IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 12, 2017, accepted January 9, 2018, date of publication February 7, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2797181

Scene Video Text Tracking With Graph Matching

WEI-YI PEI[1], CHUN YANG[1], LI-YU MENG[1], JIE-BO HOU[1], SHU TIAN[1], AND XU-CHENG YIN[2], (Senior Member, IEEE)

[1]Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2]Department of Computer Science and Technology and the Beijing Key Laboratory of Materials Science Knowledge Engineering, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

Corresponding author: Xu-Cheng Yin (xuchengyin@ustb.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61473036 and in part by the Fundamental Research Funds for the Central Universities under Grant 2302016FRF-TP-16-048A1.

ABSTRACT Video has become one of the dominant data resources with the development of the Internet. As a result, the structured sorting of videos, which can be used for storage and extraction, represents a growing concern in the community. In particular, the text within videos can carry rich semantic information, leading to many novel studies wherein text tracking and recognition are performed. One essential step in text tracking involves template matching. In general, the adjacent matrices are modeled to represent the extracted tracking object features. Then, often, the Hungarian algorithm is applied to find the correspondence pairs between consecutive frames. In many works, text features are extracted based on morphological features such as color histograms and aspect ratios. However, under those features, similar text objects are not sufficiently distinguishable to make a distinction between them. To address this issue, we regard the template matching task as a graph matching problem. The main novelty involves a graph matching approach that utilizes the relationship between two trajectories or two objects, whereby a graph matching solver can be readily used in our tracking system. By utilizing the content information, the mismatch between the same object among different frames is effectively reduced. The experimental results demonstrate that the tracker with the graph matching method tends to increase the valid correspondence of trajectories and candidate objects.

INDEX TERMS Text tracking, template matching, graph matching.

I. INTRODUCTION

Text in scene videos often carries rich semantic information, which has become an essential part of content-based video analysis and retrieval, wearable camera systems and augmented reality translators, among others.

In recent years, the tracking-by-detection paradigm [1]–[4] has become the popular multi-object tracking method in videos. Text tracking in videos based on tracking-by-detection frameworks can be regarded as a data association problem, namely, joining detection results in adjacent frames. In [5] and [6], the particle filtering method is used to find the correspondences of the text regions extracted by Discrete Cosine Transform (DCT)-based methods between adjacent frames, and the position and size information of text blocks is mainly utilized by various strategies. Minetto *et al.* [9] also use particle filtering tracking to track and improve performance using a Hungarian Algorithm to merge the detection and tracking results. In [7], a Maximally Stable Extremal Regions (MSERs) [8]-based method was also proposed to reduce false alarms by merging detection and tracking results.

However, most studies mainly focus on the feature distance of objects between the tracking tubelets and the candidate objects; we call them *structural features*. In text tracking tasks, low-level features, such as colour histograms or aspect ratios, are often used in traditional methods because of their low computational overhead. However, they are sometimes not sufficiently distinct to disambiguate candidate texts for a trajectory. As illustrated by Fig. 1, it can be difficult to discriminate texts with similar colour, font, and size. On the other hand, scene texts in videos are usually rigid, and their relative positions, as well as some other *appearance features*, can be used to mitigate the tracking difficulty.

Based on this key observation, we introduce the idea of graph matching [10] into our tracking system, for which the structural similarities among multiple objects are modeled. Specifically, we regard the existing trajectories and newly detected candidate texts as two graphs, and the trajectories

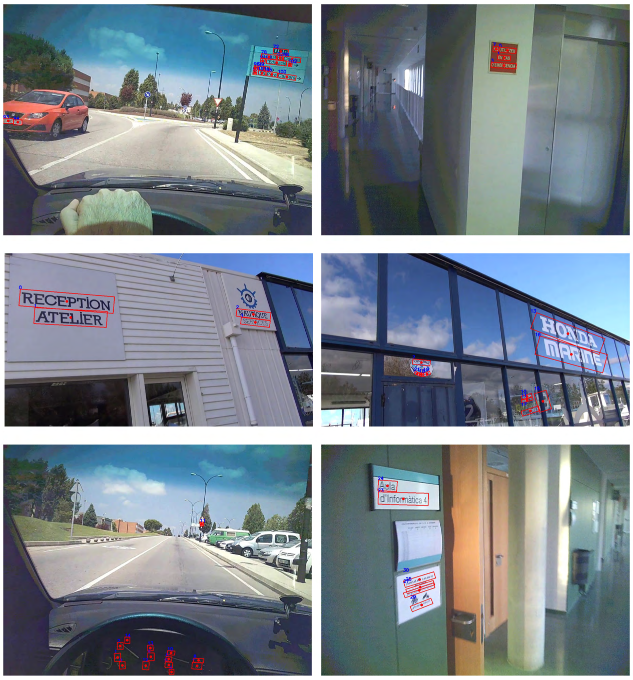


FIGURE 1. Examples of tracking multiple text blocks in a video, covering both indoor and outdoor cases. In each frame, the multiple text blocks are similar to each other, rendering reliable tracking over frames challenging. Thus, we explore another stable source of information, i.e., the spatial layout of the text blocks, which can be represented by a graph. As a result, graph matching can be used for text object association among frames.

and the candidates are regarded as the nodes in each graph. Meanwhile, we consider not only the node-to-node relationships (structural features) but also the edge-to-edge relationships (appearance features) using the graph matching model.

