This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

A Unified Framework for Tracking Based Text Detection and Recognition from Web Videos

Shu Tian, Xu-Cheng Yin*, *Senior Member, IEEE,* Ya Su, *Member, IEEE,* and Hong-Wei Hao

Abstract—Video text extraction plays an important role for multimedia understanding and retrieval. Most previous research efforts are conducted within individual frames. A few of recent methods, which pay attention to text tracking using multiple frames, however, do not effectively mine the relations among text detection, tracking and recognition. In this paper, we propose a generic Bayesian-based framework of Tracking based Text Detection And Recognition (T^2DAR) from web videos for embedded captions, which is composed of three major components, i.e., text tracking, tracking based text detection, and tracking based text recognition. In this unified framework, text tracking is first conducted by tracking-by-detection. Tracking trajectories are then revised and refined with detection or recognition results. Text detection or recognition is finally improved with multi-frame integration. Moreover, a challenging video text (embedded caption text) database (USTB-VidTEXT) is constructed and publicly available. A variety of experiments on this dataset verify that our proposed approach largely improves the performance of text detection and recognition from web videos.

Index Terms—Video text extraction, text tracking, tracking based text detection, tracking based text recognition, embedded captions.

1 Introduction

The explosive growth of smart phones and the online social media have led to the accumulation of large amounts of visual data, in particular, the massive and increasing collections of video on the Internet and social networks. These countless web videos have triggered research activities in multimedia understanding and video retrieval, where text in video contains valuable information and is exploited in widespread content-based video applications.

In the literature, a wide variety of approaches have been proposed for text detection and recognition in video [1]–[4]. On the one hand, the most direct and simple way is to recognize video text as the same as the one in static images, i.e., to recognize text with frame by frame. Hence, conventional video text extraction techniques mainly focus on detecting and recognizing text in each individual frame or some key frames, without multi-frame integration [1], [4]. On the other hand, spatial and temporal information is very important for multimedia understanding of complex videos. Consequently, there are also some video text detection and recognition methods with tracking techniques using multiple frames [4]. Most existing tracking based text detection and recognition methods can be roughly categorized into temporal-spatial based methods and fusion based ones. The former methods use temporal or spatial information to remove noises for detection or to enhance the images for recognition [5], [6]. The latter ones merge detection and tracking results, or recognition and tracking results over multiple frames [7], [8]. However, the feedback between tracking and detection or recognition is always ignored. How to effectively utilize the relations and interactions between tracking and detection or recognition is challenging for text extraction from complex videos. Moreover, previous methods either tackle these two tasks separately (tracking based text detection, and tracking based text recognition), or treat them as sequential stages in a stepwise strategy [9].

In this paper, a generic Bayesian-based framework of Tracking based Text Detection And Recognition (T^2DAR) for embedded caption text is first proposed, which performs both tracking based text detection and tracking based text recognition in a single unified pipeline. In general, the feedback information between tracking-detection and tracking-recognition in complex videos is challenging for exploiting and sharing. In this work, a unified formulation of both tracking based text detection and tracking based text recognition is designed within a Bayesian framework.

Next, a novel tracking-by-detection approach is introduced for text tracking. In this approach, the appearance model for region matching and the motion model for text tracking are adaptively designed and utilized to link the detections into trajectories in consecutive frames. A tracking based text detection method is then proposed, where two techniques are performed to retrieve the missing detections and to tune the scales of detection regions

- *S Tian and Y. Su are with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China.*
- *X.-C. Yin is with the Department of Computer Science and Technology, and also with 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 (email: xuchengyin@ustb.edu.cn).*
- *H.-W. Hao was with the Research Center of Digital Technologies, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

(a) Layered caption text (b) Embedded caption text (c) Scene text

Fig. 1. Text in video: (a) Layered caption text, (b) embedded caption text, and (c) scene text, where embedded caption text is common and challenging for detection, tracking and recognition from web videos.

respectively. We also design a tracking based video text recognition approach. To address inevitable text ID switch errors for tracking current objects, the tracking trajectories are temporally over-segmented to ensure that the detections of sub-trajectories are corresponding to the same text. An agglomerative hierarchical clustering algorithm is afterwards used to merge over-segmented trajectories and construct suitable trajectories. At last, a voting strategy is conducted to obtain the final recognition results.

We also collect and release a large database (USTB-VidTEXT) [1] for text detection and recognition from web videos (for embedded caption text). This database includes 5 web videos directly crawled from the Web, and each video sequence averagely includes 5534 frames. A variety of experiments on this dataset verify that our proposed approaches largely improves the performance of video text detection and recognition.

1.1 Embedded Caption Text in Web Videos

Following the method of categorization in [1], [4], text in video is categorized as caption or scene text (see examples in Figure 1). Caption text provides good directivity and a high-level overview of the semantic information in captions, subtitles and annotations of the video, while scene text is part of the camera-based images and is naturally embedded within objects (e.g., trademarks, signboards and buildings) in scenes. Caption text is further classified into layered caption text and embedded caption text, where layered caption text is always printed on a specifically designed background layer (see Figure 1(a)), and embedded caption text is overlaid and embedded on the frame (see Figure 1(b)). Generally speaking, scene text and embedded caption text are more challenging for detection, tracking and recognition. The embedded caption text appears more commonly in massive web videos with several typical challenges for detection and recognition, which is also the focus of our paper.

Firstly, the backgrounds of web videos are always complex (complex backgrounds, shown in Figure 2(a)), where a variety of scenes, objects and graphics are dynamically appeared and disappeared. Varied noises always follow text detection results. Secondly, the colors of text are always dynamically varied because of background changes and video compression (varied colors, shown in Figure 2(b)). That is to say, though the originally designed color of the superimposed text is a constant color, the actually displayed color of the caption text in consecutive frames has a wide range. Thirdly, the colors of the backgrounds and the foreground text are very similar in some cases (similar colors, shown in Figure 2(c)). For embedded captions, the text strokes are always interconnected with the background edges. Some characters and text regions are easily missed by the text detector. Fourthly, web video frames usually have a low resolution, where a major part of text is with low contrast and blur.

1.2 Contributions of this Paper

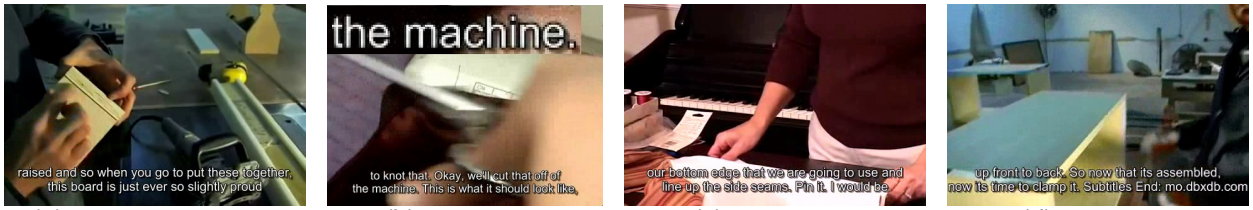
Summarily, there are four major contributions of this paper. The first contribution is a unified Bayesian-based framework for both tracking based text detection and tracking based text recognition from complex (web) videos, different from conventional methods only tackling one of these two tasks separately. To our best knowledge, this unified framework is proposed in the literature of document analysis and recognition for the first time. Moreover, this framework is presented within a Bayesian formulation for exploiting and sharing information between tracking and detection (tracking and recognition), different from conventional strategies using knowledge-based rules.

The second contribution is a novel tracking-by-detection approach for text tracking, where the appearance model for region matching and the motion model for text tracking are adaptively designed and utilized to link the detections into trajectories, different from conventional methods only focusing on region matching.

The third contribution is well-designed tracking based text detection and tracking based text recognition approaches. In tracking based text detection, feedback information between tracking and detection is exploited to retrieve the missing detections and to address the issue of varied text scales, different from most conventional methods simply using temporal or spatial information to remove noises. In tracking based text recognition, over-segmentation and hierarchical clustering are used to select and combine

1. http://prir.ustb.edu.cn/WebT2DAR/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) complex backgrounds (b) varied colors (c) similar colors (d) low contrast

Fig. 2. Challenges for detecting and recognizing embedded captions from web videos: (a) complex backgrounds with a mess of stuffs, (b) varied colors from image compression (zooming in for a part of text on the top), (c) similar colors between the foreground text and the background, and (d) low contrast and blur.

recognition results in multiple frames, different from conventional methods simply combining recognition results of different frames with heuristic rules.

