Text Line Extraction based on Integrated K-shortest Paths Optimization

Liuan Wang
Fujitsu R&D Center CO., LTD
Beijing, China
liuan.wang@cn.fujitsu.com

Jun Sun
Fujitsu R&D Center CO., LTD
Beijing, China
sunjun@cn.fujitsu.com

Seiichi Uchida
Kyushu University
Fukuoka, Japan
uchida@ait.kyushu-u.ac.jp

***Abstract*—Text in images can be utilized in many image understanding applications due to the exact semantic information. In this paper, we propose a novel integrated k-shortest paths optimization based text line extraction method. Firstly, the candidate text components are extracted by the Maximal Stable Extremal Region (MSER) algorithm on gray, red, green and blue channels. Secondly, one integrated directed graph on red, green, and blue channels are constructed upon the candidate text components, which can effectively incorporate different channels into one framework. Then, the integrated directed graph is transformed guided by the extracted text lines in gray channel to reduced the computational complexity. Finally, we use the k-shortest paths optimization algorithm to extract the text lines by taking advantage of the particular structure of the integrated directed graph. Experimental results demonstrate the effectiveness of the proposed method in comparison with state-of-the-art methods.**

***Keywords*-text line extraction; directed graph transformation; integrated k-shortest paths optimization**

I. Introduction

Text in images contains useful semantic information and the extracted text can be used in automatic annotation, indexing, and retrieval [1]. Text line extraction is a crucial prerequisite for content-based image understanding systems, and it plays an important role in Optical Character Recognition applications. However, variations of text make the problem of automatic text extraction a challenging task due to differences in terms of size, orientation, low image contrast and complex background. Some of challenging image text lines from International Conference on Document Analysis and Recognition (ICDAR) competition database are shown in Figure.1.

To address the text line extraction problem, a large number of techniques have been proposed, which can be roughly categorized into two types [2]: region-based and Connected Component (CC)-based. Region-based methods try to extract discriminative hand-crafted features and distinguish the text regions from the background. [3] implements a scale-space feature extractor that feeds an artificial neural processor to detect text blocks for video text detecting and tracking. The text tracking scheme consists of two modules: a sum of squared difference module to find the initial position and a contour module to refine the position. [4] uses a multiple-continuously adaptive mean shift algorithm on the text probability image and the scene text lines on planar rectangular surfaces are assigned with homogeneous background colors. [5] describes a robust and accurate multi-resolution approach to detect and classify text regions. Images are first segmented to detect character regions with a multi-resolution algorithm able to manage large character size variations, the segmented regions are then filtered out using shape-based classification, and neighboring characters are merged to generate text hypotheses.

As opposed to region-based methods, the CC-based methods attempt to group small text components into text lines successively. [6] extracts MSERs as character candidates using the strategy of minimizing regularized variations, and then the candidates are grouped into text lines by the single-link clustering algorithm, where distance weights and threshold of the clustering algorithm are learned automatically by a self-training distance metric learning algorithm. [7] uses a graph model built upon MSERs to incorporate various text information sources into one framework, it formulates the text detection as a bi-label (text and non-text regions) segmentation problem, and the cost function could be optimally minimized via graph cut algorithm to get the final labeling results. [8] proposes a unified scene text detection system named text flow by utilizing the minimum cost flow network model. The min-cost flow network model can solve the error accumulation problem at both character level and text line level effectively by integrating the last three sequential steps into a single process.

Figure 1. Image text line examples on ICDAR dataset.

In this paper, we propose a novel integrated k-shortest paths optimization based text line extraction method. The integrated directed graph can effectively incorporate different channels into one framework and eliminate the disorder of text components. To improve the efficiency and decrease the directed graph complexity, we transform the integrated directed graph into reduced graph guiding by the extracted gray channel text lines. Finally, the k-shortest paths optimization algorithm is utilized to extract the text