More concretely, in this work, we first design features between two trajectories or two candidate texts for modeling the similarity between tracked objects. Then, without loss of generality, we adopt an existing and popular graph matching solver, i.e., Re-weighted Random Walk Graph Matching (RRWM) [11], to combine appearance and structural features while matching term trajectories and objects. In this way, the tracking system can better discriminate similar text candidates by appearance features and achieve better performances on the template matching step. Note that our approach is agnostic to the graph matching solver, the resulting advantage being that other off-the-shelf methods, e.g., [12], [13], can also be readily used in our framework.

Summarily, the main contribution of this work is that a novel method is proposed to combine graph matching with video text tracking whereby the affinity matrix for matching input is specially designed for the given text tracking problem. In particular, our approach is based on graph matching and thus can utilize off-the-shelf graph matching solvers. To the best of our knowledge, this is the first work for adapting graph matching to text tracking in video. Moreover, we show that graph matching can notably improve the text tracking performance, especially in terms of the Multi-Object Tracking Accuracy (MOTA) measurement.

The paper is organized as follows. Section II briefly reviews related work in the literature. Section III presents the main technical details of the proposed method. Section IV shows the experimental results with the corresponding discussions, and Section V concludes this paper.

II. RELATED WORK

Our work is closely related to the tracking-by-detection paradigm, which is aimed at associating and tracking the detection results in successive frames to form the trajectory of a single object [1], [7], [14]. This method can to some extent avoid the need for re-initialization, which is designated for the case in which the object is accidentally lost in some frames, as well as reduce the false alarm rate during detection. In the following, we discuss two closely related areas along this technique line: i) template matching for tracking and ii) graph matching. Note that few works adopt graph matching in video text tracking, while we attempt to bridge this gap and improve the tracking robustness by leveraging the more stable features for tracking, i.e., graph representation among tracking objects.

A. TEMPLATE MATCHING FOR TRACKING

For tracking-by-detection-based tracking methods, the template matching problem, which attempts to associate tracking trajectories with detected objects, is a challenging task. Many works focus on extracting more powerful local appearance features for tracking.

Zhen and Zhiqiang [15] fused multiple frame trajectory results and text detection results for static text. The Harris corner features of text are used to search the corresponding position in the current frame, and the Hausdorff distance of the current text and the reference text is used to measure the similarity. The current text is considered to be the same text when the distance is less than a given threshold. Wang *et al.* [16] calculated and compared the distribution of Canny edge [17] and Harris corner [18] features to facilitate the matching of text blocks in different frames. For scrolling text, they determine the direction and speed of the text by statistical analysis. However, the start and end frames of the text are not accurately determined by this method. Nguyen *et al.* [19] presented their detection performance when utilizing currently detected text blocks. In addition, the text blocks other text in the preceding and several subsequent continuous frames. The overlap of the text block in the current frame and the text block in the previous N frames is first used to remove the false locations. Rong *et al.* [20] also used a tracking-by-detection method to track text. Scene text character (STC) prediction by an MSER-based detector is used to optimize the constraints of trajectory search. Then, the optimized trajectory is used to guide text detection and reduce the effects of motion blur.

To achieve better performance, many of these methods designed complex (high-level) features for the template matching process. The scale-invariant feature transform (SIFT) [21] and Speeded-Up Robust Features (SURF) [22] are common feature detectors. In addition, in text tracking tasks, the recognition result can also be used. Zhou *et al.* [23] combined SIFT and mean shift to achieve a consistent tracking performance. Mikolajczyk and Schmid [24] compared many descriptors, such as shape context, steerable filters, and PCA-SIFT, and found that SIFT-based descriptors perform best. Yusufu *et al.* [25] proposed a tracking system that focuses on static and scrolling video captures and estimated the text-ending frame by SURF feature point numbers. However, these well-designed features, similar to SIFT and SURF, often suffer from greater time consumptions than do low-level features. Text in scene videos suffers from various noise sources due to illumination, distortion, perspective and motion blur. These noise sources may lead to some incorrect matching cases when using low-level or even high-level features. Even if the noise is very complicated, the relative position (one of low-level features) among texts is stable. In addition, the variation degree of two texts is also stable as a low-level feature. In our study, we find that these structural low-level features achieve good performance for identifying different scene text blocks in videos. This motivates us to introduce a graph matching technique to handle the appearance and structural features simultaneously.

On the other hand, one line of work focuses on designing more robust observation models. Kuo and Von Ramm [26] described the pedestrian motion trajectories by utilizing the discriminative appearance model. Milan *et al.* [27] proposed a number of models (observation, appearance, dynamic models etc.) aiming at different cases in the tracking process. They further combined these models with a unified energy function. Zuo *et al.* [14] combined tracking by detection, spatial-temporal content learning and linear prediction into a multi-strategy tracking method. However, their method contains many hyper-parameters and is not robust in certain cases.

It remains challenging to discriminate similar text blocks using either well-designed features or ingenious models. As discussed in the introduction section, the structural features of the scene text can be more stable and differentiated. Therefore, we introduce these features and use the graph matching techniques to solve this problem. In the following, we review recent works on graph matching.