The fourth contribution is a practical dataset for text detection and recognition from web videos. This annotated dataset includes 5 typical complex web videos directly crawled from the Web and is publicly available.

The rest of this paper is organized as follows. Related work is presented in Section 2. Section 3 introduces the the unified framework for both tracking based text detection and tracking based text recognition. Text tracking, tracking based text detection, and tracking based text recognition methods of our proposed system are described in Section 4, 5 and 6 respectively. Comparative experiments are demonstrated in Section 7. Final remarks are presented in Section 8.

2 Related Work

In this section, text detection and recognition methods in individual frames are first discussed briefly. Text tracking, tracking based text detection, and tracking based text recognition approaches (with multiple frames) are then reviewed in more details.

2.1 Text Detection and Recognition in Individual Frames

Text detection in an image or an individual frame of videos are fundamental to text tracking and recognition in video. Here, we review a few typical methods. Existing text detection methods can roughly be categorized into three groups, i.e., sliding window based methods [10]–[12], connected component based methods [13]–[16], and hybrid methods [17]. Sliding window based methods (also called region based methods) scan the input image by a scanning-window and decide whether the image patch is text or not. The computational burden of these methods are high due to the multiple scales of scanning-windows. Connected component based methods extract character candidates from images by connected component analysis followed by grouping character candidates into text, probably with additional checks to remove false positives. The hybrid method presented by Pan et al. [17] exploits a sliding window classifier to detect text candidates and then extracts connected components as character candidates.

Video text recognition is conventionally performed using existing OCR techniques; in other words, text regions are first segmented from video frames and then fed into a state-of-the-art OCR engine [1]. There are also some video text recognition methods by combining well-formulated OCR techniques with improved recognition approaches. For example, Elagouni et al. [9] reduced the ambiguities involved in character segmentation by considering character recognition results and introducing linguistic knowledge. Chen et al. [18] described a multiple-hypotheses framework and proposed a new gray-scale consistency constraint (GCC) algorithm to improve character segmentation. In our paper, without loss of generality, a state-of-the-art open source OCR engine, Tesseract [2], is used in our tracking based text recognition method.

2.2 Text Tracking

In recent years, a variety of text tracking methods have been investigated in the literature, most of which are related to tracking with detection, i.e., detected text positions are used to track text across consecutive frames. These methods can be further divided into several sub-groups based on the typical tracking strategies [4], i.e., text tracking with template matching, with particle filtering, and with tracking-by-detection.

The template matching method attempts to answer some variation of the following question: Does the image contain a specified view of some features and if so, where? As one of the conventional methods of text tracking, it implements tracking by seeking the most similar region in the image compared with the template image (patch). A reference image of the object, which is so-called a template, is extracted in the first frame, and then the blocks which are matched the template best by calculating the similarities between the candidates and the template with some metrics are found in the successive frames. Lienhart et al. presented a simple block matching algorithm in which the minimum mean absolute difference (MAD) is the matching criterion [19]. Li et al.'s method [20] also tracks the text block by a similar measure where the matching criterion becomes the sum of square difference (SSD). In Xi et al.'s approach [21], the mean square error (MSE) is used as the measure of the dissimilarity between two blocks. With the development of

2. https://github.com/tesseract-ocr.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

the technology of computer vision, a variety of complicated methods for feature representation are introduced into template matching, e.g., text contour [22], overlap rate between rectangles [23], sub-blocks with frame division [24], and Scale Invariant Feature Transform (SIFT) [25].

Particle filtering, also known as the Sequential Monte Carlo method, is a nonlinear filtering technique that recursively estimates a system's state based on the available observations. It is first introduced for single-object tracking [26], and then is applied to multi-object tracking [27], [28] successfully. It is also utilized for text tracking recently [29], [30]. How to adaptively apply particle filters for text tracking is still a challenge in the literature.

The tracking-by-detection method associates detection results across successive frames to create the appearances of objects, namely, estimating the tracking trajectories using detection results. Compared with other tracking methods, tracking-by-detection successfully solves the re-initialization problem even when the object is accidentally lost in some frames. It also avoids excessive model drifts due to the similar appearances of different objects. Rong et al.'s approach [31] tracks scene characters independently by the boosted particle filter [32] and uses an online-tracking model [33] to handle the variations of lighting and appearance. Nguyen et al.'s technique [7] detects characters and words firstly, then recovers some omitted text regions by exploiting temporal redundancy, and finally links the detections into tracking trajectories by verifying with a linear classifier. However, most conventional tracking-by-detection approaches only focus on region matching for text tracking. In this paper, a new tracking-by-detection method is proposed, where the appearance model for region matching and the motion model for text tracking are adaptively designed and utilized to link the detections into trajectories across consecutive frames.

2.3 Tracking Based Text Detection

Conventional video text detection methods mainly focus on detecting text in each individual frame or in some key frames. The important temporal information (temporal redundancy) in video is always neglected. Text tracking techniques with multiple frames are consequently introduced in the detection process to reduce false alarms and to improve the accuracy of detection by exploiting temporal redundancy. In Lienhart et al.'s method [34], the text regions are discarded when they occur less than a second or a more than 0.25 dropout rate. Similarly, some approaches discard the text trajectory of which the length is less than a specific threshold [35], [36]. On the contrary, other methods are proposed to generate a "clear" image by multiple frame integration and to detect text only on the generated images instead of all the original frames. For example, Wang et al.'s method [5] averages a base frame and its frontal and latter 10 frames to generate a "clear" image. Mi et al.'s approach [37] gets 30 consecutive frames in the middle of the trajectory to generate a synthesized image, which is produced by minimum/maximum pixel search on these consecutive frames.

To reduce the computational burden, a part of methods perform a text detector periodically and track the detection results in the frames which are not missed by the detector [38], [39]. For example, SnooperTrack [38] detects text periodically and tracks every new text by a particle filter. In the frames if both detection and tracking results are gotten, it then uses a merge strategy to generate the final results. Alternatively, some fusion based methods are proposed for tracking based text detection. Yin et al. [40], [41] proposed multi-strategy tracking based scene text detection methods, where tracking-by-detection, spatial-temporal context learning, and linear prediction are performed to predict the candidate text location respectively, and the best matching text block from the candidates are adaptively selected with a rule-based method [40] or a dynamic programming algorithm [41].

Most of the above tracking based text detection methods utilize a specific tracking technique to trivially track text and heuristically combine text detection results. These methods still have limited performance because of grand challenges (e.g., complex backgrounds, varied colors, similar colors and low contrast) for embedded caption text detection from web videos. In this paper, we propose a Bayesian-based framework for tracking based text detection, where feedback information between tracking and detection is adaptively exploited to retrieve the missing detections.

2.4 Tracking Based Text Recognition

In general, multiple frame integration techniques for video text recognition can be divided into two major categories: image enhancement, which integrates the same text region images (patches) to obtain a high-resolution image through techniques such as multi-frame averaging, time-based minimum/maximum pixel value searching, and so on, and recognition results fusion, which combines recognition results from different frames into a final text string.

Image enhancement technology uses selection-based or integration-based strategies to obtain a high-resolution image from the text regions that contain the same text. For example, in [21], [35], [36], a region whose horizontal length is the longest is selected. Tanaka et al.'s method [29] selects the image by calculating six features: area and width of text region, Fisher's discriminant ratio, the number of vertical edges, sum of the absolute values of the vertical components and the vertical edge intensity. The selection method performs not well when the image is blur. The so-called "super-resolution" method becomes one of the most popular methods to get better recognition results with multiple frames. Its key idea is to create a new "clear" image through calculation on multiple frames. The new "clear" image is more suitable to be recognized by OCR [42]. The frame averaging method is a simple multi-frame integration method. The "clear" image is obtained by calculating an average image of multi-frame [5], [43]. To address the noise problem, Hua et al.'s approach [44] selects the most likely clear images for averaging. Their assumption is that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

the blocks with high contrast are more likely to be clear. Zhen et al. [42] also used the similar strategy. Time-based minimum/maximum pixel value search is another "super-resolution" method [6], [45], [46]. The assumption is that the color of text varies only slightly, but the background is changing during the video, and the text is often the minimum or maximum pixel value of the image.

Recognition results fusion simply combines the text recognition results of different frames into one final character/text, which can generally improve the overall recognition performance. For example, Mita and Hori proposed a technique to select the best results of individual characters and to integrate them into a single text string [8]. Here, voting is a commonly used combining strategy. Lienhart et al.'s approach [47] segments the character regions and recognizes characters by an OCR software. As a result, multiple independent recognition results are corresponding to the same character. The final recognition result is constituted by the most frequent one.

