Liu XC, Liu W, Ma HD *et al.* PVSS: A progressive vehicle search system for video surveillance networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(3): 634–644 May 2019. DOI 10.1007/s11390-019-1932-x

PVSS: A Progressive Vehicle Search System for Video Surveillance Networks

Xin-Chen Liu, *Member*, *CCF*, *IEEE*, Wu Liu, *Member*, *CCF*, *IEEE*
Hua-Dong Ma*, *Fellow*, *CCF*, *Senior Member*, *IEEE*, *Member*, *ACM*, and Shuang-Qun Li

Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China

E-mail: {liuxinchen1, liuwu1}@jd.com; mhd@bupt.edu.cn; shuangqunli@hotmail.com

Received October 20, 2018; revised March 22, 2019.

Abstract This paper is focused on the task of searching for a specific vehicle that appears in the surveillance networks. Existing methods usually assume the vehicle images are well cropped from the surveillance videos, and then use visual attributes, like colors and types, or license plate numbers to match the target vehicle in the image set. However, a complete vehicle search system should consider the problems of vehicle detection, representation, indexing, storage, matching, and so on. Besides, it is very difficult for attribute-based search to accurately find the same vehicle due to intra-instance changes in different cameras and the extremely uncertain environment. Moreover, the license plates may be mis-recognized in surveillance scenes due to the low resolution and noise. In this paper, a progressive vehicle search system, named as PVSS, is designed to solve the above problems. PVSS is constituted of three modules: the crawler, the indexer, and the searcher. The vehicle crawler aims to detect and track vehicles in surveillance videos and transfer the captured vehicle images, metadata and contextual information to the server or cloud. Then multi-grained attributes, such as the visual features and license plate fingerprints, are extracted and indexed by the vehicle indexer. At last, a query triplet with an input vehicle image, the time range, and the spatial scope is taken as the input by the vehicle searcher. The target vehicle will be searched in the database by a progressive process. Extensive experiments on the public dataset from a real surveillance network validate the effectiveness of PVSS.

Keywords multi-modal data analysis, progressive search system, vehicle search, video surveillance network

1 Introduction

Physical object search, which aims to find an object sensed by ubiquitous sensor networks like surveillance networks, is one of the most important services provided by the Internet of Things (IoT)[1]. Vehicle, including car, bus, truck, etc., is one type of the most common objects in video surveillance networks. Therefore the vehicle search system has many potential applications in the era of IoT. The search engines of the Internet, e.g., Google, YouTuBe, and Amazon's search engine, can assist us in looking for webpages, images, videos, and products in the information space or cyber space[2], while the task of the vehicle search engine is to find the target vehicle in the physical space[3]. The vehicle search system can provide pervasive applications such as intelligent transportation[4,5] and automatic surveillance[6]. Fig.1 shows an example, in which the user can input a query vehicle, the search area and the time interval, and the system can return the locations and timestamps of the target.

Early vehicle retrieval methods and systems are mainly focused on the attribute-based framework[7−9].

Regular Paper
Recommended by BigMM 2018
A preliminary version of the paper was published in the Proceedings of BigMM 2018.
This work was supported by the Funds for International Cooperation and Exchange of the National Natural Science Foundation of China (NSFC) under Grant No. 61720106007, the NSFC-Guangdong Joint Fund under Grant No. U1501254, the National Key Research and Development Plan of China under Grant No. 2016YFC0801005, the NFSC under Grant No. 61602049, and the 111 Project under Grant No. B18008.

*Corresponding Author

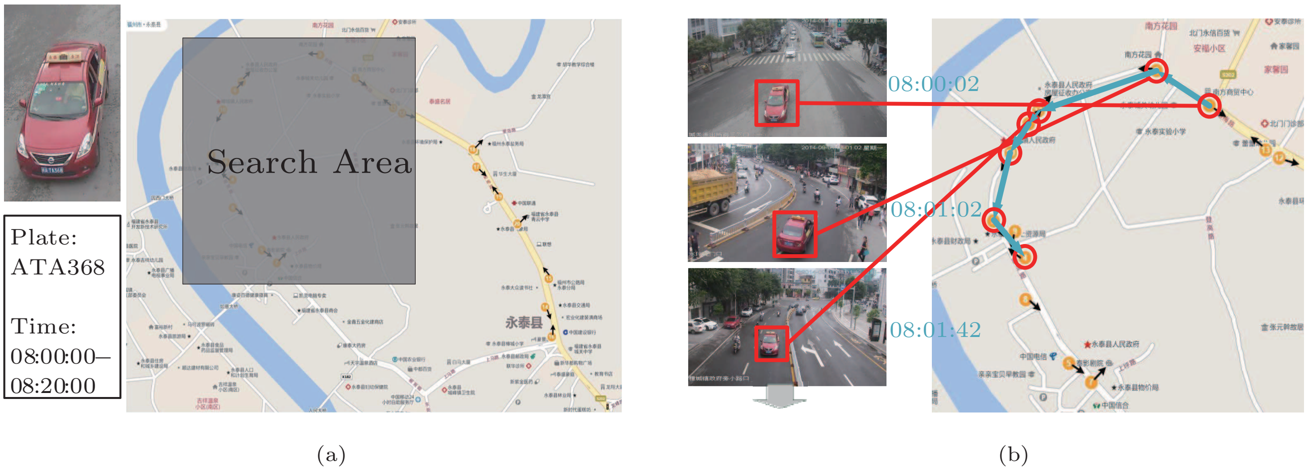


Fig.1. Typical example of vehicle search. Given a specific vehicle, the time interval, and the spatial scope, the system returns where and when the vehicle appeared in the surveillance networks. (a) Input query. (b) Result.