B. GRAPH MATCHING

Graph matching (GM) refers to the task of determining a mapping among the nodes of graphs that preserves the relationships between the nodes as much as possible; this has been a long-standing problem due to its inherent NP-hardness. Many ad-hoc and approximate algorithms have been devised. In general, one can divide graph matching methods into two scenarios. The first scenario involves two-graph matching, which is the focus of this paper. The second scenario concerns the case in which there are multiple (more than two) graphs for joint matching, and we refer the readers to a line of such work [28]–[31] and the references therein. However, they are beyond the scope of this paper.

In our case, the text block tracking process for videos corresponds to the pairwise GM problem: matching the first graph, consisting of multiple text blocks in one frame, to the other graph in the next frame with newly detected text block candidates.

The pairwise GM problem can be formulated as the Lawler's quadratic assignment problem (QAP) [32]. Given two graphs $G^1 = \{V^1, E^1, A^1\}$ and $G^2 = \{V^2, E^2, A^2\}$ of size n_1 and n_2, where V is the node set, E represents the edge set, and A denotes the attribution set, one can define an affinity matrix $\mathbf{M} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$, in which $M_{ia;jb}$, $(i, j = 1, \ldots, n_1)$, $(a, b = 1, \ldots, n_2)$, represents the relationship between the edges of the two graphs (v_i, v_j) and (v_a, v_b). The elements on the diagonal of $M_{ia;ia}$ represent the unary features of nodes v_i and v_a, while the elements on the off-diagonal are the affinity values between two edges from the two graphs.

Many works have been devoted to the affinity matrix. Cour *et al.* [33] proposed a spectral relaxation technique for approximate solutions to one-to-one and one-to-many matching problems. Leordeanu *et al.* [34] optimized in the discrete domain the quadratic score based on climbing and convergence properties. Some other works have obtained the optimal affinity matrix by machine learning methodologies. Caetano *et al.* [35] regarded the graph matching problem as labeling the pairs from graphs 'yes' or 'no'. More recently, Hu *et al.* [36] utilized a first-order compatibility term and converted the problem into a semi-supervised learning paradigm. Readers are referred to [10] for a more comprehensive literature review on recent advances in graph matching. However, these developments are orthogonal to our work, as we devise a mechanism to reuse off-the-shelf GM solvers in an out-of-the-box fashion. We regard this as one advantage of our approach.

Although there is a rich literature on template-based tracking and graph matching, these two areas are relatively separate from each other. In this paper, we want to focus our attention on these two communities by incorporating graph matching techniques into the text tracking task. In the next section, we will present our main method and demonstrate its efficacy in our empirical study.

III. PROPOSED METHOD

In this section, we propose a text tracking method with graph matching. The method performs detecting by tracking multiple text blocks frame by frame using *template matching*, *object prediction*, *trajectory initialization* and *trajectory elimination*.

A. TRACKING PIPELINE

First, a text detector, e.g., [37], is used to detect the text blocks in a new frame, and the tracker generates new prediction blocks through the existing trajectories. The detection blocks and the prediction blocks consist of the text block candidates. Then, the graph matching method is used to associate the existing trajectories and the object candidates. Then, a new trajectory is created when a detected object does not match

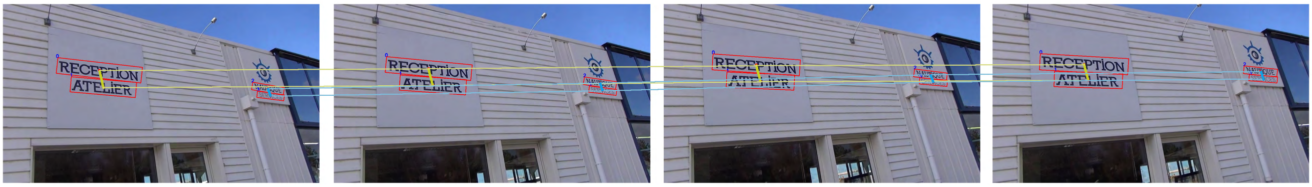


FIGURE 2. When the candidate text blocks have very similar appearances in terms of *font, colour and size*, the structural features (the *relative distance* in this case) can be discriminative and stable, as indicated by the yellow and light blue lines: the relative position of the text blocks is not changed with the changes in the positions of the individual text blocks over four consecutive frames.

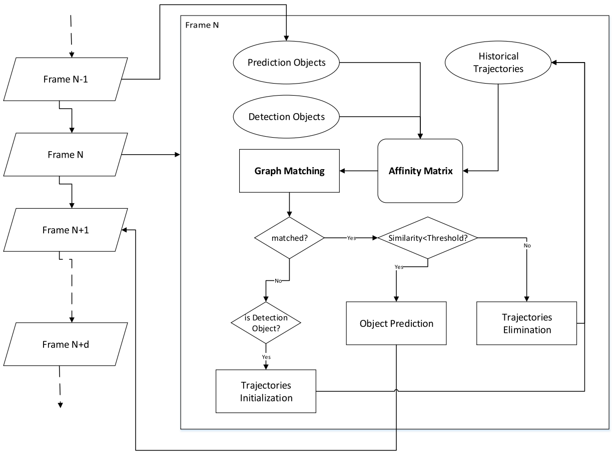


FIGURE 3. The flowchart of our tracking system. We generate the affinity matrix by extracting the *structural features*. Then, we use graph matching to associate the historical trajectories with the object candidates.

any existing trajectory. In addition, a trajectory ends when the similarity with the matched candidate falls below a given threshold. Finally, we use a text discriminator to determine whether a trajectory is a real text block and drop these false cases below a text confidence level.

