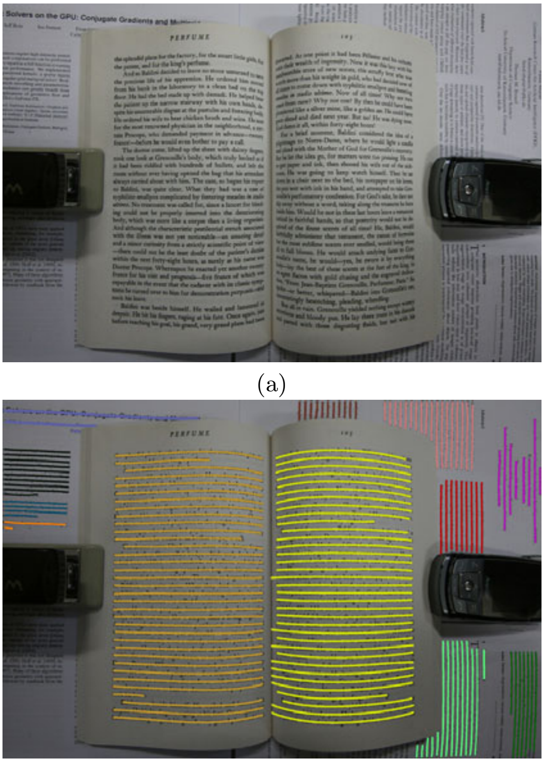
After inference, we remove non-textual clusters. Then, we re-detect text blocks because more than one text block might be merged into a single one via non-textual objects. Also, we re-detect text lines using the procedures presented in Section 3.2. However, at this time, we use a different sequence of $\{w_k\}$ (which will be presented in the experimental section) in order to prevent the over-segmentation of text lines. The final result can be found in Fig 1-(f). As shown in the figure, non-textual objects observed in Fig 1-(e) are successfully rejected.

4 Experimental Results

We have tested our algorithm with more than 300 images including camera-captured ones, and scanned ones. Inputs, binarized results, and experimental results can be found in our website (`http://ispl.snu.ac.kr/~hikoo/layout/`). Experiments were performed with parameters: $\sigma = 4.5$, $\mu_1 = 0.9$, $\mu_2 = 0.1 \times 255$, $\lambda_1 = 0.4$, $\lambda_2 = 5$, $\lambda_3 = 4$, $k = 0.125$, and $\epsilon = 2.8$. However, we have found that different settings of ϵ_1 and ϵ_2 sometimes provide better results than a default setting ($\epsilon_1 = 1.2$, $\epsilon_2 = 0.9$), and we also present such cases.

4.1 Qualitative Evaluation and Limitations

Fig. 1, Fig. 4, and results in our website show that our algorithm can detect text lines in scale, orientation, and language invariant manner. However, careful observation of them also reveals the limitations of our algorithm. First of all, our method has difficulty in detecting a single-line text because it exploits distribution pattern of text lines. Another limitation is that it is sensitive to motion



(a)

(b)

Fig. 4. Input and output of our algorithm

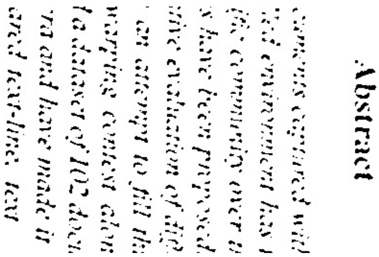
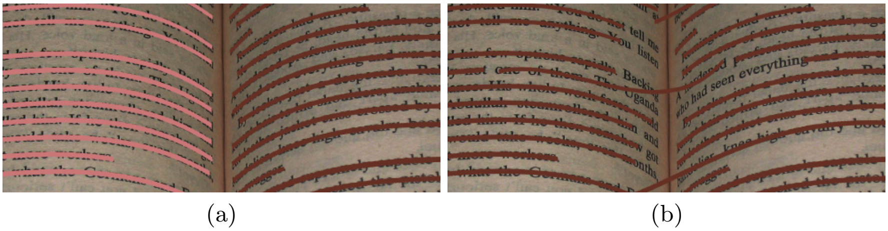


Fig. 5. The binarization result of the right upper part of Fig. 4-(a). As shown in Fig. 4-(b), text lines are successfully extracted even if the binarization performance is not good.