They first classify vehicles by types, models, and colors, and then index and retrieve them with the assigned attributes. Recently, the vehicle search research has been focused on content-based vehicle matching, also known as vehicle re-identification (Re-Id), which uses the content of images to find vehicles in the database[10,11]. Besides, multi-modal contextual information like spatiotemporal information is also explored to assist vehicle Re-Id[12−14]. With the development of representation models, such as hand-crafted descriptors and convolutional neural network (CNN), these methods obtain significant improvement. However, it is difficult to precisely find the specific vehicle only based on attributes because of the intra-instance changes in different cameras and the minor inter-instance differences between similar vehicles. Furthermore, existing vehicle Re-Id approaches assume that the vehicle images have been well cropped and aligned from the video frames. Therefore, they only consider the feature extraction and one-to-N matching for the vehicle images. Nevertheless, a vehicle search engine, as a complex system, must consist of many components like vehicle extraction, representation, indexing, and retrieval. Moreover, both the accuracy and the efficiency should be considered when designing the system.

Towards this end, we design a progressive vehicle search system, named as PVSS, in this paper. PVSS contains three key modules: the crawler of vehicle data, the vehicle indexer based on multi-grained features, and the progressive vehicle searcher. To guarantee high accuracy and efficiency during search, a series of data structures are designed for the vehicle search system. In the crawler, not only visual contents but also contextual information are extracted from the surveillance networks. Then the multimodal data is exploited by deep learning based models to obtain discriminative and robust features of vehicles, which is then organized by the multi-level indexes. In the search process, the vehicle is searched in a progressive manner, including the from-coarse-to-fine search in the feature domain and the from-near-to-distant search in the physical space. At last, extensive experiments on a large-scale vehicle search dataset collected from a real-world surveillance network show the state-of-the-art results of the proposed system.

Compared with our previous conference paper[15], we provide more analysis on contextual information such as the spatiotemporal information in surveillance networks. For example, we discuss the temporal distance between neighboring cameras in the surveillance network by analyzing the travel time of vehicles in our collected data. We also compare the characteristics of vehicles with those of persons which have been studied in related work. Based on the analysis of spatiotemporal information of vehicles in surveillance networks, we propose a new camera neighboring graph compared with the previous model[15]. Particularly, in [15] we only adopted the fixed spatial distance between neighboring cameras as the weights of edges in the graph, which is too simple to model the spatiotemporal cues. In this new manuscript, we also use the temporal distance between neighboring cameras learned from training data to model the spatiotemporal relations, which further improves the performance of the system.

2 Related Work

2.1 Multimedia Retrieval

In the past two decades, content-based multimedia retrieval (CBMR) has been extensively studied[16–21]. CBMR methods usually extract visual features from images or videos and estimate the similarity between the query and the source in the database. For examples, Video Google was proposed by Sivic and Zisserman to achieve object search in videos with the idea of webpage retrieval[16]. Lin *et al.*[22] exploited the 3-D representation models for content-based vehicle search. Farhadi *et al.*[23] proposed to represent the appearance of objects by their attributes for image retrieval. Zheng *et al.*[24] proposed a large-scale image retrieval method with an effective visual model and efficient index structures. Liu *et al.*[20] designed an instant video search system for movies search on mobile devices. However, different from the CBMR task, only depending on visual features, i.e., the appearance of vehicles, cannot give precise results because of the minor inter-class differences between very similar vehicles and varied intra-instance changes in different cameras.

2.2 Person Re-Id and Search

Content-based person Re-Id has been studied for several years[25–28]. The main topics include feature representation of images[29,30] and metric learning for feature embedding[31,32]. For example, Li *et al.*[29] adopted the CNNs to learn discriminative features from large-scale training data and obtained better results than hand-crafted features. Liao *et al.*[30] proposed a local maximal occurrence representation which was robust to environmental noise and achieved the best performance among hand-crafted features. Zhang *et al.*[31] proposed a metric learning method based on null space which can make the samples of the same person close and the samples of different persons distant in the feature space. Hermans *et al.*[32] proposed an end-to-end deep metric learning approach by incorporating deep CNN with a variant of the triplet loss to achieve the state-of-the-art performance on several public datasets. Besides person Re-Id, attributes and context information are also used for person retrieval. For example, Feris *et al.*[33] proposed a system for attribute-based people search in surveillance environments. Xu *et al.*[34] designed an object browse and retrieval system, which integrates vision features and spatial-temporal cues by a graph model for the retrieval of pedestrians and cyclists.

However, compared with the person or pedestrian, the vehicle has its unique properties which make vehicle search different from person Re-Id. First of all, vehicle is a type of rigid object, which shows extreme intra-class difference due to the varied viewpoints. Moreover, as a mature industrial product, vehicles of the same model have very similar appearance. Nonetheless, each vehicle has a unique mark, the license plate, to identify an instance of vehicle. Therefore, we can first find the appearance-similar vehicles by visual feature representation and metric learning as in person Re-Id. Furthermore, the license plate can be explored to uniquely recognize a vehicle across different cameras.

2.3 Vehicle Re-Id and Search

In recent years, vehicle search has been mainly focused on content-based vehicle Re-Id, which aims to find the target vehicle from the database with a query vehicle image[10,11]. For example, Liu *et al.*[11] proposed a deep CNN-based method, named Deep Relative Distance Learning, to jointly learn visual features and metric mapping for vehicle Re-Id. Besides appearance features, the contextual information such as license plates and spatiotemporal records is also used for vehicle Re-Id. For example, Liu *et al.*[12] proposed a progressive vehicle search method which exploits image features, license plates, and contextual information in a progressive manner. Wang *et al.*[14] proposed a framework to learn local landmarks and global features of vehicles and refine the results with a spatiotemporal regularization model. Similar to person Re-Id, existing vehicle Re-Id methods also assume that the vehicle images have been detected and well aligned from video frames. Therefore, they only consider the feature representation and similarity metrics for image matching. However, to build a complete search system, we consider not only the problems for content-based vehicle Re-Id but also the tasks of data acquisition, organization, and retrieval.