As illustrated in Fig. 3, the affinity matrix (described in Section III-B) is generated by extracting the *structural features* from the detected objects, predicted objects and historical trajectories. Then, a few-to-many graph matching method is used to associate the candidate objects and the historical trajectories. For the matched terms, a given threshold is used to adjust whether the trajectories should continue tracking or stop. For the other terms, the detected objects will initialize new trajectories. Finally, the non-text trajectories are filtered by a text discriminator.

In this paper, we mainly discuss the influences of the structural features in the template matching step, and we use the graph matching method to solve this problem. Here, we focus on the extraction of the structural features for building up the input affinity to the graph matching solvers.

We used a graph matching method to match text blocks between new candidate texts in the current frame and the existing trajectories. The purpose of graph matching is to determine correspondences between two graphs, including nodes and edges. It is used to solve the fundamental problem of obtaining a mapping between two sets of nodes with low-level features and subsequently for object tracking. We choose the Re-weighted Random Walk Graph Matching (RRWM) method [11] to solve the problem of matching candidate objects. The method is robust to noise and outliers. Thus, it can solve the problem presented by matching errors caused by the similarities between individual features. The position and colour histogram information is used as a node of the text block feature graph.

If a text block in the current frame does not match any existing trajectories, a new trajectory is initialized, and the text block is connected to the start of a trajectory. On the other hand, if a text block is matched to one trajectory, the blocks in the current frame are also connected to it. The trajectory is only valid if the length of the trajectory is greater than the given threshold; otherwise, the trajectory is invalid and discarded as noise.

To improve the tracking performance, we use a linear prediction method on each trajectory. We calculate the similarity between the previous text blocks of the trajectory and the tracking output, and when the similarity is greater than a given threshold, the tracking output is associated with the trajectory.

B. GRAPH MATCHING FOR TRACKING

As discussed above in Section II-B, an affinity matrix $\mathbf{M} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ is defined to describe the relationship between two graphs $G^1 = \{V^1, E^1, A^1\}$ and $G^2 = \{V^2, E^2, A^2\}$. $M_{ia;jb}$ is the similarity of the edge $e^1_{i,j}(v^1_i, v^1_j)$ and $e^2_{a,b}(v^2_a, v^2_b)$ corresponding to the attribute vector $\mathbf{a}^1_{i,j} \in A^1$ and $\mathbf{a}^2_{a,b} \in A^2$, as depicted in Fig. 4.

Various structural features enjoy strong invariance under video view shifting. Fig. 2 gives an example, wherein the relative center position (a structural feature) of two text blocks remains stable while the texts are moving. Meanwhile, these features are usually simple calculations and require low time consumption. Due to such advantages, the use of graph matching to combine traditional appearance and structural features can improve the performance of template matching. In our tracking system, we designed a series of this type of feature. Given a graph $G = \{V, E, A\}$, $A_{i,j} = \{bBox_i, bBox_j, O_i, O_j, hist_i, hist_j, f^{dArea}_{i,j}, f^{dX}_{i,j}, f^{dY}_{i,j}, f^{dHist}_{i,j}\}$ is the attribute of $E_{i,j}$, where $bBox_i$ and $bBox_j$ are the bounding boxes of two text blocks, O_i and O_j are the central point of the two text blocks, $hist_i$ and $hist_j$ are the colour histograms of the texts on the gray scale, and $f^{dArea}_{i,j}$, $f^{dX}_{i,j}$, $f^{dY}_{i,j}$, and $f^{dHist}_{i,j}$

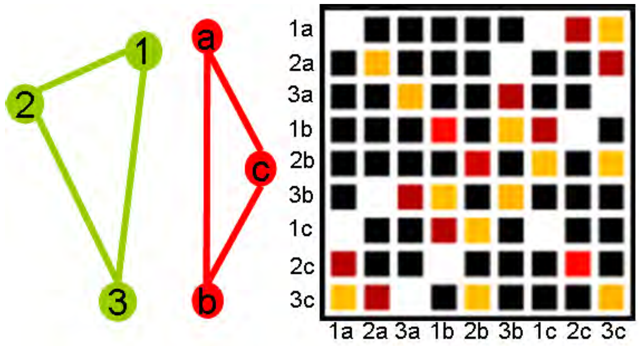


FIGURE 4. Illustration for affinity matrix of two graph [29]. The affinity matrix of two graphs, both of these graphs with 3 nodes. The elements on diagonal stand for the node-to-node similarities. While the elements off-diagonal are the affinity value between two edges from the two graphs respectively.

are the structural features. They are defined as follows:

$$f_{i,j}^{dArea} = \frac{Area_i}{Area_j} \tag{1}$$

$$f_{i,j}^{dX} = O_i.x - O_j.x \tag{2}$$

$$f_{i,j}^{dY} = O_i.y - O_j.y \tag{3}$$

$$f_{i,j}^{dHist} = \sqrt{1 - \frac{1}{\sqrt{\bar{hist_i} \cdot \bar{hist_j}}} \sum_k \sqrt{hist_i(k) \cdot hist_j(k)}} \tag{4}$$