Summarily, the feedback between text tracking and recognition is ignored in many conventional approaches. Moreover, for previous tracking based text detection or recognition methods, different strategies are always separately utilized to tackle these two tasks using knowledge-based rules. In this paper, a unified Bayesian-based framework is proposed for both tracking based text detection and tracking based text recognition from complex (web) videos for embedded captions for exploiting and sharing information between tracking and detection (tracking and recognition).

3 Unified Framework for Tracking Based Text Detection and Recognition

In general, the whole system of video text extraction includes several major components (e.g., Detection, Recognition and Tracking), and their relations and interactions (see Figure 3) [4]. Here, Detection is the task of localizing the text in each video frame with bounding boxes. Tracking is the task of maintaining the integrity of the text location and tracking text across adjacent frames. Recognition involves segmenting (if necessary) text and recognizing it using Optical Character Recognition (OCR) techniques. Obviously, Recognition is performed on text regions detected from Detection results (Detection-based-Recognition), and Tracking uses the locations identified in the Detection step to track text (Tracking-with-Detection). In general, Detection is first performed first in each frame independently; then, the detection results in sequential frames can be integrated and enhanced based on the Tracking results (Tracking-based-Detection). Similarly, Recognition can help verify the Tracking results (Refinement-by-Recognition for Tracking), and also confirm the Detection results in some cases (Refinement-by-Recognition for Detection). Meanwhile, Tracking can improve Recognition by fusing the recognition results over multiple frames (Tracking-based-Recognition).

In this paper, we propose a generic framework of Tracking based Text Detection And Recognition (T^2DAR) for embedded caption text, which performs both tracking based text detection and tracking based text recognition in a single unified pipeline. Consequently, the two major parts of the whole system, i.e., Tracking Based Text Detection, and Tracking Based Text Recognition, are focused and formulated into a unified Bayesian framework.

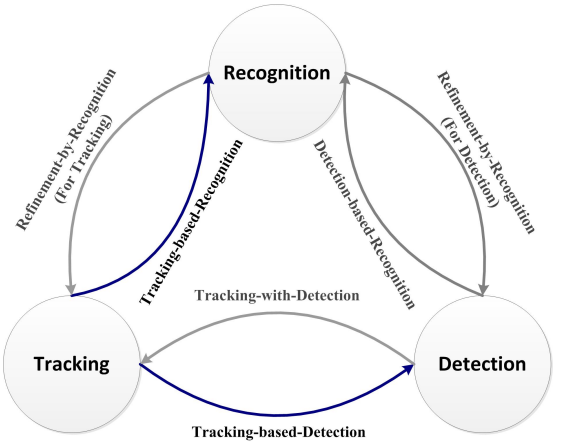


Fig. 3. A whole diagram for text detection, tracking and recognition in video [4], where tracking based text detection and tracking based text recognition are focused and formulated into a unified Bayesian framework in this paper.

3.1 Bayesian Formulation of Unified Framework

The general procedure of tracking based text detection (or recognition) using multiple frames can be described as follows. Given tracking trajectories from the prior frames and detection (or recognition) results in the current frame, more precise detection (or recognition) results will be obtained using the prior trajectories and the current detection (or recognition) results with text tracking. This procedure can be correspondingly formulated as a Bayesian process. Given the prior probability of a hypothesis (detection or recognition results) and the probability of observed data (prior trajectories), the Bayesian theorem provides a way to calculate the posterior probability of the hypothesis (detection or recognition results). It becomes a Maximum a posteriori (MAP) calculation problem. Consequently, our unified framework for tracking based text detection and recognition can be naturally described as a Bayesian formulation with MAP calculation.

Specifically, define all detections (detection results) in frame t as

$$D_t = \{d_{t,1}, d_{t,2}, ..., d_{t,m_t}\} \quad (1)$$

where $d_{t,i}$ is the i^{th} detection and m_t is the number of detections in the frame t. And all recognition results in frame t are marked as

$$R_t = \{r_{t,1}, r_{t,2}, ..., r_{t,m_t}\} \quad (2)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

where $r_{t,i}$ is the recognition result of the i^{th} detection. Then, define all trajectories generated from frame 1 to frame t as

$$\mathcal{T}_t = \{T_{t,1}, T_{t,2}, ..., T_{t,n_t}\} \tag{3}$$

where n_t is the number of trajectories in frame t and $T_{t,i}$ is the i^{th} trajectory which is composed by a temporal sequence of detection results,

$$T_{t,i} = \{d_{t,i,1}, d_{t,i,2}, ..., d_{t,i,p_{t,i}}\} \tag{4}$$

where $p_{t,i}$ is the length of the trajectory $T_{t,i}$ and $d_{t,i,j} \in \cup_{k=1}^{t} D_k$.

Here, a unified framework is proposed to improve detection and recognition results based on text tracking. In frame t, the outputs (D_t^0) of the text detector, the recognition results (R_t^0), and the trajectories ($\mathcal{T}_{t-1}^*$) generated from frame 1 to frame $t-1$ are supposed to be known. The goal is to find the most probable detections D_t^* and recognition results R_t^* (with MAP calculation) after revising and integration via text tracking.

Let $X = D$ (for tracking based text detection) or $X = R$ (for tracking based text recognition), the target is

$$X_t^* = \arg\max_{X_t} P(X_t | X_t^0, \mathcal{T}_{t-1}^*) \tag{5}$$

The key issue is how to calculate the posterior probability in Equation (5). The tracking step is correspondingly introduced. According to the Total Probability Theorem, we can get

$$P(X_t | X_t^0, \mathcal{T}_{t-1}^*) = \sum_{\mathcal{T}_t} P(X_t | \mathcal{T}_t, X_t^0, \mathcal{T}_{t-1}^*) P(\mathcal{T}_t | X_t^0, \mathcal{T}_{t-1}^*) \tag{6}$$

The most probable trajectories are obtained with

$$\mathcal{T}_t^{*,0} = \arg\max_{\mathcal{T}_t} P(\mathcal{T}_t | X_t^0, \mathcal{T}_{t-1}^*) \tag{7}$$

For simplicity, $\mathcal{T}_t^{*,0}$ are only used in a direct "successive calculation" style and the others are ignored. Hence, there are

$$\begin{cases} P(\mathcal{T}_t^{*,0} | X_t^0, \mathcal{T}_{t-1}^*) = 1 \\ P(\mathcal{T}_t | X_t^0, \mathcal{T}_{t-1}^*) = 0, \quad \text{if} \quad \mathcal{T}_t \neq \mathcal{T}_t^{*,0} \end{cases} \tag{8}$$

That is to say, if the optimal trajectories at $t-1$ ($\mathcal{T}_{t-1}^*$) are determined, the initial values ($\mathcal{T}_t^{*,0}$) of the optimal trajectories at t are correlated with $\mathcal{T}_{t-1}^*$ while the other trajectories have no correlation with them [3]. Correspondingly, Equation (6) can be simplified as

$$P(X_t | X_t^0, \mathcal{T}_{t-1}^*) = P(X_t | \mathcal{T}_t^{*,0}, X_t^0, \mathcal{T}_{t-1}^*) \tag{9}$$

The final detection or recognition results in frame t are then obtained with

$$X_t^* = \arg\max_{X_t} P(X_t | \mathcal{T}_t^{*,0}, X_t^0, \mathcal{T}_{t-1}^*) \tag{10}$$

Finally, the optimal trajectories in frame t ($\mathcal{T}_t^*$) are updated as follows,

$$\mathcal{T}_t^* = \arg\max_{\mathcal{T}_t} P(\mathcal{T}_t | X_t^*, \mathcal{T}_t^{*,0}) \tag{11}$$

These optimal trajectories will be used for optimization in frame $t+1$. The whole procedure of tracking based text detection (or tracking based text recognition) is finished after Equation (7), (9), (10) and (11) are sequentially and iteratively calculated.

In the following sections (Section 5 and 6), these processes of tracking based text detection and tracking based text recognition will be further described in more details, respectively.