3 Overview

Fig.2 illustrates the overall architecture of the PVSS system. It contains three modules.

- The offline vehicle crawler receives the video streams from surveillance cameras and crops vehicle image sequences from video frames.

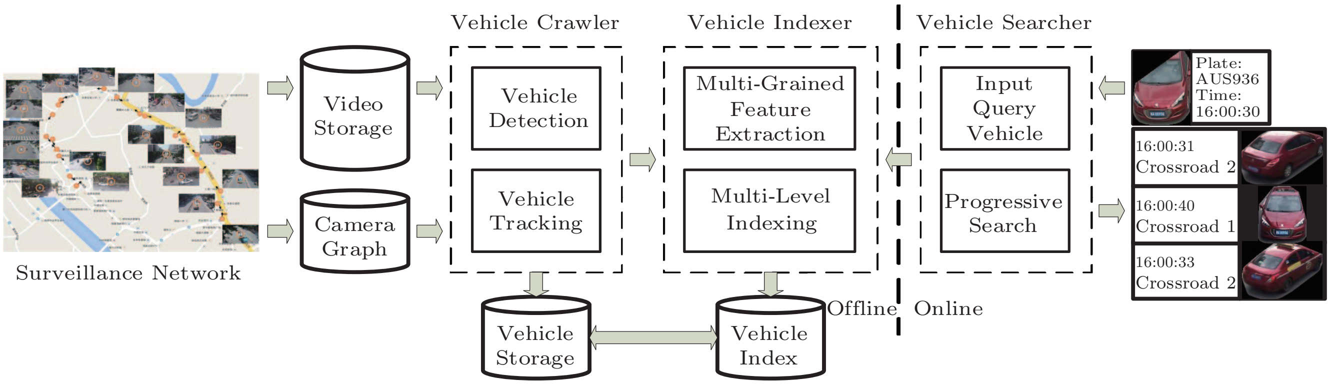


Fig.2. Architecture of the progressive vehicle search system PVSS.

- The vehicle indexer extracts multi-grained visual features from vehicle tracks and constructs the multi-level indexes for efficient search.
- The online vehicle searcher performs the progressive search process with the multi-level indexes in both the feature domain and the spatiotemporal space.

Before introducing the details of each component, we first present the main data structures of PVSS in Section 4.

4 Data Structures

The data that we can utilize is diverse and in multiple modalities. Various semantic contents like vehicle plates, types, colors, and visual features can be extracted in online or offline manner as in [12, 35]. The data modalities include text, digits, coordinates, structures, and so on. The topology and spatiotemporal context of surveillance networks can be more complex data structures such as graphs. Therefore, these data should be described in proper structures, which are effective for retrieval and flexible for extension. In this section, we first introduce the vehicle track metadata, which is to describe the image sequences of vehicles captured by surveillance cameras. Then, the camera table is designed to index the vehicle track metadata for each camera. At last, we build a camera neighboring graph to represent the spatial topology of the surveillance networks.

4.1 Vehicle Track Metadata

According to the variety of video contents and extraction approach, the vehicle track metadata is proposed to describe vehicle image sequences which are obtained from cameras. Table 1 lists the attributes and descriptions of the metadata in detail. In our system, the vehicle tracks are extracted by the vehicle crawler frame by frame, which will be presented in Subsection 5.1. The object tracking method is used to group the images of the same vehicle in neighbor frames as an instance of vehicle track. As in Table 1, the unique camera ID and vehicle ID specify a unique vehicle. Among these attributes, the visual features are the most important information to represent the multi-grained visual representation of each vehicle, and are utilized in the indexing and search procedures. The extraction of visual features will be given in Subsection 5.2.

Table 1. Vehicle Track Metadata

Name	Type	Description
Camera ID	int	Unique ID of the camera that captures the track
Vehicle ID	long	Unique ID of the vehicle track
Frame ID	long	ID of the first frame in the vehicle track
Track length	int	Frame count of the vehicle track
Trajectory	point[]	Point sequence of the vehicle track
Visual features	float[]	Multi-grained visual features extracted from the vehicle track
Duration	float	Time duration of the vehicle track
Plate	string	License plate string of the object (if recognized)

4.2 Camera Table

After the generation of vehicle track metadata, the storing and indexing of these data should be considered. In our system, the camera table is designed to index instances of vehicle track metadata for each camera.

For each camera, we allocate a camera table to index the vehicle track metadata extracted from this camera. The videos are processed by the order of time; therefore the metadata instances are also generated by the order of time and appended to the tails of camera tables.

This keeps the entries of camera tables in ascending order. Fig.3 shows the structure of the camera table. In the real implementation, the camera table can be implemented by relational databases like MySQL or distributed databases like HBase in the data center. When the scale of camera tables grows up, the tables will be organized in a tree-like structure for efficient index and search.

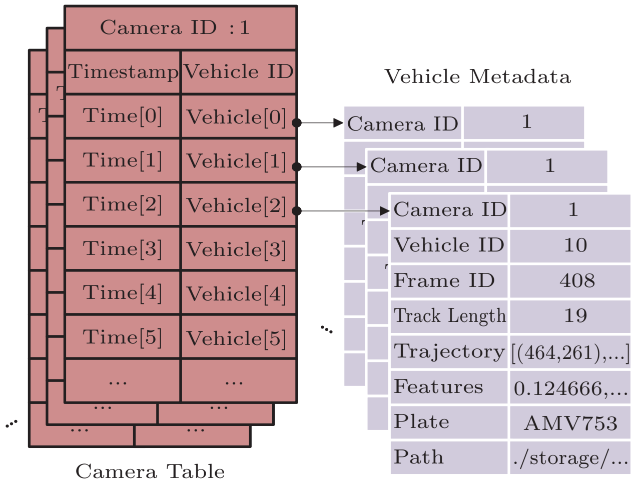


Fig.3. Structure of the camera table.