(a)

(b)

Fig. 6. *Expansion move* algorithm is sometimes stuck to poor local optima. (a) Our result for $E = 3469$, (b) Our result for $E = 3524$.

blur and shallow field of depth. Blurred inputs result in poor binarized images and they deteriorate the performance of other processes. Although our method tolerates the right upper part of Fig. 4-(a) (whose binarization result is shown in Fig. 5), it fails to handle more blurred inputs as can be found in the left lower part of Fig. 4-(a). The last problem comes from the *Expansion move* algorithm used in the minimization of (3). It sometimes stuck to local minima depending on its initialization. An example for this case is shown in Fig. 6.

4.2 Quantitative Evaluation on Camera Captured Images

For the quantitative evaluation of our method, we have selected 50 images (all of them can be found in our website) in our dataset. Fig. 1-(a) is the cropped version of one of them. For text line refinement, we use $w_1 = 0.8, w_2 = 1.0, w_3 = 1.2, w_4 = 2.0, w_5 = 4.0$, and $w_6 = 6.0$. Experimental results show that 95.7% text blocks among 185 text blocks are correctly detected (we only consider text blocks having more than one text line), and false positive and false negative are less than 2%. In text line detection, our algorithm detects 98.4% text lines correctly (we say that a line is correctly detected when the number of missed characters is less than 4). False positive and false negative are also less than 2%. If we ignore two occluded inputs (See the 25 and 36-th images in our dataset), the results will be improved more than 1%. Our algorithm takes less than 10 seconds in handling 3264 × 2488 inputs having $10,000$ CCs. Since there is much room for optimization and parallelization (e.g., construction of a data table), we believe that the computational complexity is reasonable.

4.3 Evaluation on Other Dataset

A direct comparison to existing method(s) is not a simple task. It is because our method has been developed to handle complex cases compared to conventional ones [3]. Moreover, our method includes text block identification, which has not been considered in conventional algorithms [3,16]. Therefore, we have applied our method to conventional cases (ICDAR dataset [3]) rather than applying conventional methods to our dataset. In this experiment, we use $(\epsilon_1, \epsilon_2) = (10, 10)$ rather than a default setting. It is because (1) text block segmentation is not an issue in this database (at least in terms of performance evaluation) and (2) our page segmentation method with a default setting is not suitable to detect subtitles or captions as shown in Fig 7-(b). In the text line refinement step, we use $w_1 = 0.8, w_2 = 1.0, w_3 = 1.2$, and $w_4 = 2.0$.

According to the evaluation method in [21,22], the match score is defined as

$$MatchScore(i, j) = \frac{|G_j \cap R_i|}{|G_j \cup R_i|} \tag{16}$$

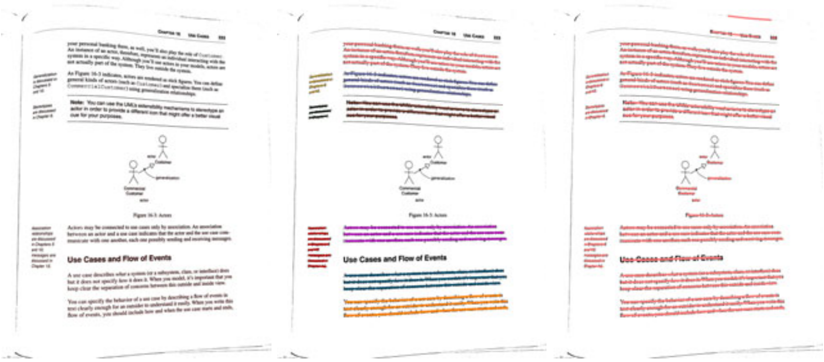
where G_j is the set of all pixels in the j-th ground truth text line and R_i is the set of all pixel in the i-th detected text line. Also, the correct segmentation accuracy [22] is defined as

$$100 \times \frac{\text{the number of matched } (G_j, R_i) \text{ pairs}}{\text{the number of ground truth text lines}} \tag{17}$$

where we consider (G_j, R_i) is a matched pair when $MatchScore(i, j) \geq 0.95$. Experimental results on 102 images show that the correct segmentation accuracy of our method is 92.76%, which is more than 1.7% higher accuracy than existing methods [22]. Some experimental results can be found in Fig 7. Due to a relatively small number of CCs, our algorithm takes less than 5 seconds on average.