4 Text Tracking with Tracking-by-Detection

As described before, there are several challenges for text detection and recognition from web videos. A better way for detecting and recognizing embedded captions is to use multiple frames, i.e., fusing text detection and recognition results from multiple consecutive frames with text tracking. Here, we argue that "tracking" is not only necessary but also important for detecting and recognizing embedded captions from web videos. In text tracking, a key step is to continuously determine the location of text across multiple frames. However, this procedure is not a trivial task though embedded caption text is seemingly still in the exact same position in each frame. Firstly, because of challenges with complex backgrounds and varied colors, text detection results (region locations) for one same caption text are not completely fixed (still), however, always with several-pixels translation. More importantly, caption text switching dynamically appears, always with the same background and similar text. It is really challenging to detect the change (cutting) of such captions in continuous frames. Moreover, because of challenges with complex backgrounds, varied colors, similar colors and low contrast, it is also not an easy task to identify two text regions in consecutive frames whether they are same or not by simple region matching techniques. As a result, "tracking" has been already introduced for detecting and recognizing caption text (even for layered caption text) in the literature.

Here, a novel tracking-by-detection [4] approach (shown in Figure 4) is introduced for text tracking, where the appearance model for matching and the motion model for tracking are adaptively utilized to link the detections into trajectories. At the first step of this approach, a base text detector is performed in each individual frame. Here,

3. There are two main reasons for using this "successive calculation" way. One reason is that with this simplification, our proposed framework for both tracking based text detection and tracking based text recognition can be formulated as a unified Bayesian formulation. So, convenient algorithms can be designed for both text detection and recognition. The other reason is that this calculation is similar to utilize correlations among core variables given a variety of random variables (e.g., a Markov chain) in complex situations. Hence, effective algorithms can be constructed for text detection and recognition from complex web videos.

4. In object tracking, a tracking-by-detection strategy always first detects the targets in a pre-processing step by background subtraction or using a discriminative classifier, and then estimates the trajectories with these detection results. Recently, tracking-by-detection is also a typical tracking technique for tracking based text detection and recognition [4].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

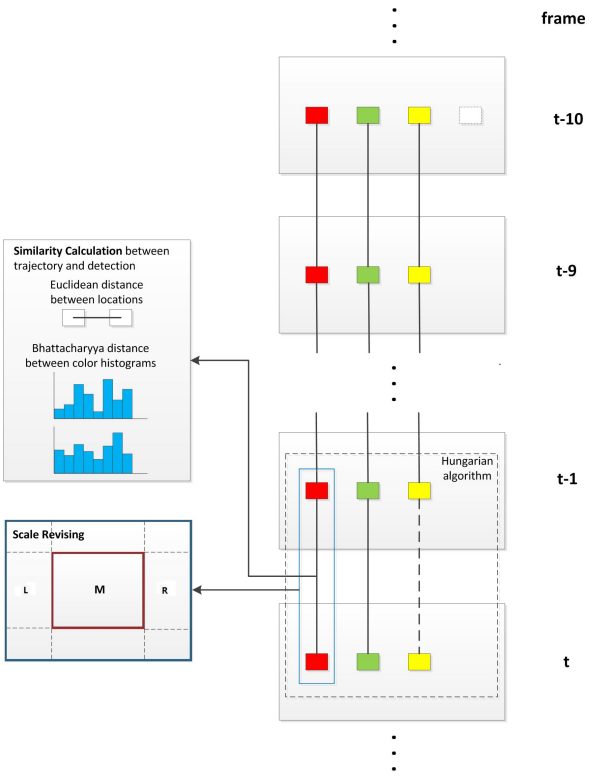


Fig. 4. Overview of text tracking with a tracking-by-detection technique, where the text detections are marked with the rectangles, and the false matching (linking) results are marked with a dashed line.

our previous robust text detector [15] is used. Then, the detection results across consecutive frames are linked into trajectories, where the detections in frame t are sequentially added to the trajectories from frame $t-1$. The similarity between a trajectory and one detection is calculated based on the location information and the color histograms. After the similarities are computed, the Hungarian algorithm [48] is used to match the detections and the trajectories. In Figure 4, the trajectories with the red and green rectangles are matched the detections from frame $t-10$ to frame t. But the trajectory with yellow rectangles is not matched with any detections in frame t. To improve performance, the missing detections are retrieved by comparing the color histograms in the same location in frame t with the one from frame $t-1$ to frame $t-10$. After that, the scales of the detections are also retrieved in frame t by comparing the same text in frame $t-1$ (see "Detection Revising by Tracking" in Section 5.1).

Following the unified framework, one key issue is how to calculate Equation (7), i.e.,

$$\mathcal{T}_t^{*,0} = \arg\max_{\mathcal{T}_t} P(\mathcal{T}_t | D_t^0, \mathcal{T}_{t-1}^*) \tag{12}$$

There are two major aspects for this tracking based text detection method, i.e., similarity calculation between the trajectories and detections, and trajectory initialization and termination.

4.1 Similarity Calculation

In similarity calculation between D_t^0 and $\mathcal{T}_{t-1}^*$, the Hungarian algorithm is used to compute the assignments between the trajectories and detections. The detection (paired with one trajectory) is then decided whether or not to be linked into the trajectory by checking its similarity (low than a given threshold).

Here, the similarity between a trajectory and one detection is decomposed into the appearance similarity and the location similarity (see Figure 4) as follows,

$$S(T_{t-1,i}^*, d_{t,j}^0) = S_a(T_{t-1,i}^*, d_{t,j}^0) \cdot S_l(T_{t-1,i}^*, d_{t,j}^0) \tag{13}$$

where $T_{t-1,i}^*$ is the i^{th} element of $\mathcal{T}_{t-1}^*$, $d_{t,j}^0$ is the j^{th} element of D_t^0, $S_a(\cdot)$ is the appearance similarity between the trajectory and the detection, and $S_l(\cdot)$ is the location similarity by measuring the distance between two text regions.

First, the appearance similarity, $S_a(T_{t-1,i}^*, d_{t,j}^0)$, is computed as

$$S_a(T_{t-1,i}^*, d_{t,j}^0) = \exp\left(\frac{DB(H(d_{t-1,i,\text{final}}^*), H(d_{t,j}^0))^2}{2\sigma_a^2}\right) \tag{14}$$

where σ_a is a parameter (empirically estimated as 0.15), $DB(\cdot)$ means the Bhattacharyya distance [49] (p. 99), $H(\cdot)$ is the RGB color histogram of a text region, and $d_{t-1,i,\text{final}}^*$ is the region of a trajectory $T_{t-1,i}^*$ in the frame which is the last time when the trajectory $T_{t-1,i}^*$ can be visible.

Second, the location similarity, $S_l(T_{t-1,i}^*, d_{t,j}^0)$, is formulated as

$$S_l(T_{t-1,i}^*, d_{t,j}^0) = \exp\left(\frac{DE(l_{t-1,i}^*, l_{t,j}^0)^2}{2\sigma_l^2}\right) \tag{15}$$

where $DE(\cdot)$ means the Euclidean distance, $l_{t-1,i}^*$ is the predicted location of a trajectory $T_{t-1,i}^*$ in frame t, and $l_{t,j}^0$ is the location of detection $d_{t,j}^0$ in frame t, and σ_l is a parameter (empirically estimated as 10 in our system).

How to calculate $l_{t-1,i}^*$ is a crucial step. Here, when the trajectory's temporal length is short (less than 5), the Kalman filter is used to to estimate the text's position. On the contrary, when the text stays in video for a long time, the text's position is estimated as follows. The trajectory with the detection is first smoothed [50]. The average velocity is then calculated in recent several frames. The text's position in frame t is finally predicted by adding the average velocity (calculated across the recent 5 frames) to the text's position in frame $t-1$.

4.2 Trajectory Initialization and Termination

For trajectory initialization, a new trajectory will be generated only if the detections across three consecutive frames are matched. The detection with none or with only one detection matched in the successive frame will be considered as noise. Moreover, to reduce the runtime, an exit mechanism is introduced for the trajectories. In frame t, similarities between all detections and all trajectories are not necessary to be calculated totally. Instead, a trajectory

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

is terminated if it does not match any detection across five consecutive frames. Correspondingly, $\mathcal{T}_{t-1}^{*}$ are only composed by the remaining trajectories after trajectory termination.

5 Tracking Based Text Detection

Based on the above text tracking technique, a tracking based text detection method is also proposed in our system, where two strategies are designed to retrieve the missing detections and to tune the scales of existing detections respectively. This is mainly related to calculation of Equation (9), namely, $P(D_t^*|\mathcal{T}_t^{*,0}, D_t^0, \mathcal{T}_{t-1}^*)$, in the whole system. Here, two main steps of our tracking based text recognition method, i.e., detection revision and trajectory updating, are described in the following.

5.1 Detection Revising by Tracking

One important issue in tracking based text detection is that quite a few detections are considered as noises in trajectory initialization. Consequently, the core step for constructing D_t^* is how to retrieve (recall) the missing detections according to text tracking and tracking trajectories.