4.3 Camera Neighboring Graph

4.3.1 Topology Construction

The camera neighboring graph records the geo-locations of cameras and the topology of the surveillance networks, which is obtained from the infrastructure companies and the map services.

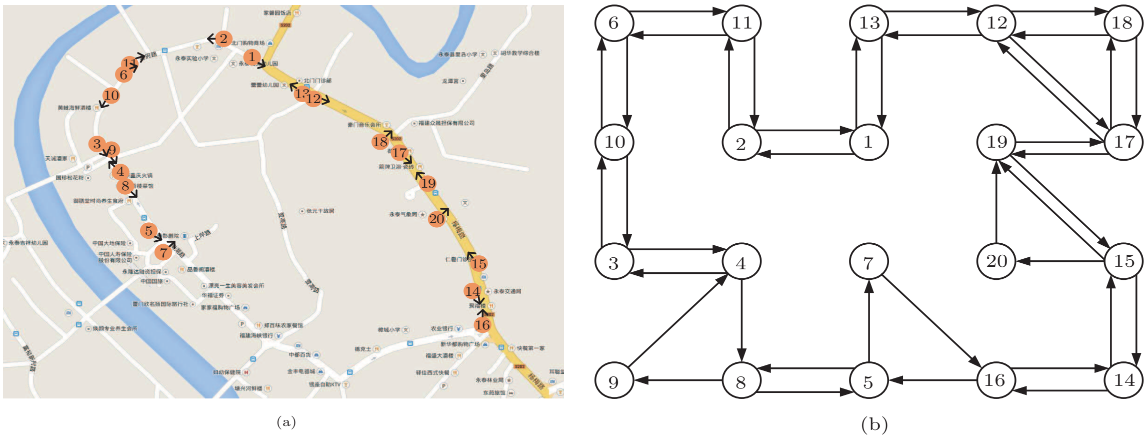
We define the graph as a directed graph $G = (N, E, W)$. The graph is composed by the node set $N = \{n_1, ..., n_C\}$, the edge set $E = \{e_{i,j}\}$, and the weight set $W = \{w_{i,j}\}$. Fig.4 illustrates an example of the camera neighboring graph which is built from a subset of a real-world surveillance network. The nodes represent the set of cameras, which consist of the GPS coordinates and settings of cameras. The edges constitute the set of directed connections between neighboring cameras. The edges are determined not only by the topology of the city roads but also by the heading directions and fields of view (FOV) of cameras. Thus we define the view-connected edge as below.

Definition 1 (View-Connected Edge). *A view-connected edge $e_{i,j} = (n_i, n_j)$ connects a pair of cameras in N, and if a vehicle can reappear in the FOV of camera j directly after appearing in the FOV of camera i, then there is a view-connected edge $e_{i,j}$ from n_i to n_j.*

4.3.2 Weight Modeling

The weight set W of G contains two parts. The first part is W_t. It stores the spatial distances of neighboring cameras, which can be obtained from map services like Google Map. The second part is W_s which contains the temporal distances between neighboring cameras learned from the training data. Here we will give details about the learning of W_s.

Several studies have proposed models to estimate the travel time in surveillance networks. The authors of [36] proposed a graph-based vehicle search model. According to this model, the weight of an edge is modeled by the mean time cost of all vehicles that traveled the edge during the search time. When given a search time interval, the history records in the time interval are used to compute the mean time cost in this time interval. Xu *et al.*[34] proposed a graph model for re-



(a)

(b)

Fig.4. Example of the camera neighboring graph. (a) Camera locations and the city map of a real-world surveillance network. (b) Graph abstracted from the network.

lated object search in a campus. This model estimates the time delay between cameras using object reappearance. It is assumed that the speed of an object changes slightly; therefore the time delay is negatively linearly correlative to the travel speed. Using the labeled data collected from the surveillance network, a linear model of time delay and optical flow is learned with a standard regression method.

However, according to the statistics on our dataset as shown in Fig.5, the above two models cannot be directly applied to our scenario. We select five sequential edges in the surveillance network and plot the records in about one hour from 15:59:58 to 16:59:58. Fig.5(a) shows the time cost versus object speed plots. We can find that the time costs are not linearly correlated with the speed of objects. Because we can only obtain the speed in individual cameras but cannot know the speed between cameras, the behaviors of vehicles between cameras are unpredictable with only surveillance videos. The traffic lights, pedestrians, and traffic jams make the actual model very complex. Thus the linear model of time cost and speed would fail in our scenario.

Fig.5(b) illustrates the time cost versus record time plots. From the observation on this part, we find that in different time intervals the travel time of different vehicles changes slightly. In this case, we use a slot-mean model to build the weights. We segment the whole time line into time slots with the fixed length. Supposing that set $C = \{c_{k,l}\}$ contains the time cost records on edge $e_{i,j}$ that fall in the time slot k. We have the mean time cost $m_{i,j,k}$:

$$m_{i,j,k} = \frac{\sum_{l=1}^{|C|} c_{k,l}}{|C|}.$$

In each time slot k, $m_{i,j,k}$ is used as a parameter of the weight function. In addition, we use $\tau_{i,j,k}$ as the other parameter of the weight which is computed as follows:

$$\tau_{i,j,k} = \sqrt{\frac{\sum_{l=1}^{|C|} (c_{k,l} - m_{i,j,k})^2}{|C|}}.$$

After computing $(m_{i,j,k}, \tau_{i,j,k})$ on all time slots, we have a step function for the weight vector $\boldsymbol{w}_{i,j}(x) = (m_{i,j}(x), \tau_{i,j}(x))^{\mathrm{T}}$ on the edge:

$$\boldsymbol{w}_{i,j}(x) = (m_{i,j}(x), \tau_{i,j}(x)) = \sum_{k=1}^{t} \chi_{i,j,k}(x)(m_{i,j,k}, \tau_{i,j,k}),$$