4.4 Application to Skew Estimation

Our method can be applied to a skew estimation problem by modeling text lines as straight lines and computing the average angle of the detected text lines. Although the accuracy of this method is not high compared to conventional methods such as [23], our method is able to handle challenging cases. To be specific, experimental results on 30 vertically flowing text in [23] show that our method achieves an average error of 0.19° with a maximum error of 0.5°, while the method in [23] fails for 3 inputs (See. Table. 4 in [23]).



(a) (b) (c)

Fig. 7. Input and output of our algorithm on the dataset in [3]. (a) Input, (b) Result using $(\epsilon_1, \epsilon_2) = (1.2, 0.9)$, (c) Result using $(\epsilon_1, \epsilon_2) = (10, 10)$. Although the latter setting does not provide text block information, it can provide better text line extraction performance. Therefore, we use the latter setting for the evaluation.

5 Conclusion

In this paper, we have presented a novel approach to document image processing: text block identification and text line extraction. In order to handle complex cases, we assume that the states (line spacing, orientation, and other parameters describing the local properties of text region) of CCs are spatially varying, and the states are estimated using an energy minimization framework. Using the estimated states, we have also presented a new algorithm that performs text block identification and text line extraction. Experimental results on the extensive dataset show that our method is efficient, robust, and provides promising results.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of (NRF) funded by the Ministry of Education, Science and Technology (2009-0083495)

References

1. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. 15, 1162–1173 (1993)
2. Liang, J., DeMenthon, D., Doermann, D.: Flattening curved documents in images. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2005)
3. Shafait, F., Breuel, T.M.: Document image dewarping contest. In: Int. Workshop on Camera-Based Document Analysis and Recognition, pp. 181–188 (2007)

4. Stamatopoulos, N., Gatos, B., Pratikakis, I., Perantonis, S.: A two-step dewarping of camera document images. In: International Workshop on Document Analysis Systems, pp. 209–216 (2008)
5. Cao, H., Ding, X., Liu, C.: A cylindrical surface model to rectify the bound document. In: International Conference on Computer Vision, ICCV (2003)
6. Koo, H.I., Kim, J., Cho, N.I.: Composition of a dewarped and enhanced document image from two view images. IEEE Trans. Image Process. 18, 1551–1562 (2009)
7. Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 30, 941–954 (2008)
8. Zheng, Y., Li, H., Doermann, D.: Machine printed text and handwriting identification in noisy document images. IEEE Trans. Pattern Anal. Mach. Intell. 26, 337–353 (2004)
9. Xiao, Y., Yan, H.: Text region extraction in a document image based on the delaunay tessellation. Pattern Recognition 36, 799–809 (2003)
10. Kise, K., Iwata, M.: Segmentation of page images using the area voronoi diagram. Computer Vision and Image Understanding 70, 370–382 (1998)
11. Bukhari, S.S., Shafait, F., Breuel, T.M.: Coupled snakelet model for curled textline segmentation of camera-captured document images. In: International Conference on Document Analysis and Recognition, pp. 61–65 (2009)
12. Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30, 79–116 (1998)
13. Yin, F., Liu, C.L.: Handwritten chinese text line segmentation by clustering with distance metric learning. Pattern Recogn. 42, 3146–3157 (2009)
14. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry. Springer, Heidelberg (2000)
15. Antonacopoulos, A.: Page segmentation using the description of the background. Computer Vision and Image Understanding 70, 350–369 (1998)
16. Bukhari, S., Shafait, F., Breuel, T.: Segmentation of curled textlines using active contours. In: The Eighth IAPR International Workshop on Document Analysis Systems, DAS 2008, pp. 270–277 (2008)
17. Li, Y., Zheng, Y., Doermann, D., Jaeger, S.: Script-independent text line segmentation in freestyle handwritten documents. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1313–1329 (2008)
18. Pilu, M., Pollard, S.: A light-weight text image processing method for handheld embedded cameras. In: BMVC (2002)
19. Pogalin, E., Smeulders, A., Thean, A.: Visual quasi-periodicity. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
20. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23, 1222–1239 (2001)
21. Gatos, B., Antonacopoulos, A., Stamatopoulos, N.: Handwriting segmentation contest. In: International Conference on Document Analysis and Recognition, vol. 2, pp. 1284–1288 (2007)
22. Bukhari, S.S., Breuel, T.M., Shafait, F.: Textline information extraction from grayscale camera-captured document images. In: IEEE International Conference on Image Processing (ICIP), pp. 2013–2016 (2009)
23. Dey, P., Noushath, S.: e-pcp: A robust skew detection method for scanned document images. In: Pattern Recognition (2009)