Then, we define the similarity $M_{ia;jb}$ of the edges between two graphs:

$$M_{ia;jb} = sArea * sX * sY * sHist \tag{5}$$

where $sArea, sX, sY, sHist$ are the similarities of each attribute, defined as follows:

$$sArea(f_{i,j}^{dArea}, f_{a,b}^{dArea}) = \frac{min(f_{i,j}^{dArea}, f_{a,b}^{dArea})}{max(f_{i,j}^{dArea}, f_{a,b}^{dArea})} \tag{6}$$

$$sX(f_{i,j}^{dX}, f_{a,b}^{dX}) = \exp(-\sqrt{\left|f_{i,j}^{dX} - f_{a,b}^{dX}\right|}) \tag{7}$$

$$sY(f_{i,j}^{dY}, f_{a,b}^{dY}) = \exp(-\sqrt{\left|f_{i,j}^{dY} - f_{a,b}^{dY}\right|}) \tag{8}$$

$$sHist(f_{i,j}^{dHist}, f_{a,b}^{dHist}) = \frac{min(f_{i,j}^{dHist}, f_{a,b}^{dHist})}{max(f_{i,j}^{dHist}, f_{a,b}^{dHist})} \tag{9}$$

As $M_{ia;jb}$ is calculated using the above equation, the affinity matrix **M**, which combines appearance and structural information, is generated, and then, the graph matching algorithm is used to obtain the correspondence of trajectories and new text candidates.

Specifically, as mentioned above, we adopt RRWM [11] for solving the two-graph matching problem, which is a pairwise graph matching method. By iteratively updating and exploiting the confidences of candidate correspondences, it maintains a high accuracy when adding noisy and achieves state-of-the-art performance among graph matching algorithms.

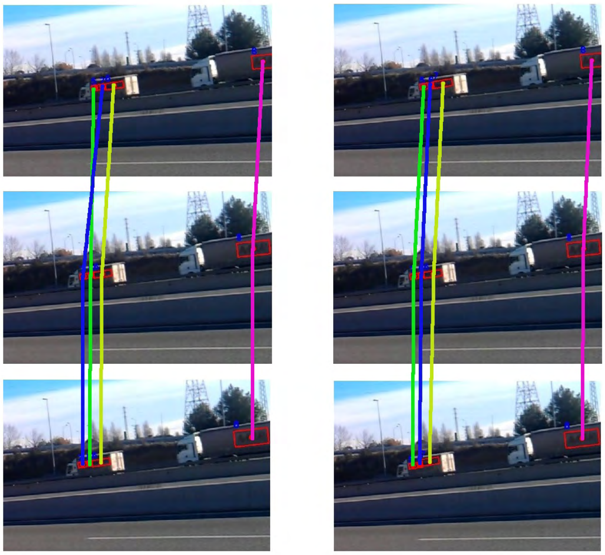


FIGURE 5. As shown by the above three frames, a situation whereby texts move fast, resulting in mismatching cases in the tracking system, is common. The left column is the result of Zuo's [14] tracker, while the right column is the result of the proposed method. Note that there is a mismatching in the left column, whereas our tracker performs consistently well.

IV. EXPERIMENTS AND DISCUSSION

A. DATASET AND EVALUATION PROTOCOL

1) ICDAR 2015 DATASET

Our experiments are performed on the ICDAR 2015 dataset[1] (Robust Reading Competition Challenge 3: Text in Videos), which contains a training set of 25 videos (13 and 450 frames in total) and a test set of 24 videos (14,374 frames in total). The dataset was collected from different countries and selected according to the standard of representing typical real-life applications and covering indoor and outdoor scenarios. In addition, four different cameras were used for different sequences to cover a variety of possible hardware uses.

2) MINETTO'S DATASET

The Minetto's dataset [9] contains 5 typical videos: i) text moving horizontally, ii) text with noise from a shadow, iii) concentrated text blocks, iv) text with perspective transformation, and v) text with noise from illumination. Specifically, the first video (*Bateau*) is about a scene of two text blocks on an embankment. In addition, the text blocks move horizontally as the camera moves. In this video, which has 800 frames, 2 text blocks appear in total. The second video (*Bistor*) includes two videos of text blocks on a parasol. The text blocks are affected by hard illumination changes and occlusion. The video includes 1089 frames and shows 2 text blocks. The third video (*Cambronne*) is at the crossroads. There are many text blocks in traffic signs, shop signs and billboards. The text blocks are concentrated and difficult to distinguish. The video contains 226 frames, and 18 text blocks appear in the video. The fourth video (*Navette*)

[1] Website: http://rrc.cvc.uab.es/?ch=3&com=introduction, accessed in April 2017.