where

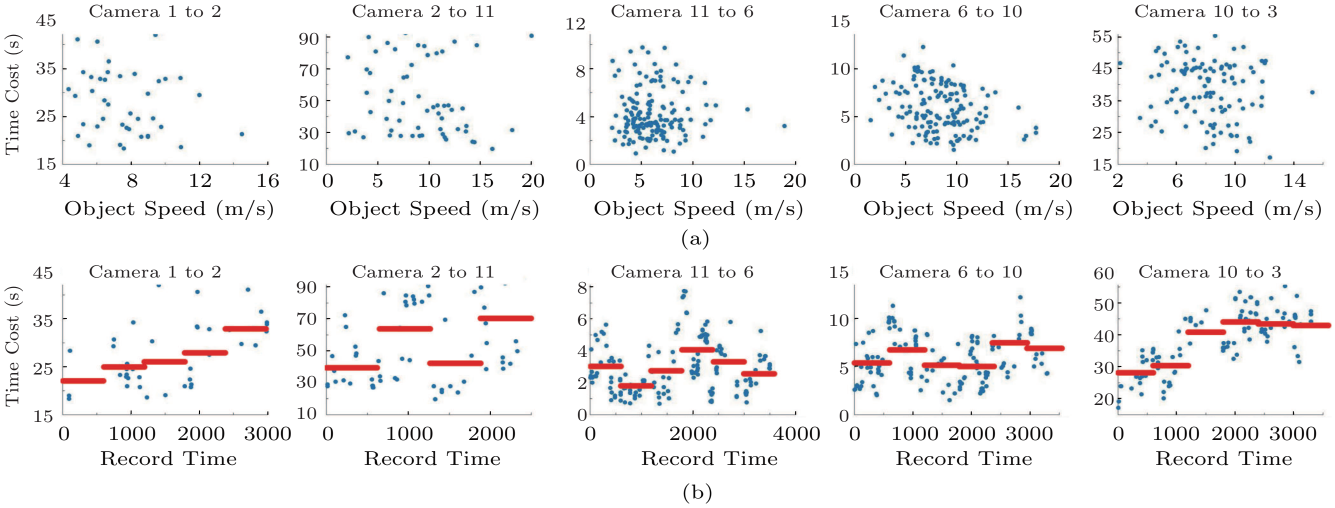
$$\chi_{i,j,k}(x) = \begin{cases} 1, & \text{if } x \text{ is in time slot } k, \\ 0, & \text{otherwise,} \end{cases}$$

where x is an object metadata instance in the start camera i of edge $e_{i,j}$, and t is the total number of time slots. All weight functions on the edges constitute the temporal weight set W_t of graph G.

5 Functional Modules

5.1 Vehicle Crawler

The vehicle crawler aims to detect and crop vehicle images from video frames streamed by the surveillance



(a)

(b)

Fig.5. Scatter plot of time difference statistics. (a) Time cost vs object speed plots. (b) Time cost vs record time plots. The red lines in (b) are the mean time cost in each 600-second time slot.

network. It plays a similar role to the conventional web crawler of the Internet search engines, which crawls and downloads webpages from the World Wide Web.

To effectively locate the vehicles in the video frames, we adopt the state-of-the-art deep learning based object detection model, i.e., Faster R-CNN[37]. Faster R-CNN contains two CNN-based parts. The first is the region proposal network, which is a fully convolutional network to generate object proposals from the input frames. The second is a fully connected network to regress the bounding boxes of objects and the corresponding categories. To achieve precise vehicle detection, we adopt a ResNet-50[38] based Faster R-CNN structure which is pretrained on the ImageNet dataset[39]. Then, the network is finetuned on large-scale vehicle bounding boxes from surveillance videos annotated by ourselves. After detection, a nearest neighbor tracking algorithm is adopted to associate vehicle bounding boxes of the same vehicle between neighbor frames. In our implementation, the Faster R-CNN is deployed on the GPU servers to achieve efficient vehicle detection.

For each track, it is assigned a unique vehicle ID under the corresponding camera. The first frame of the track, the track length, and the sequence of pixel coordinates are recorded into the metadata, while the track that is shorter than 5 will be discarded. After that, we use the off-the-shelf plate recognition tool to extract the plate numbers with a confidence measure. If the tool cannot recognize the plate or return a very low confidence, the plate will be assigned as UNAVAL which means unavailable. At last, the vehicle track metadata is appended to the camera table, meanwhile the image sequence of the track is stored on the vehicle storage server.

5.2 Vehicle Indexer

The vehicle indexer contains two functions: the first is multi-grained visual feature extraction, and the second is multi-level index construction.

For the vehicle tracks, we extract the appearance-based coarse representation and the license plate based fine-grained feature. To learn discriminative and robust feature of vehicle appearance, we adopt the ResNet-50[38] pretrained on ImageNet[39] as the basic network. The network is finetuned on the VeRi dataset[10] with a multi task loss function, which contains a cross entropy loss and a contrastive loss[40]. To learn effective plate feature, a ResNet-18 based siamese neural network for plate verification is trained on massive license plate pairs as in [12]. The above two feature extractors are deployed on the GPU servers for efficiency. In the implementation, we use the 2048-D "pool5" layer of ResNet-50 and the 1024-D "conv3" layer of ResNet-18 as the appearance feature and the plate feature, respectively. For the images in the track, the features are extracted separately and fused by average pooling, which means that each vehicle track has a 2048-D coarse-grained feature and a 1024-D fine-grained feature.

After feature extraction, we build a two-level index for vehicle tracks with the state-of-the-art approximate nearest neighbor index algorithm, i.e., FLANN[41], due to its high efficiency. The level-1 index is built on the appearance feature vectors, while the level-2 is built on the plate feature vectors.

5.3 Vehicle Searcher

In this subsection, we discuss the main procedures of online vehicle search. Given a vehicle image cropped by a user and a time interval, a list of candidate target vehicles and their states will be returned, as shown in Fig.1. As mentioned above, the progressive search contains two aspects: from-coarse-to-fine search with multi-grained features and from-near-to-distant search with the spatiotemporal context.