978-1-5386-3346-5/18 $31.00 © 2018 IEEE
DOI 10.1109/DAS.2018.68

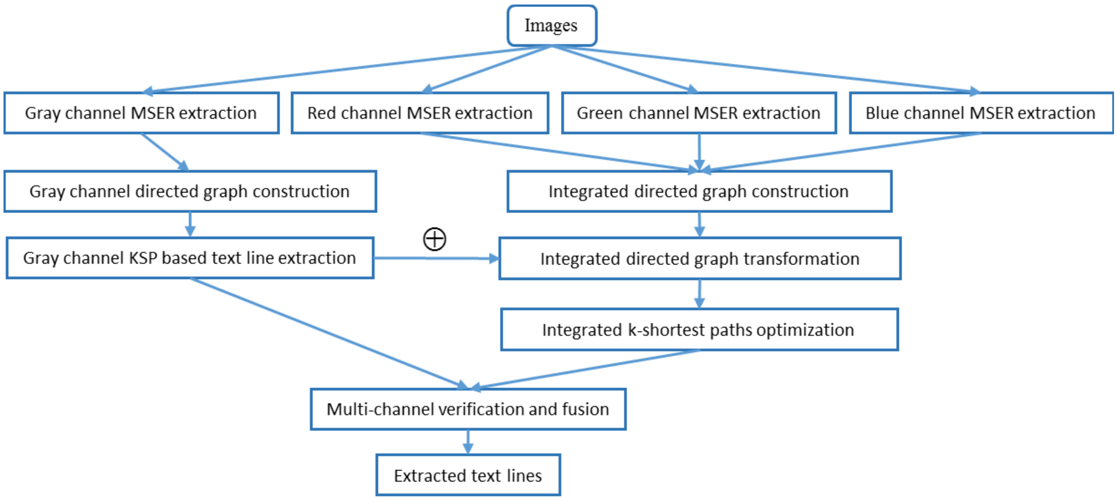


Figure 2. The flowchart of our proposed integrated k-shortest paths optimization based text line extraction.

lines by taking advantage of the particular structure of the integrated directed graph. The flowchart of the proposed method is shown in Figure. 2.

The main contributions of the proposed text line extraction method are described as follows.

- The integrated directed graph can effectively describe the relationship structure and eliminate the disorder of text components among different channels.
- The integrated directed graph can be transformed guiding by the extracted gray channel text lines, which can improve the efficiency and decrease the directed graph complexity.
- We demonstrate that the integrated k-shortest paths optimization based text line extraction method can effectively improve the performance of text line extraction by incorporating candidate text components of different channels into one framework.

II. Text Component Extraction

Candidate text components are extracted by MSER method [9] in gray, red, green and blue channels. For each channel, Two polarities of text components, named black text on white background and white text on black background, are extracted based on the MSER method. The Maximal Stable Extremal Region [10] has been one of the most widely used method for text component extraction. It achieved promising performance and won the first place in ICDAR2011 [11] and ICDAR2013 [12] competitions. The channel combination and candidate text line certification on different channels can effectively improve the text line performance.

Although the MSER method can extract most of the text components even in challenging low quality images. It suffers from non-text false alarms in terms of low precision, which increase the difficulty to identity the true texts and text line extraction. We discriminate the non-text components from true positives by employing a Convolutional Neural Network (CNN) [13]. It is capable of meaningful high-level features and semantic representation for image text component. Given a binary text MSER, it is firstly normalized to fixed 40×40 pixels by keeping aspect ratio and fed into a well-trained CNN to predict the text or non-text label. The designed CNN consists of three convolutional layers, which produce 32, 64, 128 feature maps with 3×3 kernels in one stride, respectively. Each convolutional layer follows a max-pooling layer with 3×3 kernels and a stride of 2. Finally, A fully connection layer with 1024 neurons is concatenated in the last layer and a soft-max classifier is applied in the output layer to estimate the text and non-text probability distribution. We harvest the training samples on the ICDAR2011 [11] and SVT [14] training dataset by using the MSER detector.

III. Text Directed Graph Construction

To incorporate different channels into one framework, we construct one integrated directed graph on red, green, blue channels. The directed graph connects all the neighboring text pairs and it can effectively construct the text relationship. The target text lines are assumed included as paths in the constructed directed graph.

A. Integrated Directed Graph Construction

The integrated directed graph $G_m = (V, E)$ is constructed upon the extracted text components in red, green, blue channels in the assigned text line direction from left to right according to the position relationship. Algorithm 1 gives the directed graph construction details. The vertices of directed graph $V = \{v_c^i\}_{i=1}^{n_c}$ are comprised of text components location, where n_c denotes the number of text component vertex at the given channel. The directed edges $E = \{e_{i,j} \mid v_i, v_j \in V\}$ are built between any two linked vertices from v_i to v_j for each channel in the integrated directed graph. Since text cannot exchange the channel, there are no directed edges between channels and the integrated directed graph is made of disconnected channels as shown in Figure. 3b.

