Text Line Extraction of Curved Document Images Using Hybrid Metric

Zuming Huang, Jie Gu, Gaofeng Meng and Chunhong Pan
Institute of Automation, Chinese Academy of Sciences (CASIA)
No.95, Zhongguancun East Road, Beijing 100190, P.R. China
{huangzuming2014, gujie2014}@ia.ac.cn, {gfmeng, chpan}@nlpr.ia.ac.cn

Abstract

This paper proposes a novel approach to extracting text lines from curved document images that are captured from an opened thick and bounded book or a curled document sheet. We first extract the connected components (CCs) in a binary image and then remove the non-textual CCs. Additionally, we estimate the orientation of each CC through local projections and a feature vector is accordingly defined to describe each CC. Furthermore, a hybrid metric is designed based on the distances between CCs and the corresponding minimum spanning tree which can well exploit the overall structure of the curved text lines is constructed. A tree pruning strategy is finally proposed to cluster the CC-s into separated text lines. Experimental results on a wide variety of curved document images demonstrate the effectiveness and efficiency of the proposed method.

1. Introduction

Text line extraction is one of the active and importan-t area of research in document image analysis. It plays an essential role in text block segmentation, character segmentation, optical character recognition (OCR) and the rectification of curved document images [4, 3, 6, 10]. However, it is generally a quite challenging problem. To accurately extract the text lines in curved document images, many complicated factors have to be considered, such as distorted image contents, degraded image quality, multiple skews, non-textual regions, non-uniform font sizes, character adhesion, etc. Therefore, many successful methods that work well for text lines segmentation and layout analysis, including projections, document spectrum and minimum spanning tree (MST) based methods [3, 4, 11], often fail in this case.

In this paper, we propose a novel method to extract text lines from curved document images captured from opened books and curled paper sheets. Our method is a hybrid method combining local projections and MST. It first extracts connected components (CCs) from a binary document image. Then the non-textual CCs are removed by a simple yet efficient shape filter. In addition, the local projections of image is implemented to incorporate the orientation information of text lines into the feature vector of each CC. By doing so, a hybrid metric is constructed by the Euclidian distance between CCs, and then the corresponding MST is built to exploit the overall structure of CCs. Final-ly, a tree pruning strategy is proposed to divide the tree into separated text lines.

2. Related Work

Generally, there are two types of text line extraction methods. One is designed for handwritten documents so as to facilitate the recognition of each handwritten character. Noted that in these document images, the distortions are mainly from the variations of handwriting. The other is for printed documents, and the distortions are usually caused by page curl. Some representative extraction methods are reviewed below.

Yin and Liu [11] propose a text line segmentation algorithm based on MST clustering with distance metric learning. The distance metric is first constructed by supervised learning on a dataset of pairs of CCs. Then the CCs of document image are grouped into a tree structure, from which text lines are extracted by dynamically cutting edges using a new hypervolume reduction criterion and a straightness measure. This algorithm is robust to deal with various handwritten documents with multi-skewed and slightly curved text lines. However, the algorithm fails to handle document images with large distortions.

Stamatopoulos *et al.* [9] propose a goal-oriented rectification methodology and apply it directly to a single image to correct its geometric distortion. It first preprocesses the curved document images via adaptive binarization and black border removal [7]. Then all the words in the image are detected by connected components analysis. Text lines are finally extracted by linking the neighboring words. The method is quite simple and easy to implement. However, it often fails when a document image is severely distorted.

Tian and Narasimhan [10] propose a line tracing based text line extraction approach that can be implemented di-

978-1-4799-6100-9/15/$31.00 ©2015 IEEE

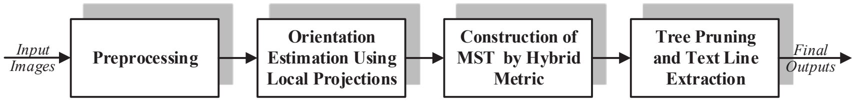

Input
Images
Preprocessing
Orientation Estimation Using Local Projections
Construction of MST by Hybrid Metric
Tree Pruning and Text Line Extraction
Final
Outputs

Figure 1. Flowchart of the proposed text line extraction method.

rectly on a gray-scale image. The method automatically identifies and densely traces the text lines in the image. The great advantages of this method are that it doesn't rely on some noise sensitive operations such as image thresholding and character segmentation, and no priori knowledge about font size, types or alphabet is needed. However, the method is often difficult to determine an appropriate step size for line tracing, and it does not work in non-textual regions.