5.3.1 *From-Coarse-to-Fine Feature Matching*

Vehicle search is generally a one-to-N feature matching problem, in which the similarity between the query and the gallery is estimated and ranked to find the most similar target vehicle to the query. During searching, the query image or track is fed into the feature extraction module to extract its visual feature and plate feature as in Subsection 5.2. Then the visual feature of query is searched with the level-1 index to obtain the coarse similarity, S_c, between the query vehicle and the gallery vehicle. Similarly, the fine similarity, S_f, is obtained with the level-2 index using the plate feature. With the above two similarity scores, the visual similarity between the query vehicle, V_q, and one gallery vehicle, V_g, is:

$$S_v = \lambda \times S_c + (1 - \lambda) \times S_f,$$

where λ is a hyper-parameter to balance the two scores.

In addition to the visual similarity, we also explore the spatiotemporal similarity between the query and the gallery. Given the metadata of V_q and V_g, we can obtain their spatial distance, D_s, and temporal distance, D_t, as

$$D_s = |L(c(V_q)) - L(c(V_g))|,$$
$$D_t = |T(V_q) - T(V_g)|,$$

where $c(\cdot)$ is the operation to get the camera ID of a vehicle, $L(\cdot)$ is the location of a camera, and $T(\cdot)$ is the timestamp of a vehicle. Then, we adopt a two-layer fully connected neural network, i.e., the multi-layer perceptron (MLP), $F(\cdot)$, to model the spatiotemporal similarity of V_q and V_g. The input and the output dimensions of the two fully connected layers are (2, 64) and (64, 1), respectively. The activation functions of the two layers are ReLU and Sigmoid, respectively. The spatiotemporal similarity, S_{st}, is denoted as

$$S_{st} = F([D_s, D_t]),$$

where $[\cdot, \cdot]$ is the concatenation of two elements.

At last, to effectively integrate the visual similarity, S_v, and the spatiotemporal similarity S_{st}, we exploit a fully connected layer with sigmoid activation, $G(\cdot)$, to learn the suitable fusion parameter. Therefore, the final similarity can be computed by

$$S = G([S_v, S_{st}]).$$

The neural networks $F(\cdot)$ and $G(\cdot)$ are trained with the binary cross entropy loss, which can guide the model to determine whether the query and one gallery are the same vehicle or not. During searching, the results are ranked by the similarity scores $\{S_{q,g}\}$ between the query and the set of gallery vehicles.

5.3.2 From-Near-to-Distant Search

To achieve efficient vehicle search, we utilize the camera neighboring graph, G, to achieve the from-near-to-distant search. Given the camera ID of the query, we traverse G in the breadth-first manner. It means that the query vehicle is matched first to the vehicles in the nearest neighboring cameras and then to the distant ones. After each traverse of current neighboring cameras, a list of candidate results is returned. The results will update with the traverse of G but the length of the list remains constant, which guarantees that the most similar results can be shown to users.

6 Experiments

6.1 Dataset

In this paper, we compare the proposed PVSS with different vehicle search methods on the VeRi dataset[42]. The VeRi dataset is collected from 20 surveillance cameras in a real-world surveillance network, which contains about 50 000 images and 9 000 tracks of 776 vehicles. Each vehicle in the VeRi dataset is labeled with various attributes, such as 10 types of colors and nine categories. Moreover, the license plate numbers of vehicles are annotated for more precise vehicle search. Furthermore, the contextual information, such as the spatiotemporal information and the topology of the surveillance network, and distances are annotated. Therefore, it is suitable to evaluate the proposed progressive vehicle search system.

6.2 Experimental Settings

As the similar settings in [42], cross-camera matching is performed, which means that one vehicle image from one camera is used as the query to search for images of the same vehicle captured by other cameras. Vehicle matching is in a track-to-track manner, which means units of the query set and the gallery are both tracks of vehicles cropped from surveillance videos. In our experiments, we use 1 678 query tracks and 2 021 testing tracks as in [42].

To evaluate the accuracy of the methods, HIT@1 (precision at rank 1) and HIT@5 (precision at rank 5) are adopted. In addition, since the query has more than one ground truth, the precision and the recall should be considered in our experiments. Hence, we also use mean average precision to evaluate the comprehensive performance as in [42]. The average precision (AP) is computed for each query as

$$AP = \frac{\sum_{k=1}^{n} P(k) \times gt(k)}{N_{gt}},$$

where n and N_{gt} are the numbers of tests and ground truths respectively, $P(k)$ is the precision at the k-th position of the results, and $gt(k)$ is an indicator function that equals 1 if the k-th result is correctly matched and 0 otherwise. Over all queries, the mean average precision (mAP) is formulated as

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q},$$

in which Q is the number of queries.

6.3 Comparison with Vehicle Re-Id Methods

In this subsection, we first compare the appearance-based search component in PVSS with five appearance-based vehicle Re-Id methods. Among them, methods

1 and 2 are two vehicle Re-Id methods, while methods 3 and 4 are two state-of-the-art approaches for video-based person Re-Id. Then we compare the complete progressive vehicle search system with three state-of-the-art multi-modal data-based approaches. Vehicle matching is achieved by computing the Euclidean distance between a pair of vehicles. The details of all methods are as follows.

1) *Fusion of Color and Attribute* (*FACT*)[10]. This method is the baseline method on the VeRi dataset, which integrates hand-crafted features, e.g., SIFT and Color Name, with attributes extracted by GoogleNet.

2) *Progressive Vehicle Search* (*Progressive*)[12]. This is a progressive vehicle search framework, which uses appearance features and plate verification for vehicle matching and refines the results with spatiotemporal information.