For each vertex v_i, we firstly check any other vertex v_j whose horizontal coordinate x_j is larger ($x_j > x_i$) than the

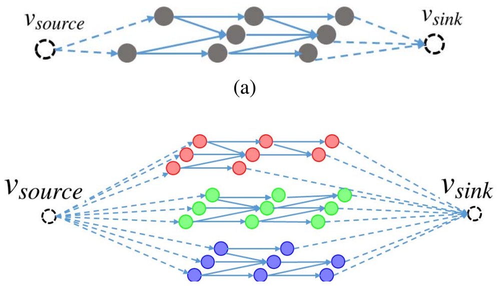


Figure 3. The proposed text line extraction involves one gray-channel directed graph and one integrated directed graph. (a) gives one example of gray-channel directed graph. (b) illustrates the integrated directed graph.

horizontal coordinate x_i of vertex v_i. Then, the horizontal overlapping constraint, shown in Figure. 4, is certificated to build the candidate vertex pair $p_{i,j}$ from vertex v_i to v_j. Finally, we find the nearest text component vertex v_j from the candidate pair set $p_{i,j}$ to build the directed edge $e_{i,j}$ from current vertex v_i to selected vertex v_j. To allow each text component to be the start or end location of the text line, two additional virtual vertices v_{source} and v_{sink} are added to the integrated directed graph, where v_{source} and v_{sink} denote the possible source and sink point, respectively. Meanwhile, we build the directed edges $e_{source,i}$ from virtual vertex v_{source} to each vertex v_i and $e_{i,sink}$ from each vertex v_i to sink vertex v_{sink}. The virtual source and sink are connected to each vertex in every channel, which is illustrated in Figure. 3 to avoid overloading.

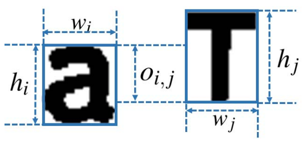


Figure 4. Illustration of overlapping constraint, the overlapping ratio r_o can be calculated by equation $r_o = \frac{o_{i,j}}{h_i+h_j}$.

The gray-channel directed graph can also be constructed by the Algorithm 1 by only utilizing the gray channel candidate text components. The integrated directed graph can be depicted as a duplication of the graph with other channels.

B. Cost Function of Directed Graph Edge

The cost function $c(e_{i,j})$ measures the cost value from MSER node v_i to MSER node v_j in the directed graph edge. We employ the unary cost function and pairwise cost function to calculate the cost of each directed edge. The unary cost function, including probability of recognition and variation of MSER, tries to measure the cost value for classifying the candidate MSER into text. As opposed to unary cost function, The pairwise cost function attempts to measure the cost value for the discontinuity of two linked text candidates in the text line, which is comprised of location distance and overlapping ratio of linked vertices.

Algorithm 1 integrated directed graph construction.

Require:
The set of extracted MSER vertices in red, green, blue channels, $V = \{v_c^i\}_{i=1}^{n_c}$.
Ensure:
Constructed integrated directed graph, $G_m = (V, E)$.

```
for channel = {red, green, blue} do
  for i = 1 to n_c do
    Build directed edge e_{source,i} from virtual source vertex v_{source} to vertex v_i;
    Build directed edge e_{i,sink} from vertex v_i to virtual sink vertex v_{sink};
    for j = 1 to n, j ≠ i do
      if x_j > x_i then
        Verify the pair by the overlapping constraints in the corresponding channel;
        Build component pair p_{i,j} between vertex v_i to v_j, and add it to the candidate set {p};
      end if
    end for
    Find the nearest vertex v_j with minimum distance in the set of {p};
    Build directed edge e_{i,j} from vertex v_i to v_j;
  end for
end for
return G_m = (V, E);
```

We assign an integrated cost value $c(e_{i,j})$ comprised of each item of unary cost and pairwise cost to the directed edge $e_{i,j}$ from node v_i to node v_j.

$$c(e_{i,j}) = -\log\left(\frac{\sum \lambda \cdot c^*}{1 - \sum \lambda \cdot c^*}\right), \tag{1}$$

where the coefficient λ is the weight trade-off between each cost value, which is set as 1.0 for equal weight. c^* is normalized to (0-1) for each item of unary cost and pairwise cost. Note that the cost values of directed edge $e_{source,i}$ and $e_{i,sink}$ are set as 0 to allow each text component to be the start or end location of the text line at no cost.

IV. INTEGRATED K-SHORTEST PATHS OPTIMIZATION

The text line extraction can be reformulated as a minimum cost path flow optimization problem with a 0-1 constraint in the directed graph. One text line can be regarded as a path from source text component to the sink text in the direction from left to right. If 1, it suggests the vertices are part of a text line.