3. The Proposed Method

In this section, we will present our proposed method in details. The method consists of four main steps, i.e., preprocessing, orientation estimation using local projections, construction of MST by hybrid metric, tree pruning and text line extraction. Fig. 1 illustrates the flowchart of the proposed text line extraction method. More details of these steps will be given in the following subsections.

3.1. Preprocessing

Our method starts with a binary image. Before extracting the text lines from the curved document image, we need to remove the non-textual noises to facilitate the subsequent connected components extraction and analysis.

Firstly, we extract all the CCs from the binary image by connected component analysis. These CCs often contain some non-textual objects due to inserted photos, tables and marginal noises. They can be removed by a simple shape filter based on the character size.

This shape filter can be obtained by estimating the dominant character height (DH). We calculate the height of the bounding box of each CC in the image and construct a height histogram. Since a document image generally consists of large number of characters with similar font sizes, the histogram often has a high peak. We just pick the height with highest peak as the dominant height of CCs.

We remove a CC if it is too large or too small. More specifically, we remove a CC if it satisfies the following conditions:

$$h > 4 \cdot DH \ \ or \ \ w > 4 \cdot DH \ \ or \ \ h \cdot w < DH^2/9, \qquad (1)$$

where h and w represent the height and width of CCs respectively. The constants above pare determined empirically by refining them on a number of images. These rules, although quite simple, are very effective to remove the non-textual objects in the image quickly.

It should be noted that the textual noises coming from neighboring pages are removed based on [7]. Besides, we record all the removed small CCs. In the last step of our method, we will check the association of these small CCs with the extracted text lines.

3.2. Orientation Estimation using Local Projections

After preprocessing, we estimate the text line orientation of each CC. Text line orientation is a very important feature of CC and it contributes to our hybrid metric.

Due to the distortions of the input image, different CCs tend to have obviously different orientations. To solve this problem, we use a local projection strategy. We first divide the text region of the document image into several square patches. Then we estimate the orientations of each patch by Radon transform. Finally, the orientation of each CC can be estimated based on the patches' orientations.

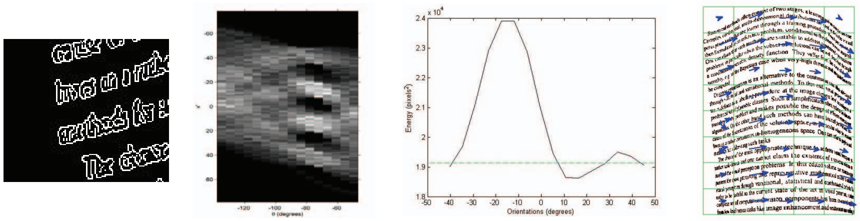
3.2.1 Estimation of Patch Orientation

To begin with, we estimate the bounding box (BBox) of the text region by viewing all the characters in the image as one CC. Then we divide the image in the BBox into a 2D patch array, i.e., the longer side of the BBox is first divided evenly into n sections, and the shorter side is then divided based on n and the aspect ratio of the BBox.

For each image patch, we first use Canny operator to compute its edge map and then apply Radon transform to project it along 16 specific orientations in $[-45°, 45°]$. The energy E_i $(1 \leq i \leq 16)$ of the Radon transform can be calculated by $E_i = \|R_i\|_2^2$, where R_i denotes the Radon transform for the i-th specific orientation. Moreover, an orientation histogram is formed and each sample in the histogram is weighted by its corresponding energy.

One obvious way to estimate patch orientation would be to find the orientation with the highest peak. However, the orientation histogram typically has multiple peaks and each local peak corresponds to a dominant orientation of the patch. Inspired by Lowe's work [2], we select the orientations with local peaks higher than 80% of the highest peak as the dominant orientations of the patch. By doing so, some patches may have more than one dominant orientations, these in fact significantly improve the robustness of the orientation estimation. Finally, we fit a parabola to the three histogram values closest to each peak to interpolate the peak position for better accuracy.