Actually, a part of the detection candidates considered as noises are text regions in some cases. These candidates non-matched with the trajectories are partly because that the variations of backgrounds are always complex when the similarity between the detection and the last frame of the trajectory is low. Hence, in our system, the appearance similarities between one detection result considered as noise and the text regions of all trajectories in the last 10 frames are all calculated. If one of these similarities is larger than a given threshold (empirically set as 0.7 in our experiments), the detection is regarded as a correct detected text region and is correspondingly retrieved. Moreover, if a trajectory from frame $t-1$ matches none of the detections in frame t, we will calculate the color and contour similarity between the predicted position and the trajectory's position in previous 10 frames. If they are similar in more than 5 frames, the predicted position (region) will also be decided as a missing detection and be retrieved again.

Another issue is how to finely tune the text scale. At the left bottom of Figure 4, the red rectangle represents the detection in the current frame, and the blue rectangle is the position of the same text from the previous frame. These rectangles are often with a little difference. To improve the recall, we only tune the text scale when the blue rectangle is bigger than the red one. It means that the scale of a text region increases but not decreases after tuning. The scale refining strategy is based on three color histograms: the color histograms of the "L" region in the current frame H_{L_c} and the previous frame H_{L_p}, and the color histogram of the overlapping region of the red rectangle and the blue rectangle (marked as "M" at the left bottom of Figure 4) in the current frame H_{M_c}. H_{M_c} is compared with H_{L_p} and H_{L_c}. If H_{M_c} is similar to at least one of them, "L" region is decided to be a part of the text region in the current frame. "R" region is decided whether or not to be a part of the text region by the same strategy. After the determination of the width of text region, the height of text region is also refined in the same way.

5.2 Trajectory Updating

After detection revising, the step for trajectory updating is fairly simple. The revised detections are used instead of the original detections in $T_t^{*,0}$, and the retrieved detections are updated and added into the corresponding trajectories. Then, Equation (11) is easily calculated, and the process of tracking based text detection is certainly finished.

6 Tracking Based Text Recognition

We also design a tracking based video text recognition technique with multi-frame integration. Text tracking is always not perfect in complex videos. Specifically, ID switch cannot be avoided. To deal with inevitable text ID switch errors for tracking current objects, in our approach, the tracking trajectories are temporally over-segmented to ensure that the detections of each sub-trajectory are corresponding to the same text. An agglomerative hierarchical clustering algorithm is afterwards used to merge over-segmented trajectories and construct suitable trajectories, i.e., these constructed trajectories have a moderate length for multi-frame integration. At last, a voting strategy is conducted to obtain the final recognition results with these suitable trajectories. Here, an open source OCR software (*tesseract-OCR*) is used as a text recognizer to recognize words in individual frames.

The tracking based text recognition system is shown in Figure 5, where each trajectory is handled independently. The text detection and recognition results from frame t to $t+4$ are shown as an example. The string above the text region is the recognition result, and the value below the text image is the confidence from the recognizer. In the example of Figure 5, an ID switch error occurs in frame $t+3$. In our approach, the recognition results with high confidence (0.65 in our system) (shown in red) is treated as the correct ones. Based on these recognition results with high confidence, the trajectory is first correspondingly over-segmented into several sub-trajectories. Sub-trajectories are then clustered by an agglomerative hierarchical clustering algorithm. Thereafter, the sub-trajectories with noise text are identified and the boundaries of these sub-trajectories are refined. Finally, a voting strategy is utilized to recover the incorrect text.

Here, two major issues of this tracking-based text recognition approach are described in the following subsections, namely, trajectory over-segmentation and merging, and multi-frame integration.

6.1 Over-Segmentation and Merging

In tracking based text recognition, the key part is how to generate the trajectories with the recognition results, i.e.,

$$\mathcal{T}_t^{*,0} = \arg\max_{\mathcal{T}_t} P(\mathcal{T}_t|R_t^0, \mathcal{T}_{t-1}^*) \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

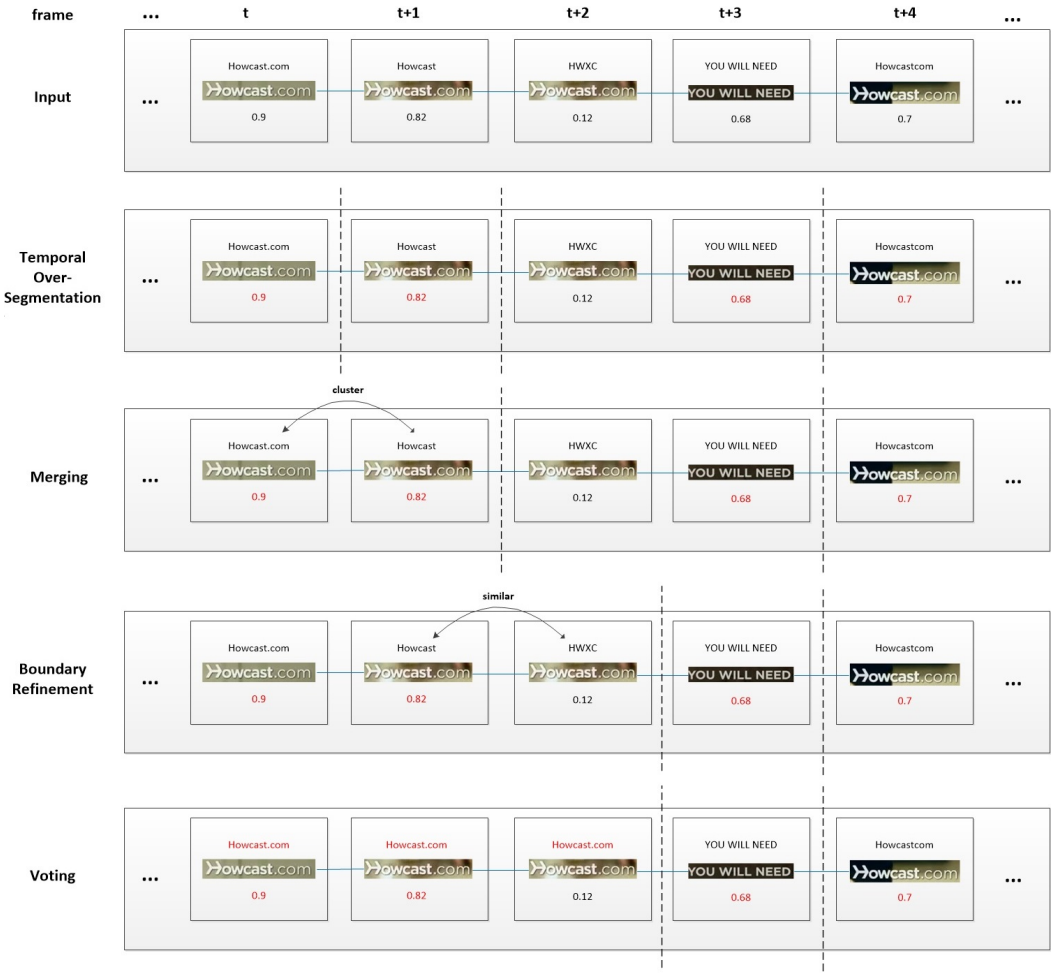


Fig. 5. Flowchart of tracking based text recognition for embedded captions, which include four major steps: over-segmentation, merging, boundary refinement, and voting (multi-frame integration).

The text regions (image patches) are fed into the recognizer in individual frames. The locations of these patches are derived from the detector frame by frame.

First, each trajectory is independently and temporally over-segmented. In this **temporal over-segmentation** step, The recognized text with high confidence is regarded as the reference text, and the frame with the reference text is regarded as a reference frame. The recognition results in other frames are regarded as the noise text.The trajectory is segmented into several short sub-trajectories based on the reference frame. The boundary of two segmentations is determined to be the median frame between two successive reference frames. Correspondingly, the detections on the frames between two successive boundaries compose each sub-trajectory. Obviously, all text from one trajectory is considered as the same one in text tracking. Moreover, the parameters of temporal segmentation are tuned to ensure all real text (ground truth text) of each sub-trajectory is almost the same one. Hence, the segmentation of one trajectory is likely to be over-segmented, which is not perfect but better than under-segmentation where the ID switch is always occurs.