The text line path selecting and routing in the integrated directed graph can be solved by the k-shortest paths global optimization algorithm [15]. The cost function is minimized to find the optimal solution in the directed graph from the virtual source vertex v_{source} to sink vertex v_{sink} by the following equation.

$$f = \underset{G_m=(V,E)}{\operatorname{argmax}} \sum c(e_{i,j}) \cdot l(e_{i,j}), \tag{2}$$

where $c(e_{i,j})$ and $l(e_{i,j})$ denote the cost function and label for edge $e_{i,j}$ from vertex v_i to vertex v_j, respectively.

A. K-shortest Path Optimization

The k-shortest paths algorithm is widely applied for path programming and object trajectory tracking [16] in videos. It attempts to extract k paths $\{p_1, p_2, ..., p_k\}$ iteratively with minimum total cost value, where k is fixed. The text line presents any feasible path flow (red text path arrows as shown in Figure. 5) in the directed graph between v_{source} and v_{sink}.

Figure 5. Text line paths between source vertex v_{source} and sink vertex v_{sink}.

We apply the Bellman-Ford algorithm [17] to generate the single shortest path in initialization. At the n_{th} iteration, the n shortest text line paths can be produced by updating previous $n-1$ shortest text line paths $\{p_1, p_2, ..., p_{n-1}\}$. The total cost value of the $n-$shortest paths at the n_{th} iteration can be calculated as following:

$$c(P_n) = \sum_{i=1}^{n} \sum_{e_{i,j} \in p_l} c(e_{i,j}), \tag{3}$$

where l is the l^{th} path at n^{th} iteration and the single shortest text line path cost value of p_l can be calculated by all of the cost value of graph edge $e_{i,j}$ belonging to the l^{th} shortest path.

The total cost value $c(P_n)$ and $c(P_{n-1})$ is compared to analyze the variation at adjacent iteration. The total path cost value is monotonically increasing and the global optimal minimum is achieved when the cost value change sign and become decreasing. Meanwhile, we obtain the optimal parameter k of the text line number.

$$\begin{cases} c(P_{n-1}) \leq c(P_n) \\ c(P_n) \geq c(P_{n+1}) \end{cases} \tag{4}$$

B. Integrated Directed Graph Transformation

In the integrated directed graph, the computational complexity can be significantly reduced by grouping unambiguously text nodes. Firstly, the consistent text paths are generated by k-shortest paths optimization algorithm on single gray channel commodity. We then transform the integrated directed graph on red, green, and blue channel commodities by grouping the unambiguously nodes into tuplets. The transformed directed graph is treated as a reduced graph with with tuplets nodes and individual nodes. Finally, we apply the k-shortest path optimization algorithm on the reduced integrated graph to extract the red, green, blue channels text line paths.

In the consistent text paths generation step, we can obtain a set of text paths $\{p_1, p_2, ..., p_k\}$, which are set as the guided text lines. For each guided text line, we track it in every layer of the integrated directed graph and find the duplicate text line segments. Meanwhile, the guided text lines can be split into connected text line segments to match the duplicate text line in the integrated directed graph.

The text path segments in the integrated directed graph, whose replica can be tracked in guided text lines, are defined as unambiguously text line tuplets due to exact verification in different channels. For each unambiguously text line tuplet, we disconnect all the directed edges between input and output vertices and build one directed edge e_{tuplet} from the input vertex to output vertex. The cost value of directed edge $c(e_{tuplet})$ can be calculated by all the edges E_{tuplet} from the input vertex to output vertex in the original directed graph,

$$c(e_{tuplet}) = \sum_{e_{i,j} \in E_{tuplet}} c(e_{i,j}), \tag{5}$$

We now transform the original directed graph $G_m = (V, E)$ to new integrated directed graph $G_m{}^* = (V^*, E^*)$, the vertices of which are composed of tuplet vertices and individual vertices. Figure. 6 illustrates the procedure of integrated directed graph transformation. Assumed that two text line paths (Figure. 6a) $p_1 = \{(v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4)\}$ and $p_2 = \{(v_5 \rightarrow v_6 \rightarrow v_7\}$ are extracted in gray channel by k-shortest path optimization algorithm, and then we track these two text line paths in integrated directed graph, as shown in Figure. 6b. The reduced integrated directed graph can be obtained by transformation.