After that, the skew angle of each vertical strip which reflects the global text skew of the patches in the same column of the patch array is estimated by voting method. That



(a) (b) (c) (d)

Figure 2. Orientation estimation by local projections. (a) Edge map of an image patch. (b) The Radon transform. (c) The orientation histogram. The green dash line denotes the 80% of the highest energy peak. (d) The estimated patch orientations. The length of each arrow corresponds to the energy of each dominant orientation.

is, each dominant orientation of each patch in the vertical strip provides one vote weighted by its corresponding energy. Hence, the skew angle can be calculated by

$$\Omega_j = \frac{\sum_k \theta_{j,k} \cdot E_{\theta_{j,k}}}{\sum_k E_{\theta_{j,k}}}, \tag{2}$$

where Ω_j is the skew angle of the j-th vertical strip, $\theta_{j,k}$ denotes the k-th dominant orientation in the j-th vertical strip, and $E_{\theta_{j,k}}$ represents the weight of $\theta_{j,k}$.

3.2.2 Estimation of CC Orientation

The orientation of each CC can be estimated by its corresponding strip and image patch. Since the orientations of the image patch may have multiple values, we assign the value which is closest to the skew angle of the strip to the CC orientation. That is,

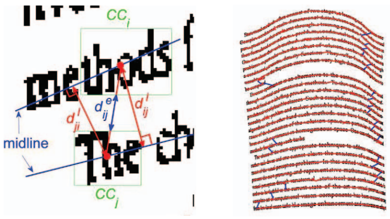
$$\hat{\theta}_i = \arg \min_{\theta_k \in \mathbf{p}} |\theta_k - \Omega_j| \tag{3}$$

where $\hat{\theta}_i$ is the estimated orientation of the i-th CC (CC_i), θ_k denotes the k-th orientation of the patch $\mathbf{p}$. Fig. 2 demonstrates the process of orientation estimation.

3.3. Construction of MST by Hybrid Metric

This subsection consists of three steps. We first design a hybrid metric between CCs. Then we use CCs to form a weighted graph based on the hybrid metric. Furthermore, the MST of the graph is constructed.

A well-designed metric should make the distance between two neighboring CCs small in the same text line, while make that large in different text lines. To meet this condition, we first construct a feature vector for each CC, and then design a hybrid metric based on the feature vectors.



(a) (b)

Figure 3. Construction of MST with hybrid metric. (a) The edge to edge distance and the centroid to midline distance between CC_i and CC_j. (b) The MST. The blue lines denote the between-line edges.

The feature vector $\mathbf{F}$ can be described as

$$\mathbf{F}_i = (x^c_i, y^c_i, w_i, h_i, \hat{\theta}_i), \tag{4}$$

where (x^c_i, y^c_i), (w_i, h_i), and $\hat{\theta}_i$ denotes the centroid's coordinate, the size, and the orientation of CC_i respectively. The hybrid metric which utilizes the feature vectors above is a combination of two factors, as illustrated in Fig. 3(a). The first is the edge to edge Euclidian distance along the line joining the centroids of two neighboring CCs. The second is the Euclidian distance between the centroid of a CC and its neighboring CC's midline (the line which crosses the centroid of CC and has the same orientation). In more detail, the distance between CC_i and CC_j, i.e., D_{ij}, is formulated as Eq. 5. The d^e_{ij} is the edge to edge distance of CC_i and CC_j, d^l_{ij} and d^l_{ji} is the centroid to midline distance between CC_i and CC_j, and the parameter λ denotes the trade-off between the two factors, and the parameters λ_1 and λ_2 can be adjusted to fit the various line spacings of the document image.

Then, a weighted graph is formed by applying Delaunay triangulation to the centroid of CCs. The weight of each edge in the graph is assigned based on Eq. 5. Finally, the

$$\hat{d}_{ij} = \begin{cases} d^e_{ij} + \lambda \cdot \sqrt{d^l_{ij} d^l_{ji}} & d^e_{ij} \leq \lambda_1 \cdot DH \text{ and } \sqrt{d^l_{ij} d^l_{ji}} \leq \lambda_2 \cdot DH \\ +\infty & \text{otherwise} \end{cases}, \tag{5}$$

MST of the graph is found by Prim's algorithm [5] to exploit the overall structure of the curved text lines. Fig. 3(b) shows an example of MST construction.

3.4. Tree Pruning and Text Line Extraction

Since the constructed MST contains the text lines, we can extract text lines by pruning the between-line edges. According to Eq. 5, the weights of between-line edges typically have infinite values. Therefore, the tree pruning step is very easy to operate. we can simply cut the edges with weight $+\infty$ to cluster the CCs into text lines.