3) *Identity Feature with LSTM* (*ResNet + LSTM*). This approach adopts CNN+LSTM which is the state-of-the-art method for video-based person Re-Id[43]. It can model dynamic patterns of persons like actions and gaits for person Re-Id.

4) *Top-Push Distance Learning* (*TDL*)[44]. This method is one of the state-of-the-art metric learning methods for video-based person Re-Id. We use the visual features extracted by ResNet as basic features. Then the TDL method is used to aggregate and map the original features into the latent space. Finally, vehicles are matched by the Euclidean distance of features.

5) *Appearance-Based Search in PVSS* (*PVSS-App*). This is a part of PVSS, which uses only appearance features learned by CNNs trained by a combination of the cross-entropy loss and the contrastive loss.

6) *Orientation Invariant Feature Embedding and Spatial Temporal Regularization* (*OIFE + STR*)[14]. This method proposes an Orientation Invariant Feature Embedding model to learn 20 landmarks and extracts both local and global features from vehicle images.

7) *Siamese-CNN and Path-LSTM* (*SC + P-LSTM*)[13]. This approach exploits two ResNets[38] in a siamese structure to learn visual features and a one-layer LSTM to model the spatiotemporal context.

8) *PROgressive Vehicle Re-ID* (*PROVID*[42]). This progressive vehicle search framework searches for vehicles in a three-step way: appearance-based coarse filtering, license plate-based fine search, and spatiotemporal re-ranking.

9) *PVSS-App-Plate.* This is a part of the proposed PVSS, which uses the appearance and plate features for vehicle search.

10) *PVSS.* This is the complete progressive vehicle search system proposed in our paper, which performs vehicle search in a from-coarse-to-fine and from-near-to-distant fashion.

Table 2 lists the mAP, HIT@1, and HIT@5 of different approaches. For appearance-only methods, we can find that the traditional methods, i.e., FACT and Progressive, are worse than deep learning based methods. This is because the hand-crafted features cannot effectively model the appearance of a vehicle and comprehensively represent the vehicles. By comparing LSTM-based methods with other deep learning based models, we can see that LSTM-based methods obtain worse results. Although LSTM can model dynamic representation from action or gait for video-based person Re-Id, it cannot be directly utilized for video-based vehicle Re-Id. It is noteworthy that the TDL method for person Re-ID is better than the two baseline appearance-only methods for vehicle Re-ID, i.e., FACT and Progressive, which shows the effectiveness of the hand-crafted features in TDL. But our appearance-based part in PVSS-App achieves the best results because it adopts a multi-task CNN which is optimized by a combination of classification loss and the contrastive loss for metric learning. By this means, the deep CNN can not only learn robust visual features but also map the features into a discriminative metric space in which samples from the same vehicle become close and samples of different vehicles are apart from each other.

Table 2. Results of Vehicle Re-Id Methods on VIVID

Method	mAP	HIT@1	HIT@5
FACT[10]	18.00	52.44	72.29
Progressive[12]	25.11	61.26	75.98
ResNet + LSTM[43]	28.11	56.20	79.14
TDL[44]	35.65	69.61	88.02
PVSS-App	51.00	85.64	95.35
OIFE + STR[14]	51.42	68.30	89.07
SC + Path-LSTM[13]	58.27	83.49	90.04
PROVID[42]	53.42	81.56	95.11
PVSS-App-Plate	61.12	89.69	96.31
PVSS	**62.62**	**90.58**	**97.14**

Note: The best results are in bold.

For multi-modal methods, OIFE + STR and SC + Path-LSTM obtain worse results than the proposed PVSS-App-Plate, because these two methods neglect license plates to uniquely identify vehicles. Moreover, by incorporating spatiotemporal context, PVSS outperforms other multi-modal search methods and achieves the best results.

7 Conclusions

This paper proposed PVSS, a progressive vehicle search system, which can crawl and index vehicles captured by large-scale surveillance networks and provide vehicle search services for users. For the vehicle crawler, the vehicle detection and tracking algorithms are adopted to crop vehicle images from surveillance videos. Then, vehicle images are fed into the vehicle indexer to extract multi-grained visual features, which are utilized to build a multi-level index for vehicle search. In the online search stage, the target vehicle is searched in a from-coarse-to-fine manner with the multi-level index and in a from-near-to-distant way based on the spatiotemporal context of the surveillance network. Extensive evaluations on the VeRi dataset showed the excellent performance of PVSS.

References

[1] Ma H D. Internet of Things: Objectives and scientific challenges. *Journal of Computer Science and Technology* (*JCST*), 2011, 26(9): 919-924.

[2] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 1998, 30(1/2/3/4/5/6/7): 107-117.

[3] Ma H D, Liu W. A progressive search paradigm for the Internet of things. *IEEE MultiMedia*, 2018, 25(1): 76-86.

[4] Zhang J P, Wang F Y, Wang K F *et al.* Data-driven intelligent transportation systems: A survey. *IEEE Trans. Intelligent Transportation Systems*, 2011, 12(4): 1624-1639.

[5] Yuan J Z, Chen H, Zhao B, Xu Y Y. Estimation of vehicle pose and position with monocular camera at urban road intersections. *JCST*, 2017, 32(6): 1150-1161.

[6] Valera M, Velastin S A. Intelligent distributed surveillance systems: A review. *IEE Proceedings-Vision, Image and Signal Processing*, 2005, 152(2): 192-204.

[7] Feris R, Siddiquie B, Zhai Y *et al.* Attribute-based vehicle search in crowded surveillance videos. In *Proc. the 1st Int. Conf. Multimedia Retrieval*, April 2011, Article No. 18.

[8] Feris R S, Siddiquie B, Petterson J *et al.* Large-scale vehicle detection, indexing, and search in urban surveillance videos. *IEEE Transactions on Multimedia*, 2012, 14(1): 28-42.

[9] Wang Q, Ding Y D. A novel fine-grained method for vehicle type recognition based on the locally enhanced PCANet neural network. *JCST*, 2018, 33(2): 335-350.