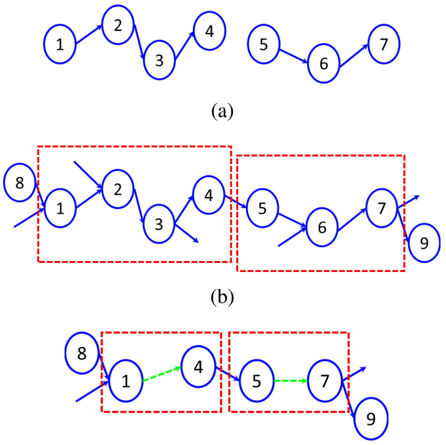
V. Text Line Fusion

We produce the text line paths in gray, red, green, and blue channels. The extracted text lines can be certificated by different channels. Meanwhile, the duplicate text lines should be removed to avoid repeat extraction. If all of the text components in the text line are included in text lines from other channels, the included text line should be removed as the duplicate noise.

VI. Experimental Results

To evaluate the effectiveness of the proposed integrated k-shortest paths optimization based text line extraction method, we demonstrate the experimental results on publicly available benchmark scene text detection database ICDAR2011 [11], which contains 484 images comprised of 229 images for training and 255 images for testing.

The traditional region-based evaluation metrics [18] named precision, recall, and f-measure are employed to



(a)

(b)

(c)

Figure 6. The integrated directed graph transformation. Supposed the extracted text lines are presented in (a) and we track them in the integrated directed graph as depicted in (b). (c) illustrates the transformation result.

quantify the performance. The precision measures the ratio between the extracted true positives and all extracted text lines, and the recall measures the ratio between the extracted true positives and all truth text lines, respectively. While, the f-measure is an overall evaluation by combining the precision rate and recall rate.

We firstly extract the guided text lines on gray channel by applying the k-shortest path optimization algorithm, it's set as the baseline for comparison with the proposed integrated path flow method. The baseline recall, precision, and f-measure performance are 0.716, 0.902, 0.798, respectively. As show in Figure. 7a, the text line "DEBENHAMS" and "www.debenhams.com" are extracted successfully by the k-shortest optimization method. However, we cannot extract the text line "BRITAIN'S FAVOURITE DEPARTMENT STORE" due to complex background on gray channel.

In the integrated directed graph, the guided text lines "DEBENHAMS" and "www.debenhams.com" are employed to deal with the transformation. The directed graph paths $p = \{(v_D \rightarrow v_E \rightarrow v_B \rightarrow v_E \rightarrow v_N \rightarrow v_H \rightarrow v_A \rightarrow v_M \rightarrow v_S)\}$ is transformed into $p^* = \{(v_D \rightarrow v_S)\}$ on all integrated channels. In each channel, the directed graph nodes are reduced from 9 nodes to 2 nodes and the edges are reduced from 8 edges to 1 edge, respectively. Meanwhile, The cost value of directed edge $c(e_{D,S})$ can be calculated by all the cost value of edges from the vertex v_D to vertex v_S. We can make similar transformation for the guided text line "www.debenhams.com" in the integrated directed graph. Obviously, the integrated directed graph transformation can effectively degrade the computation complexity. We also observed 1.5 times acceleration in the integrated k-shortest paths optimization by comparison with separated channels. Finally, we can extract the text lines in the integrated directed graph comprised of different channels, as shown in Figure. 7b.

We illustrate the performance of integrated k-shortest

(a) (b)

Figure 7. Text line extraction results. (a) shows the gray channel k-shortest path optimization result and (b) shows the text line extraction results of integrated k-shortest paths optimization method.

paths optimization method in Table.I, the proposed method achieves the recall, precision and f-measure performance of 0.788, 0.866, 0.825, respectively. Comparison with the baseline, The proposed integrated k-shortest paths optimization method by incorporating different channels into one framework can significantly improve the text line extraction performance. We also achieve competitive text line extraction performance in comparison with the state-of-the-art methods in terms of recall and f-measure.

Table I
EXPERIMENTAL RESULTS OF THE PROPOSED TEXT LINE EXTRACTION METHOD ON ICDAR2011 DATABASE.

method	recall	precision	f-measure
[19]	0.581	0.672	0.623
[20]	0.647	0.731	0.687
[7]	0.631	0.833	0.718
[21]	0.750	0.820	0.730
[6]	0.683	0.863	0.762
Baseline	0.716	0.902	0.798
[22]	0.760	0.860	0.800
[8]	0.762	0.862	0.809
The proposed method	0.788	0.866	0.825

Although the proposed integrated k-shortest paths optimization based text line extraction method can achieve encourage performance, there are still some limitations. Firstly, the precision performance will degrade a little because more false positive text characters are generated in red, green, and blue channels. Secondly, the integrated text components extraction is incapable of extracting single character text lines due to no path existing.