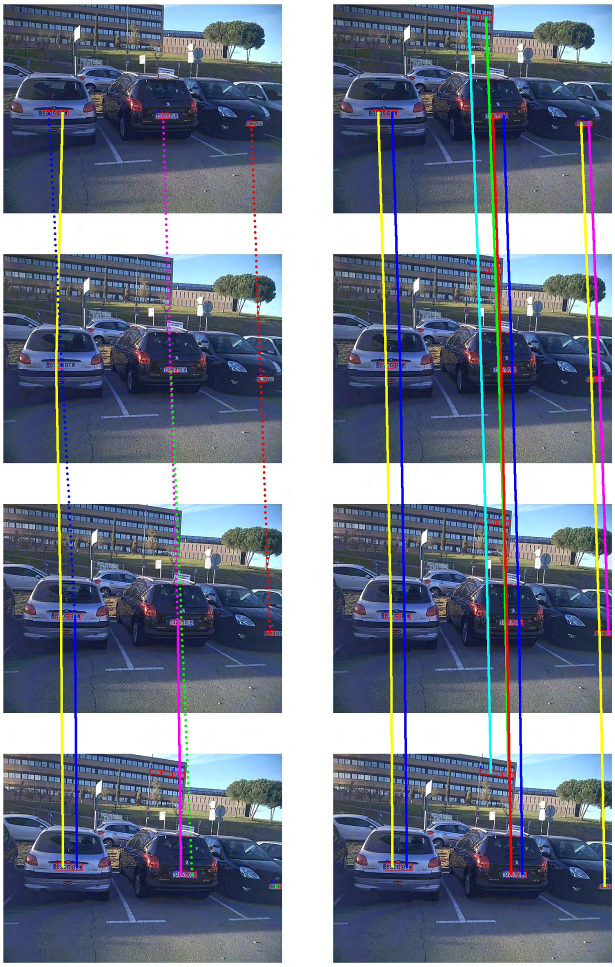
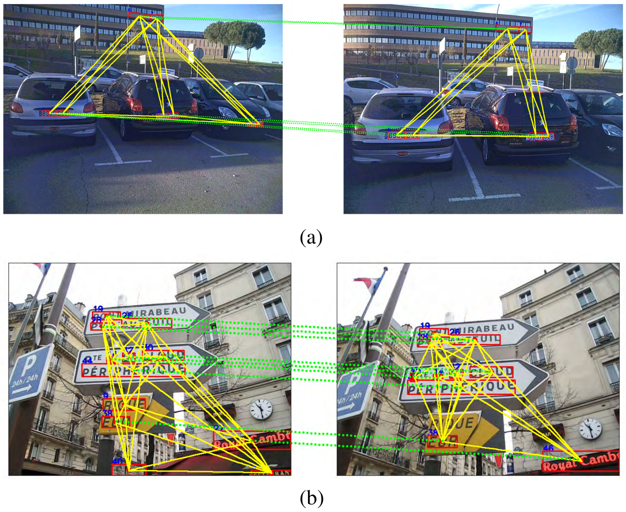


FIGURE 6. **This is another situation found in the tracking process. The left column is the result of Zuo's [14] tracker, while the right column is the result of the proposed method. The dotted lines indicate that the candidates do not match the trajectories from the most recent frame, which means that tracking was lost in the previous frames. In contrast, our tracker can robustly and consistently track the text blocks without any target ID switching.**

is about a situation where a yacht with text signs moves far away. The text blocks sometimes experience perspective transformations. The video contains 400 frames and have 3 text objects in total. The last video (*Zara*) is about the Zara store's signboard. The text blocks are affected by hard illumination changes and the irregular movement of the cameras. This video consists of 1250 frames, and 1 text block appears in total.

3) EVALUATION METRICS

We follow the evaluation metrics used in [38], which is widely adopted in the text tracking community. Specifically, the MOTP (Multi-Object Tracking Precision) measures the deviation of the tracking objects to the real objects, and the MOTA (Multi-Object Tracking Accuracy) measures the tracking trajectories to the real trajectories. This means that the MOTA considers not only the position errors of the objects but also the semantic mistakes, including mismatches and broken trajectory.



(a)

(b)

FIGURE 7. **Two examples of graph matching. (a) is the matching status from the 2nd frame to the 10th frame for ICDAR15' Video_9. (b) is the matching status from the 76th frame to the 80th frame of Cambronne. In each frame, the yellow lines denote the graph created by the nodes (text blocks), where the nodes are fully connected. The green dotted lines denote the matching status.**

B. RESULTS AND DISCUSSIONS

On the ICDAR 2015 dataset, we divide our experiment into two parts. For the first part, we use the ground truth as the detection result to observe the tracking performance in a non-noisy environment. We compare our proposed method to Zuo's method [14], which represents the state of the art. For the second part, we use a multi-orientation scene text detector, as presented in [39], to evaluate the tracking performance in a noisy environment. The comparison methods are the published methods on the official ICDAR Robust Reading Competition website.