To further improve the performance, similar to some strategies in image segmentation, a cluster algorithm is then utilized to merge the sub-trajectories with a very short length. This **merging** process is performed by an agglomerative hierarchical clustering algorithm. The trajectory's features are derived from the reference text. The dissimilarity between two features is measured with the edit distance. In the process of clustering, the distance between two (interim) clusters is calculated on two representative points (one in each cluster) which are the closest pair points in the two (interim) clusters. The temporal information is not used in the merging step because the same text may not occur continuously since the error of the ID switch is likely to appear in text tracking. After merging, the trajectories of a modest length with high confidence are generated and updated.

Next, refining the segmentation's boundary (**boundary refinement**) is conducted for temporal segmentation of a trajectory. All the recognition results with high confidence have been processed and marked in the previous step. Each noise text identified before is temporally replaced between two consecutive frames of the reference text. The noise text

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

is determined to be as the same as one of the reference text in two frames according to the edit distance between the noise text and each reference text. New trajectories with over-segmentation are correspondingly generated.

6.2 Multi-Frame Integration

The final step of tracking based text recognition is multi-frame integration. In text recognition, some interesting results are observed in our empirical study. First, the text recognizer outputs many noise results with some wrong or incomplete detections in some cases. Second, the text recognizer sometimes classifies words wrongly though the difference between the recognized text and the ground truth is slight. For example, the real text "Howcast.com" can be recognized as ")-)owcast.com", "Howcastcom", "Howcast.c6m", "Howcast.c0m", and so on. We further categorize recognized text into three groups: good text (same to the ground truth), right text (very similar to the ground truth), and noise text (wrongly recognized words with noise results). Consequently, a simple and effective multi-frame recognition integration method is designed in our system. Here, the key strategy is how to compute Equation (9), i.e.,

$$P(R_t^*|R_t^0, \mathcal{T}_{t-1}^*) = P(R_t^*|\mathcal{T}_t^{*,0}, R_t^0, \mathcal{T}_{t-1}^*) \qquad (17)$$

For each sub-trajectory generated as described above, the final recognition results of the sub-trajectory are determined through **voting** on all results from all reference text. If two or more text candidates are the same, the text candidate with the highest confidence wins, i.e., it is regarded as the final recognition result.

7 Experiments

7.1 Video Text Dataset (USTB-VidTEXT)

There are a variety of publicly available datasets for scene text detection and recognition in video, such as ICDAR'13/15 Robust Reading Competition datasets (Video Text Detection and Recogntion), Minetto's dataset [38]. However, there are very few benchmark datasets for web videos with embedded captions. Therefore, we collect five long web video sequences and construct a practical and challenging embedded captions dataset (USTB-VidTEXT) [5], of which all videos are downloaded from YouTube. The ground truths are annotated and available at the sentence level for text in frames.

Compared with the datasets for scene text detection and recognition, the challenges of USTB-VidTEXT are different. Though scene text is with multiple orientations and perspective distortions in many cases, the foreground's (text) color is relatively consistent and clearly contrast to the background's color. There are several specifical challenges of USTB-VidTEXT, e.g., complex backgrounds, varied colors, similar colors and low contrast (see discussions in Section 1.1). Overall, USTB-VidTEXT is challenging for text detection and recognition in video with embedded captions. The brief information of USTB-VidTEXT is shown in Table 1, where NoObj means how many different text objects (sentences in our dataset) appear in the video sequence, and NoGT means how many times all text objects appear. Some samples of screen shots are also shown Figure 6. In our experiments, the experimental systems are constructed and tuned on the *Putin Speech* sequence (training set), and tested on other four videos (testing set).

TABLE 1
The USTB-VidTEXT dataset with 5 typical video sequences directly crawled from the Web.

Seq. Name	No. of Frames	Resolution	NoObj	NoGT
Curtains	2760	480*320	19	3383
Zippers	5250	496*360	41	8520
Wood Box	4556	480*320	33	5661
Biscuit Joiner	5265	480*320	68	9876
Putin Speech	9839	480*360	145	14492
Total	27670	-	306	41932

7.2 Experimental Results

All experiments are performed on a Windows laptop with a 2.6GHz Intel Core i5 CPU (8GB RAM). Experiments with different tracking based video text detection methods are first conducted. Then, the whole end-to-end text recognition system with different recognition strategies is empirically analyzed.

7.2.1 Tracking Based Video Text Detection

In our experiments, the metrics for text detection are from ICDAR 2011 Robust Reading Competition [51], i.e., precision (p), recall (r) and f-score. Our proposed tracking based text detection method is compared with state-of-the-art methods of many kinds. These methods can be categorized into three groups. The first group includes some state-of-the-art text detection methods within single images (individual frames): TexStar [15], Yi' method [14] and Neumann's method [52]. The second one includes two typical video text detection and tracking methods, Nguyen's method [7] and SnooperTrack [38]. The third one includes our tracking based text detection approach (OURS) with the above text detection methods as the base text detectors (TexStar, Yi's method and Neumann's method), i.e., OURS (TexStar), OURS (Yi's) and OURS (Neumann's). Nguyen's method and SnooperTrack are implemented by ourselves, where TexStar is used as the base text detector for fairly comparing.

Experimental results are quantitatively shown in Table 2, where "NoD" means the number of detected text regions for each method in a video sequence. Our proposed method with TexStar (OURS (TexStar)) gets the best precision, recall and f-score among all methods on all video sequences. More importantly, all performances of OURS (TexStar), OURS (Li's) and OURS (Neumann's), i.e. experimental systems of our tracking based text detection method with

5. USTB-VidTEXT will be publicly available in short at http://prir.ustb.edu.cn/T2DAR/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

(a) Curtains (b) Zippers

(c) Wood Box (d) Biscuit Joiner (e) Putin Speech

Fig. 6. Screen shot samples of videos in USTB-VidTEXT.

different base text detectors, are largely higher than the ones of the base text detectors (TexStar, Li's method and Neumann's method) respectively. To some extent, our proposed method is effective and robust to reduce false alarms and to retrieve the missing detections. These experimental results also verify that our text tracking technique is effective and can largely improve the performance of text detection in video. Moreover, our framework is a generic framework for tracking based video text detection. If more effective text detectors are utilized in our unified framework, more promising performances will be obtained.

7.2.2 Tracking Based Video Text Recognition

In experimental evaluations of tracking based video text recognition (the whole end-to-end video text recognition system), detection results (detected regions) and annotated regions (locations) from the ground truths are first matched by a greedy algorithm in each individual frame for deciding whether two text regions (one from detection and the one from annotation) have the same location [6]. The difference between the text recognized and the text annotated is measured with the edit distance [53]. The corresponding evaluation metrics (recall (r), precision (p) and f-score) are formulated as follows,

$$r = \frac{1}{N}\sum_{i}^{N} \text{sim}(\text{TA}_i, \text{TR}_{j_i}), \tag{18}$$

$$p = \frac{1}{M}\sum_{j=1}^{M} \text{sim}(\text{TR}_j, \text{TA}_{i_j}), \tag{19}$$

and

$$f = \frac{2 \times p \times r}{p + r}, \tag{20}$$

where N and M are the total number of the ground truths and the detection results respectively, TR_{j_i} is the corresponding text recognized which has the mostly same location with the i^{th} text annotated (TA_i), and TA_{i_j} is the corresponding text annotated which has the mostly same location with the j^{th} text recognized (TR_j). Here, $\text{sim}(\text{T}_i, \text{T}_j)$ is defined as

$$\text{sim}(\text{T}_i, \text{T}_j) = \begin{cases} 1 - \text{dis}(\text{T}_i, \text{T}_j), & \text{dis}(\text{T}_i, \text{T}_j) < 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

where $\text{dis}(\text{T}_i, \text{T}_j)$ is the edit distance between two (i and j) text strings.

We compare our approach with three typical video text recognition methods, i.e., text recognition in each individual frame (IND), text recognition with frame averaging (AVE), and Phan's method [54]. The base recognizer and all text locations (from detection) are set to the same. Here, a state-of-the-art open source OCR engine, *tesseract-OCR*, is chosen as the base recognizer [7]. Our proposed tracking based text detection method with TexStar, OURS (TexStar) (see Table 2), is selected to detect the text regions (locations) for recognizing.

Experimental results for our proposed tracking based text recognition method (namely, the whole end-to-end video text recognition system, T^{2}DAR) and other comparative methods are shown in Table 3. Several conclusions can be drawn. Firstly, T^{2}DAR gets the best r, p and f on all testing videos. These results verify that tracking can

6. Here, two text regions can be regarded to have the same location if the percent of the overlapping size between them is more than 50%.