It should be noted that some small CCs removed in preprocessing may also be a part of the extracted text lines. Hence, we check the association of the small CCs with the text lines by minimum distance criteria. For each CC, we find its closest text line. The small CC to text line distance is defined by the minimum edge to edge distance between small CC and CCs in the text line. If the distance between the small CC and its closest text line is less than $2 \cdot DH$, we regard the small CC as a part of its closest text line.

4. Experimental Results

We have implemented our text line extraction method in Matlab R2014a and have tested it on a wide variety of curved document images with different image sizes, layouts and distortions (our dataset and CBDAR 2007 dewarping contest dataset). The experiments are performed with parameters $n = 10$, $\lambda = 5$, $\lambda_1 = 2.5$, $\lambda_2 = 0.9$. Fig. 4 shows some representative results of our experiments. The first four images are selected from our dataset and the last four images are selected from CBDAR 2007 dewarping contest dataset. It should be noted that in the last four images, the text regions coming from neighboring pages are viewed as textual noises and are removed in preprocessing.

We also compared our algorithm with two state-of-the-art methods, i.e., Stamatopoulos *et al.*'s method[1] [9], and Tian and Narasimhan's method[2] [10]. Fig. 5 and Tab. 1 illustrate the comparisons of the three methods.

The first image in the first row of Fig. 5 is capatured from an opened thick book with slight distortions and textual noises. The second image in the second row is capatured from a curved document sheet with severe distortions and degraded text quality. As can be seen from the results, all methods produce a desirable result with slightly distorted text lines. However, Tian and Narasimhan's method cannot deal with the non-textual object, while our method and Stamatopoulos *et al.*'s method can address this problem. Besides, when the document image has degraded text quality or textual noises coming from neighboring page, Stamatopoulos *et al.*'s method fails. In comparison, the proposed method and Tian and Narasimhan's method can handle this case quite well. Furthermore, both of the two compared methods do not work with severe distortions, while our method yields a high quality result, which is comparable to the ground-truth.

Table 1. Performance evaluation of three text line extraction methods on CBDAR 2007 dewarping contest dataset.

Method	Recall Rate	Precision Rate
The proposed	92.67%	69.91%
Stamatopoulos *et al.* [9]	89.02%	42.51%
Tian and Narasimhan [10]	66.28%	56.50%

Additionally, we evaluate the performances of the three methods on CBDAR 2007 dewarping contest dataset, which consists of 102 curved document images with approximately 3119 text lines. Based on the evaluation method in [8], the match score between an extracted text line and a ground-truth text line is defined as

$$MatchScore(i, j) = \frac{|G_j \cap R_i|}{|G_j \cup R_i|}, \tag{6}$$

where G_j is the set of all points in the j-th ground-truth text line and R_i is the set of all points in the i-th extracted text line. The one-to-one correspondence between the extracted text lines and the ground-truth ones is found by Munkres algorithm [1]. We consider an extracted text line as correctly extracted if its match score is at least 0.95. The average recall rate (the percentage of correctly extracted text lines out of the ground-truth text lines) and the average precision rate (the percentage of correctly extracted text lines out of the extracted text lines) of the three methods are calculated.

As we can see in Tab. 1, our method, compared with other two methods, produces promising results both in recall rate and precision rate. These results show the effectiveness and efficiency of the proposed method. Nevertheless, our method also have some limitations. Firstly, our method has difficulty in dealing with complex objects such as mathematical equations and engineering diagrams. Besides, our method is sensitive to line spacings since we assume that the within-line spacings are not too large and the between-line spacings are not too small.

5. Conclusion and Future Work

This paper proposes a novel approach to extract the text lines from curved document images. We first extract the CCs of a binary image and remove the non-textual regions by a shape filter. Then we estimate the orientation of each CC using local projections. Moreover, we construct the MST of CCs based on a hybrid metric. Finally, we extract the text lines with a tree pruning strategy. Experi-

[1] This method was re-implemented by ourselves.

[2] The implementation is available at http://www.cs.cmu.edu/~ILIM/projects/IM/document_rectification/document_rectification.html.