TABLE 1. **Experimental results on ICDAR15's dataset. 'GT' stands for using the ground-truth text block areas as the tracking targets. 'Det' denotes using the text detection result by the state-of-the-art detector [37]. Note that the methods 'Deep2Text I', 'Deep2Text II', 'AJOU', and 'StradVision-1' are from the public ICDAR15 website, with no references disclosed.**

Method	MOTP	MOTA
Zuo's method (GT) [15]	81.35	72.96
Proposed Method (GT)	81.26	**81.30**
Deep2Text I	71.01	40.77
Deep2Text II	71.33	49.33
AJOU	**73.25**	53.45
StradVision-1	70.82	47.58
Proposed Method (Det)	72.11	**58.19**

In Table 1, the first two rows are the experimental results of the tracking method, where we use ground truth text blocks as the detection result, and the next five rows are the experimental results of the trackers, which combine the feature detector in [39] and the proposed tracking method. From the results, one can observe that the proposed method in general achieves a higher MOTA score and obtains almost the same MOTP score. The higher MOTA score suggests that the cases of ID

TABLE 2. Experimental results on Minetto's dataset. Here, GM stands for our graph-matching-based method, Zuo refers to the state-of-the-art tracker [14], and IDS means ID switches.

Videos	MOTP		MOTA		IDS	
	GM	Zuo	GM	Zuo	GM	Zuo
Bateau	0.7607	0.7604	0.7483	0.7457	14	15
Bistor	0.8219	0.8226	0.9710	0.9663	3	5
Cambronne	0.6985	0.6985	**0.4346**	0.3786	**46**	81
Navette	0.6544	0.6539	0.3671	0.3636	0	0
Zara	0.7178	0.7178	0.3645	0.3645	0	0
Avg.	0.7307	0.7307	0.5771	0.5637	12.6	20.2

switch (IDS) are significantly fewer in number than under the competing methods. We believe this is because our tracker can better capture the global layout of text blocks in videos, which leads to more stable tracking trajectories. We further use Fig. 5 to illustrate this advantage, where the texts on trucks move very fast. For the same position, text in one frame will be changed to other text in the next frame. Therefore, an ID switch occurs by Zuo's method. Our graph-matching-based method shows a stable tracking capability throughout the sequence from better exploring the outer information. Moreover, as shown in Fig. 6 and Fig. 7, our method achieves a higher recall than the comparison method using the same detector. In particular, for the case with two text blocks in one license plate, many mismatch mistakes occur under the comparison method, and the proposed method preforms well due to the structural features.

We also evaluate the method on Minetto's dataset [9]. The peer methods are i) Zuo: the state-of-the-art tracking method [14] with a detector presented in [37] and ii) GM (graph matching): the proposed tracking method with the same detector in [37] for fair comparison.

In Table 2, the performances of the two methods are quite similar, except on the third video: *Cambronne*. The MOTA and IDS measurements under our method are much better than those under the comparison method. In our analysis, this is because the third video involves more than 15 text blocks for tracking in one scene, and the other videos only have 3 or fewer text blocks. This requires a tracker with a better ability to discriminate among different text blocks. Because of the introduction of the structural features, our tracker can better distinguish different text blocks even though the text blocks share similar appearance features.

V. CONCLUSION AND FUTURE WORK

In this work, we have presented a novel scene video text tracking approach that incorporates graph matching techniques. The underlying rationale is that current text tracking methods mainly explore different local text block features to achieve effective tracking, whereas our method turns to the idea of graph matching, where the global geometrical layout of the text blocks in one frame can be more effectively accounted for. As a result, our approach is more robust, and in particular, the Multi-Object Tracking Accuracy (MOTA) measurement can be improved notably. One possible future issue is to investigate different feature extraction strategies (e.g., Principle Component Analysis, and Neutral Vector Variables [40]) for graph matching. Another future topic will explore the adaptation of multiple graph matching techniques by involving multiple frames for off-line tracking.

ACKNOWLEDGMENTS

Wei-Yi Pei and Chun Yang contributed equally to this work.

REFERENCES

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

[2] C. Yang *et al.*, "Tracking based multi-orientation scene text detection: A unified framework with dynamic programming," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3235–3248, Jul. 2017.

[3] S. Tian, X.-C. Yin, Y. Su, and H.-W. Hao, "A unified framework for tracking based text detection and recognition from Web videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 542–554, Mar. 2018, doi: 10.1109/TPAMI.2017.2692763.

[4] 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.

[5] M. Tanaka and H. Goto, "Text-tracking wearable camera system for visually-impaired people," in *Proc. 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[6] H. Goto and M. Tanaka, "Text-tracking wearable camera system for the blind," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 141–145.

[7] L. Gómez and D. Karatzas, "MSER-based real-time text detection and tracking," in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 3110–3115.

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

[9] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "Snoopertrack: Text detection and tracking for outdoor videos," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 505–508.

[10] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2016, pp. 167–174.

[11] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 492–505.

[12] Y. Tian, J. Yan, H. Zhang, Y. Zhang, X. Yang, and H. Zha, "On the convergence of graph matching: Graduated assignment revisited," in *Computer Vision—ECCV*, A. Fitzgibbon, S. Lazebnik, P. Perona, S. Sato, and C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 821–835.

[13] J. Lee, M. Cho, and K. M. Lee, "Hyper-graph matching via reweighted random walks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1633–1640.

[14] Z. Zuo, S. Tian, W. Pei, and X. Yin, "Multi-strategy tracking based text detection in scene videos," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, 2015, pp. 66–70.

[15] W. Zhen and W. Zhiqiang, "An efficient video text recognition system," in *Proc. 2nd Int. Conf. Intell. Human-Mach. Syst. Cybern.*, vol. 1. 2010, pp. 174–177.