7. In the experiments, we use the default setting of *tesseract-OCR* (downloaded from https://github.com/tesseract-ocr) with English.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2692763, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 2
Experimental results for tracking based text detection on USTB-VidTEXT, where "OURS" is our proposed tracking based text detection method within the unified Bayesian-based framework.

Video	Method	r	p	*f*	NoD
Curtains	TexStar	0.5896	0.7882	0.6746	2550
	OURS (TexStar)	**0.9768**	**0.9333**	**0.9546**	3540
	Yi's method	0.1466	0.1107	0.1261	8369
	OURS (Yi's)	0.6896	0.5622	0.6194	9252
	Neumann's method	0.09264	0.1697	0.1199	1050
	OURS (Neumann's)	0.2764	0.4893	0.3533	2637
	Nguyen's method	0.7444	0.8700	0.8023	2890
	SnooperTrack	0.3726	0.5140	0.4320	3360
Zippers	TexStar	0.7907	0.8769	0.8315	7694
	OURS (TexStar)	**0.9663**	**0.9876**	**0.9769**	8336
	Yi's method	0.4288	0.3973	0.4125	15004
	OURS (Yi's)	0.7553	0.7651	0.7601	13336
	Neumann's method	0.0385	0.2480	0.0666	1150
	OURS (Neumann's)	0.0976	0.3656	0.1540	1595
	Nguyen's method	0.8778	0.9142	0.8957	8202
	SnooperTrack	0.2773	0.3574	0.3123	8751
Wood B.	TexStar	0.7808	0.8683	0.8222	5109
	OURS (TexStar)	**0.9766**	**0.9806**	**0.9786**	5651
	Yi's method	0.3313	0.2990	0.3143	10618
	OURS (Yi's)	0.7462	0.7405	0.7433	12200
	Neumann's method	0.0209	0.1940	0.0377	502
	OURS (Neumann's)	0.2459	0.4624	0.3211	1632
	Nguyen's method	0.8351	0.8909	0.8621	5335
	SnooperTrack	0.3052	0.4168	0.3524	5892
Biscuit J.	TexStar	0.9363	0.9397	0.9380	9849
	OURS (TexStar)	**0.9922**	**0.9913**	**0.9918**	9885
	Yi's method	0.4752	0.4102	0.4403	17420
	OURS (Yi's)	0.8077	0.6833	0.7403	22175
	Neumann's method	0.0256	0.2607	0.0466	923
	OURS (Neumann's)	0.2973	0.7586	0.4272	3167
	Nguyen's method	0.9507	0.9573	0.9540	9818
	SnooperTrack	0.1359	0.1613	0.1475	10119

TABLE 3
Experimental results for tracking based text recognition on USTB-VidTEXT, where **IND**, **AVE** and **T^{2}DAR** represent text recognition methods with each individual frame, the average image (frame averaging) and our proposed video text recognition system with the unified framework, respectively.

Sequence name	Method	r	p	f
Curtains	IND	0.4272	0.4082	0.4175
	AVE	0.1031	0.0985	0.1008
	Phan's method	0.1135	0.1084	0.1109
	T^{2}DAR	**0.5531**	**0.5285**	**0.5405**
Zippers	IND	0.5532	0.5654	0.5593
	AVE	0.4655	0.4758	0.4706
	Phan's method	0.2849	0.2912	0.2880
	T^{2}DAR	**0.6025**	**0.6158**	**0.6091**
Wood Box	IND	0.5444	0.5454	0.5449
	AVE	0.0572	0.0573	0.0572
	Phan's method	0.5068	0.5077	0.5073
	T^{2}DAR	**0.5892**	**0.5906**	**0.5899**
Biscuit Joiner	IND	0.6759	0.6753	0.6754
	AVE	0.0550	0.05450	0.0550
	Phan's method	0.5021	0.5017	0.5019
	T^{2}DAR	**0.7007**	**0.7000**	**0.7003**

largely improve the performance of text recognition in video. Secondly, the frame averaging method has the worst performance. One main reason is from text tracking, where the ID switch cannot be avoided in complex situations. When the ID errors occur, the performance of the frame averaging method reduces largely. Finally, Phan's method is based on enhancing the binary image by exploiting temporal information, which is similar to frame averaging but is more trivial and robust. As a consequence, Phan's method has better performance. Summarily, our tracking based text recognition approach obtains a very impressive performance on USTB-VidTEXT.

Note that a variety of parameters are correlated and related to the performances (recall and precision) in our tracking based text detection and recognition approaches. Though there is an overall balance between recall and precision, ROC graphs of them are rather complicated in our experiments. Empirical results show that the threshold of the trajectory's length and the similarity (between a trajectory and one detection) are two main factors (controlling parameters) in our method. In general cases, the threshold of the trajectory's length and the similarity threshold can be set as from 3 to 6, and 0.0001 to 0.001, respectively. In our experimental system, 3 is set for the length threshold for improving recall, and 0.0006 is empirically set for the similarity threshold from the training set.

There are also some failed cases of our proposed approach probably because of text tracking. In text tracking, different text regions are assigned to one ID when the captions have similar locations, similar scales and same backgrounds across consecutive frames with a fairly high probability. In the future, robust features for region matching in text tracking should be further investigated to deal with such related issues.

8 Conclusion

In this paper, we propose a generic Bayesian-based framework of Tracking based Text Detection And Recognition (T^{2}DAR) from web videos for embedded captions. This framework includes three major components, i.e., text tracking, tracking based text detection, and tracking based text recognition. For tracking based text detection, a tracking-by-detection method is used to track the text and a revising mechanism is designed to improve the recall and precision of text detection. For tracking based text recognition, a temporal over-segmentation technique and an agglomerative hierarchical clustering algorithm are employed to alleviate the unavoidable ID switch errors of text tracking, following by multi-frame integration with a voting strategy. A variety of experimental results show that our framework has an impressive performance.

In general, temporal redundancy in video is helpful to improve the performances of text detection and recognition. Text tracking will advance a great deal as a key component for text extraction from complex videos in the future. However, object tracking is an opening issue. In particular, ID switch always occurs inevitably in complex situations. Hence, text tracking in complex videos (e.g., web videos)

is still a challenging topic, and post-processing techniques of text tracking should be further investigated.

ACKNOWLEDGMENTS

The authors are grateful to the Associate Editor and the anonymous reviewers for their constructive comments. The research was partly supported by National Natural Science Foundation of China (61473036).

REFERENCES

[1] 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. 1, 2, 3

[2] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: a survey," *International Journal of Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 84–104, 2005. 1

[3] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, 2015. 1

[4] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2752–2773, 2016. 1, 2, 3, 5, 6

[5] B. Wang, C. Liu, and X. Ding, "A research on video text tracking and recognition," in *Proceedings of SPIE*, 2013, vol. 8664. 1, 4

[6] R. Wang, W. Jin, and L. Wu, "A novel video caption detection approach using multi-frame integration," in *The 17th International Conference on Pattern Recognition (ICPR'04)*, vol. 1, Aug 2004, pp. 449–452. 1, 5

[7] P. X. Nguyen, K. Wang, and S. Belongie, "Video text detection and recognition: Dataset and benchmark," in *IEEE Winter Conference on Applications of Computer Vision (WACV'14)*, March 2014, pp. 776–783. 1, 4, 10

[8] T. Mita and O. Hori, "Improvement of video text recognition by character selection," in *The Sixth International Conference on Document Analysis and Recognition (ICDAR'01)*, 2001, pp. 1089–1093. 1, 5

[9] K. Elagouni, C. Garcia, and P. Sbillot, "A comprehensive neural-based approach for text recognition in videos using natural language processing," in *The ACM International Conference on Multimedia Retrieval (ICMR'11)*, April 2011, p. 23. 1, 3

[10] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in *International Conference on Computer Vision and Pattern Recognition (CVPR'04)*, vol. 2, June 2004, pp. 366–373. 3

[11] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1631–1639, Dec 2003. 3

[12] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *IEEE International Conference on Document Analysis and Recognition*, 2011, pp. 429–434. 3

[13] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'10)*, June 2010, pp. 2963–2970. 3

[14] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2594–2605, Sept 2011. 3, 10

[15] 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, May 2014. 3, 7, 10

[16] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1930–1937, September 2015. 3

[17] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 800–813, March 2011. 3

[18] D. Chen, J.-M. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," *Pattern Recognition*, vol. 37, no. 3, pp. 595–608, 2004. 3

[19] R. Lienhart and W. Effelsberg, "Automatic text segmentation and text recognition for video indexing," *Multimedia Systems*, vol. 8, no. 1, pp. 69–81, 2000. 3