Figure.8 demonstrates some successful and failure examples of the proposed text line extraction method. The green, blue, red text line bounding boxes present the correct, false positive and failure of the proposed method. Obviously, The proposed integrated k-shortest paths optimization based text line extraction method is effective for text line various from polarity, multi-lines, low contrast and complex background text lines.

VII. CONCLUSIONS

In this paper, we propose a novel integrated k-shortest paths optimization based text line extraction method on gray, red, green, and blue channels. In gray channel, we build one directed graph upon the text components vertices and apply the k-shortest paths optimization method to produce the guided text lines. In red, green, and blue

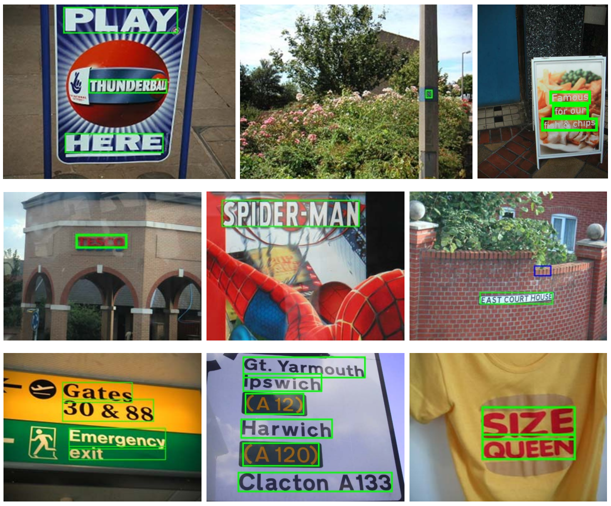


Figure 8. Examples of integrated path flow based text line extraction results.

channels, the integrated directed graph are constructed to integrate different channels into one framework to improve the text line extraction performance. Then, we transform the integrated directed graph by the guided text lines to reduce the computation complexity. Finally, we use the k-shortest paths optimization algorithm to extract the text lines by taking advantage of the particular structure of the integrated directed graph. The experimental results demonstrate that the text line extraction performance can be effectively improved by incorporating candidate text components of different channels into one integrated directed graph.

References

[1] C. Sumathi, T. Santhanam, and G. G. Devi, "A survey on various approaches of text extraction in images," *International Journal of Computer Science and Engineering Survey*, vol. 3, no. 4, p. 27, 2012.

[2] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: a survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977 – 997, 2004.

[3] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE transactions on image processing*, vol. 9, no. 1, pp. 147–156, 2000.

[4] K. Jung, K. I. Kim, and J. Han, "Text extraction in real scene images on planar planes," vol. 3, pp. 469–472, 2002.

[5] R. Minetto, N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui, "Snoopertext: A multiresolution system for text detection in complex visual scenes," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 3861–3864.

[6] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 970–983, 2014.

[7] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, "Scene text detection using graph model built upon maximally stable extremal regions," *Pattern recognition letters*, vol. 34, no. 2, pp. 107–116, 2013.

[8] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan, "Text flow: A unified text detection system in natural scene images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4651–4659.

[9] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *European Conference on Computer Vision*. Springer, 2008, pp. 183–196.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[11] A. Shahab, F. Shafait, and A. Dengel, "Icdar 2011 robust reading competition challenge 2: Reading text in scene images," in *2011 international conference on document analysis and recognition*. IEEE, 2011, pp. 1491–1496.

[12] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "Icdar 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1484–1493.

[13] Y. LeCun, K. Kavukcuoglu, C. Farabet *et al.*, "Convolutional networks and applications in vision." in *ISCAS*, 2010, pp. 253–256.

[14] K. Wang and S. Belongie, "Word spotting in the wild," in *European Conference on Computer Vision*. Springer, 2010, pp. 591–604.

[15] J. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, no. 2, pp. 125–145, 1974.

[16] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1614–1627, 2014.

[17] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, pp. 87–90, 1958.

[18] C. Wolf, J. michel Jolion, and B. J. Verne, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.

[19] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2963–2970.

[20] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3538–3545.

[21] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1241–1248.

[22] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2558–2567.