[16] B. Wang, C. Liu, and X. Ding, "A research on video text tracking and recognition," *Proc. SPIE*, vol. 8664, p. 86640G, Mar. 2013.

[17] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[18] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 1–6.

[19] P. X. Nguyen, K. Wang, and S. Belongie, "Video text detection and recognition: Dataset and benchmark," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 776–783.

[20] X. Rong, C. Yi, X. Yang, and Y. Tian, "Scene text recognition in multiple frames based on text tracking," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2014, pp. 1–6.

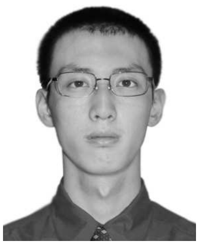
[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer Vision—ECCV*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 404–417.

[23] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Comput. Vis. Image Understand.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.

[24] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[25] T. Yusufu, Y. Wang, and X. Fang, "A video text detection and tracking system," in *Proc. Int. Symp. Multimedia*, 2013, pp. 522–529.
[26] J. Kuo and O. T. V. Ramm, "Three-dimensional motion measurements using feature tracking," *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 55, no. 4, pp. 800–810, Apr. 2008.
[27] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.
[28] J. Yan, Y. Tian, H. Zha, X. Yang, Y. Zhang, and S. M. Chu, "Joint optimization for consistent multiple graph matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1649–1656.
[29] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, "Consistency-driven alternating optimization for multigraph matching: A unified approach," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 994–1009, Mar. 2015.
[30] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. M. Chu, "Graduated consistency-regularized optimization for multi-graph matching," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 407–422.
[31] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, Jun. 2016.
[32] E. L. Lawler, "The quadratic assignment problem," *Manage. Sci.*, vol. 9, no. 4, pp. 586–599, 1963.
[33] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *Proc. 20th Annu. Conf. Neural Inf. Process. Syst.*, 2006, pp. 313–320.
[34] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and MAP inference," in *Proc. 23rd Annu. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1114–1122.
[35] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, "Learning graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1048–1058, Jun. 2009.
[36] N. Hu, R. M. Rustamov, and L. J. Guibas, "Graph matching with anchor nodes: A learning approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2906–2913.
[37] W. He, X. Zhang, F. Yin, and C. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2017, pp. 745–753.
[38] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, p. 246309, May 2008.
[39] 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.
[40] Z. Ma, J.-H. Xue, A. Leijon, Z.-H. Tan, Z. Yang, and J. Guo, "Decorrelation of neutral vector variables: Theory and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 129–143, Jan. 2018.

WEI-YI PEI received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2010, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include text detection, object tracking, and pattern recognition.

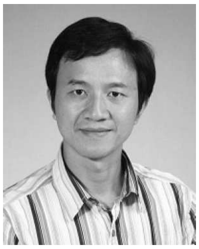


CHUN YANG received the B.Sc. and Ph.D. degrees in computer science from the University of Science and Technology Beijing, China, in 2011 and 2017, respectively. He is currently a member of the Faculty with the School of Computer and Communication Engineering, University of Science and Technology Beijing. He has authored over ten research papers, including the IEEE TIP, PLoS ONE, *Information Fusion*, ICDAR, and ICPR. His research interests include pattern recognition, classifier ensemble, and document analysis and recognition.

LI-YU MENG received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2016. She is currently pursuing the degree with the Department of Computer Science and Technology, University of Science and Technology Beijing. Her research interests include video text detection, tracking, and recognition.

JIE-BO HOU received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include text detection, pattern recognition, and deep learning.

SHU TIAN received the B.Sc. and Ph.D. degrees in computer science from the University of Science and Technology Beijing, China, in 2010 and 2016, respectively. He is currently a member of the Faculty with the School of Computer and Communication Engineering, University of Science and Technology Beijing. He has authored about ten research papers, including the IEEE TPAMI, the IEEE TIP, *Neurocomputing*, IJCAI, ICDAR, and IJCNN. His research interests include object tracking, pattern recognition, and multimedia understanding.



XU-CHENG YIN (M'10–SM'16) received the B.Sc. and M.Sc. degrees in computer science from the University of Science and Technology Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2006. From 2006 to 2008, he was a Researcher with the Information Technology Laboratory, Fujitsu Research and Development Center. He was a Visiting Researcher with the School of Computer Science, University of Massachusetts Amherst, USA, from 2013 to 2014. He was a Visiting Professor with the Department of Quantitative Health Sciences, University of Massachusetts Medical School, USA, in 2016. He is currently a Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing.

He has authored over 50 research papers, including the IEEE TPAMI, the IEEE TIP, *Information Fusion*, *Information Sciences*, IJCAI, SIGIR, CIKM, ICMR, ICDAR, and ICPR. His research interests include pattern recognition, computer vision, image processing, information retrieval, and document analysis and recognition. His team won the first place for both Text Localization in Real Scenes and Text Localization in Born-Digital Images in the ICDAR 2013 Robust Reading Competition, the first place for both End-To-End Text Recognition in Real Scenes (Generic) and End-To-End Text Recognition in Born-Digital Images (Generic) in the ICDAR 2015 Robust Reading Competition, and the first place for Challenge on COCO-Text (End-to-End Text Recognition) in the ICDAR 2017 Robust Reading Competition.