[20] H. Li and D. Doermann, "Automatic text tracking in digital videos," in *IEEE Second Workshop on Multimedia Signal Processing*, 1998, pp. 21–26. 3

[21] J. Xi, X.-S. Hua, X.-R. Chen, L. Wenyin, and H.-J. Zhang, "A video text detection and recognition system," in *IEEE International Conference on Multimedia and Expo (ICME'01)*, Aug 2001, pp. 873–876. 3, 4

[22] 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, Jan 2000. 4

[23] C. Wolf, J. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," in *The 16th International Conference on Pattern Recognition (ICPR'02)*, vol. 2, 2002, pp. 1037–1040. 4

[24] W. Huang, P. Shivakumara, and C. Tan, "Detecting moving text in video using temporal information," in *The 19th International Conference on Pattern Recognition (ICPR'08)*, Dec 2008, pp. 1–4. 4

[25] Y. Na and D. Wen, "An effective video text tracking algorithm based on sift feature and geometric constraint," in *Advances in Multimedia Information Processing - PCM 2010*, 2010, vol. 6297, pp. 392–403. 4

[26] M. Isard and A. Blake, "CONDENSATION–Conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998. 4

[27] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multimodality through mixture tracking," in *The Ninth IEEE International Conference on Computer Vision (ICCV'03)*, vol. 2, Oct 2003, pp. 1110–1116. 4

[28] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805 – 1819, 2005. 4

[29] M. Tanaka and H. Goto, "Text-tracking wearable camera system for visually-impaired people," in *The 19th International Conference on Pattern Recognition (ICPR'08)*, Dec 2008, pp. 1–4. 4

[30] H. Goto and M. Tanaka, "Text-tracking wearable camera system for the blind," in *The 10th International Conference on Document Analysis and Recognition (ICDAR'09)*, July 2009, pp. 141–145. 4

[31] X. Rong, C. Yi, X. Yang, and Y. Tian, "Scene text recognition in multiple frames based on text tracking," in *IEEE International Conference on Multimedia and Expo (ICME'14)*, July 2014, pp. 1–6. 4

[32] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Computer Vision - ECCV 2004*, 2004, vol. 3021, pp. 28–39. 4

[33] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Computer Vision - ECCV 2008*, 2008, vol. 5302, pp. 234–247. 4

[34] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 256 – 268, 2002. 4

[35] H. Shiratori, H. Goto, and H. Kobayashi, "An efficient text capture method for moving robots using dct feature and text tracking," in *The 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, 2006, pp. 1050–1053. 4

[36] M. Tanaka and H. Goto, "Autonomous text capturing robot using improved dct feature and text tracking," in *The Ninth International Conference on Document Analysis and Recognition (ICDAR'07)*, vol. 2, Sept 2007, pp. 1178–1182. 4

[37] C. Mi, Y. Xu, H. Lu, and X. Xue, "A novel video text extraction approach based on multiple frames," in *The Fifth International Conference on Information, Communications and Signal Processing*, 2005, pp. 678–682. 4

[38] R. Minetto, N. Thome, M. Cord, N. Leite, and J. Stolfi, "Snooper-track: Text detection and tracking for outdoor videos," in *The 18th IEEE International Conference on Image Processing (ICIP'11)*, Sept 2011, pp. 505–508. 4, 10

[39] L. Gomez and D. Karatzas, "Mser-based real-time text detection and tracking," in *The 22nd International Conference on Pattern Recognition (ICPR'14)*, Aug 2014, pp. 3110–3115. 4

[40] Z.-Y. Zuo, S. Tian, W.-Y. Pei, and X.-C. Yin, "Multi-strategy tracking based text detection in scene videos," in *The 13th International*

Conference on Document Analysis and Recognition (ICDAR'15), 2015, pp. 66–70. 4

[41] S. Tian, W.-Y. Pei, Z.-Y. Zuo, and X.-C. Yin, "Scene text detection in video by learning locally and globally," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*, 2016, pp. 2647–2653. 4

[42] W. Zhen and W. Zhiqiang, "An efficient video text recognition system," in *International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC'10)*, vol. 1, Aug 2010, pp. 174–177. 4, 5

[43] H. Li and D. Doermann, "Text enhancement in digital video using multiple frame integration," in *The Seventh ACM international conference on Multimedia (ACM MM'99)*, 1999, pp. 19–22. 4

[44] X.-S. Hua, P. Yin, and H.-J. Zhang, "Efficient video text recognition using multiple frame integration," in *International Conference on Image Processing (ICIP'02)*, vol. 2, 2002, pp. 397–400. 4

[45] J.-C. Shim, C. Dorai, and R. Bolle, "Automatic text extraction from video for content-based annotation and retrieval," in *The Fourteenth International Conference on Pattern Recognition (ICPR'98)*, vol. 1, Aug 1998, pp. 618–620. 5

[46] J. Zhou, L. Xu, B. Xiao, R. Dai, and S. Si, "A robust system for text extraction in video," in *International Conference on Machine Vision (ICMV'07)*, Dec 2007, pp. 119–124. 5

[47] R. Lienhart and F. Stuber, "Automatic text recognition in digital videos," 1996, pp. 180–188. 5

[48] H. W. Kuhn, "The hungarian method for the assignment problem," in *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, 2010, pp. 29–47. 7

[49] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2^{nd} ed. Academic Press Professional, Inc., 1990. 7

[50] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002. 7

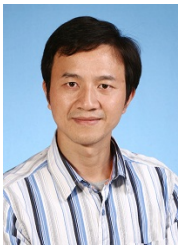
[51] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," *The 12th International Conference on Document Analysis and Recognition (ICDAR'11)*. 10

[52] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*, June 2012, pp. 3538–3545. 10

[53] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006. 11

[54] T. Phan, P. Shivakumara, T. Lu, and C. Tan, "Recognition of video text through temporal integration," in *The 12th International Conference on Document Analysis and Recognition (ICDAR'13)*, Aug 2013, pp. 589–593. 11

Shu Tian received the B.Sc and Ph.D degrees in Computer Science from University of Science and Technology Beijing, China, in 2010 and 2016, respectively. Currently, he is a member of the faculty with the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He has published about 10 research papers (IEEE TIP, Neurocomputing, IJCAI, ICDAR, IJCNN, etc.). His research interests include object tracking, pattern recognition, and multimedia understanding.



Xu-Cheng Yin (M'10, SM'16) received the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences in 2006, and the B.Sc and M.Sc. degrees both in Computer Science from University of Science and Technology Beijing, China, in 1999 and 2002, respectively. He is a Professor in Department of Computer Science and Technology, University of Science and Technology Beijing, China. He was a visiting researcher at the School of Computer Science, University of Massachusetts Amherst, USA, from Jan 2013 to Jan 2014, and from Jul 2014 to Aug 2014. From 2006 to 2008, he was a researcher at IT Lab, Fujitsu R&D Center.

His team won the first place of both "Text Localization in Real Scenes" and "Text Localization in Born-Digital Images" in the ICDAR 2013 Robust Reading Competition, and won the first place of both "Focused (Scene) Text End-to-End Recognition (Generic)" and "Born-Digital Text End-to-End Recognition (Generic)", and also won the first place of "Video Text Localisation" in the ICDAR 2015 Robust Reading Competition. He has published more than 50 research papers (IEEE TPAMI, IEEE TIP, Information Sciences, Information Fusion, PLoS ONE, IJCAI, SIGIR, ICDAR, ICPR, CIKM, ICMR, etc.). His research interests include pattern recognition, computer vision, machine learning, information retrieval, and document analysis and recognition.

Ya Su (M'11) received the B.Sc., M.Sc., and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 2003, 2006, and 2010, respectively.

He was a Postdoctoral Fellow with State Key Laboratory of Intelligent Technology and Systems, Department of Electronic Engineering, Tsinghua University, Beijing, China. He is currently a member of the faculty with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. He has published more than 10 research papers (IEEE TIP, IEEE TSMC, IEEE TCSVT, ICPR, etc.). His research interests include machine learning and computer vision.

Hong-Wei Hao (1967-2016) received the Ph.D. degree in 1997 from the Institute of Automation, Chinese Academy of Sciences, China, where he was a Professor from 2012 to 2016. From 1999 to 2011, he was an Associate Professor and then a Professor at the Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China. From 2002 to 2003, he was a Visiting Researcher with the Central Research Laboratory, Hitachi Ltd., Tokyo, Japan. His research interests included pattern recognition, OCR, large-scale semantic computing theory and technology, and large-scale machine learning theory.