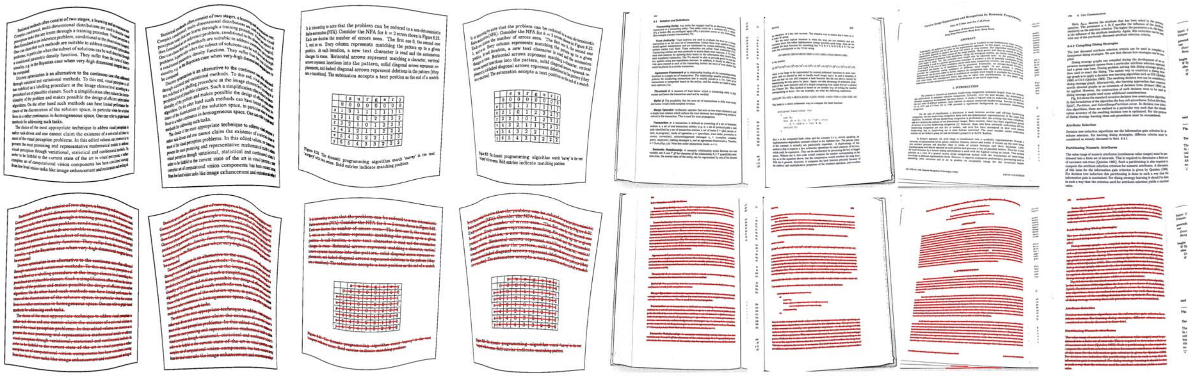
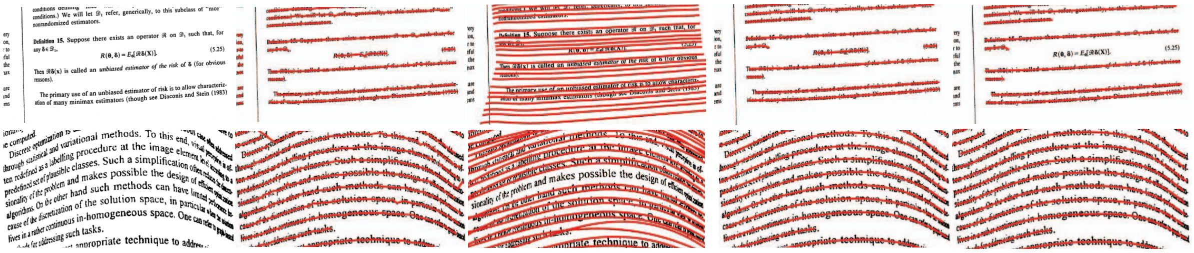


Figure 4. Some examples results by our method on different types of curved document images. (the first row) Curved document images. (the second row) The text extraction results.



Original Stamatopoulos *et al.* Tian and Narasimhan Our Method Ground-truth

Figure 5. Comparisons of our method with Stamatopoulos *et al.*'s method and Tian and Narasimhan's method on text line extraction of curved document images. From left to right: original inputs, Stamatopoulos *et al.*'s results, Tian and Narasimhan's results, our results and the ground-truths.

ments on a wide variety of curved document images show the effectiveness and efficiency of our proposed method.

In the future, we would like to improve our algorithm to deal with document images with complex layouts, such as complicated non-textual objects, various line spacings, etc. We also wish to extend our method to handle more general documents with various character types (e.g., Chinese documents, Hindi documents, etc.).

Acknowledgement

The authors would like to thank anonymous reviewers for valuable comments. This work was supported by the National Natural Science Foundation of China (Grant No. 61370039).

References

[1] F. Bourgeois and J. C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the Acm*, 14:802–804, 1971. 4

[2] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, pages 91–110, 2004. 2

[3] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, 1992. 1

[4] O'Gorman and L. The document spectrum for page layout analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1162–1173, Nov 1993. 1

[5] R. C. Prim. Shortest connection networksand some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957. 4

[6] F. Shafait. Document image dewarping contest. In *in 2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, pages 181–188, 2007. 1

[7] N. Stamatopoulos, B. Gatos, and A. Kesidis. Automatic borders detection of camera document images. In *2nd International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil*, pages 71–78, 2007. 1, 2

[8] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei. Icdar 2013 handwriting segmentation contest. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1402–1406. IEEE, 2013. 4

[9] N. Stamatopoulos, B. Gatos, I. Pratikakis, and S. J.Perantonis. Goal-oriented rectification of camera-based document images. *IEEE Transactions on Image Processing*, 20(4):910–920, 2011. 1, 4

[10] Y. Tian and S. G. Narasimhan. Rectification and 3d reconstruction of curved document images. In *Computer Vision and Pattern Recognition*, Jun 2011. 1, 4

[11] F. Yin and C.-L. Liu. Handwritten chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition*, 42(12):3146–3157, 2009. 1