[10] Liu X C, Liu W, Ma H S, Fu H Y. Large-scale vehicle re-identification in urban surveillance videos. In *Proc. the 2016 IEEE International Conference on Multimedia and Expo*, July 2016, Article No. 145.

[11] Liu H, Tian Y, Yang Y, Pang L, Huang T. Deep relative distance learning: Tell the difference between similar vehicles. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), June 2016, pp.2167-2175.

[12] Liu X C, Liu W, Mei T, Ma H D. A deep learning based approach to progressive vehicle re-identification for urban surveillance. In *Proc. the 14th European Conference on Computer Vision*, October 2016, pp.869-884.

[13] Shen Y, Xiao T, Li H, Yi S, Wang X. Learning deep neural networks for vehicle re-ID with visual-spatio-temporal path proposals. In *Proc. the 2017 IEEE International Conference on Computer Vision* (*ICCV*), October 2017, pp.1918-1927.

[14] Wang Z, Tang L, Liu X *et al.* Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *Proc. the 2017 ICCV*, October 2017, pp.379-387.

[15] Liu X C, Liu W, Ma H D, Li S Q. A progressive vehicle search system for video surveillance networks. In *Proc. the 4th IEEE Int. Conf. Multimedia Big Data*, September 2018, Article No. 40.

[16] Sivic J, Zisserman A. Video Google: A text retrieval approach to object matching in videos. In *Proc. the 9th ICCV*, October 2003, pp.1470-1477.

[17] Lew M S, Sebe N, Djeraba C, Jain R. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2006, 2(1): 1-19.

[18] Hu W, Xie N, Li L *et al.* A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 2011, 41(6): 797-819.

[19] Mei T, Rui Y, Li S, Tian Q. Multimedia search reranking: A literature survey. *ACM Computing Surveys,* 2014, 46(3): Article No. 38.

[20] Liu W, Mei T, Zhang Y D. Instant mobile video search with layered audio-video indexing and progressive transmission. *IEEE Transactions on Multimedia*, 2014, 16(8): 2242-2255.

[21] Zheng L, Yang Y, Tian Q. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 40(5): 1224-1244.

[22] Lin Y L, Tsai M K, Hsu W H, Chen C W. Investigating 3-D model and part information for improving content-based vehicle retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013, 23(3): 401-413.

[23] Farhadi A, Endres I, Hoiem D *et al.* Describing objects by their attributes. In *Proc. the 2009 CVPR*, June 2009, pp.1778-1785.

[24] Zheng L, Wang S, Liu Z, Tian Q. Fast image retrieval: Query pruning and early termination. *IEEE Transactions on Multimedia*, 2015, 17(5): 648-659.

[25] Gong S, Cristani M, Yan S C, Loy C C. Person Re-Identification. Springer-Verlag London, 2014.

[26] Zheng L, Shen L, Tian L *et al.* Scalable person re-identification: A benchmark. In *Proc. the 2015 ICCV*, December 2015, pp.1116-1124.

[27] Zheng L, Yang Y, Hauptmann A G. Person re-identification: Past, present and future. arXiv:1610.02984, 2016. http://arxiv.org/abs/1610.02984, March 2019.

[28] Sun Y, Zheng L, Deng W, Wang S. SVDNet for pedestrian retrieval. In *Proc. the 2017 ICCV*, October 2017, pp.3820-3828.

[29] Li W, Zhao R, Xiao T, Wang X. DeepReID: Deep filter pairing neural network for person re-identification. In *Proc. the 2014 CVPR*, June 2014, pp.152-159.

[30] Liao S, Hu Y, Zhu X, Li S Z. Person re-identification by local maximal occurrence representation and metric learning. In *Proc. the 2015 CVPR*, June 2015, pp.2197-2206.

[31] Zhang L, Xiang T, Gong S. Learning a discriminative null space for person re-identification. In *Proc. the 2016 CVPR*, June 2016, pp.1239-1248.

[32] Hermans A, Beyer L, Leibe B. In defense of the triplet loss for person re-identification. arXiv:1703.07737, 2017. http://arxiv.org/abs/1703.07737, March 2019.

[33] Feris R, Bobbitt R, Brown L, Pankanti S. Attribute-based people search: Lessons learnt from a practical surveillance system. In *Proc. the 2014 Int. Conf. Multimedia Retrieval*, April 2014, Article No. 153.

[34] Xu J, Jagadeesh V, Ni Z *et al.* Graph-based topic-focused retrieval in distributed camera network. *IEEE Transactions on Multimedia*, 2013, 15(8): 2046-2057.

[35] Yang L, Luo P, Change Loy C, Tang X. A large-scale car dataset for fine-grained categorization and verification. In *Proc. the 2015 CVPR*, June 2015, pp.3973-3981.

[36] Liu X C, Ma H D, Fu H Y, Zhou M. Vehicle retrieval and trajectory inference in urban traffic surveillance scene. In *Proc. the 2014 Int. Conf. Distributed Smart Cameras*, November 2014, Article No. 26.

[37] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. the 2015 Annual Conference on Neural Information Processing Systems*, December 2015, pp.91-99.

[38] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proc. the 2016 CVPR*, June 2016, pp.770-778.

[39] Russakovsky O, Deng J, Su H *et al.* ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015, 115(3): 211-252.

[40] Chopra S, Hadsell R, LeCun Y. Learning a similarity metric discriminatively, with application to face verification. In *Proc. the 2015 CVPR*, June 2005, pp.539-546.

[41] Muja M, Lowe D G. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014, 36(11): 2227-2240.

[42] Liu X C, Liu W, Mei T, Ma H D. PROVID: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia*, 2018, 20(3): 645-658.

[43] Yan Y, Ni B, Song Z *et al.* Person re-identification via recurrent feature aggregation. In *Proc. the 14th European Conference on Computer Vision, Part VI*, October 2016, pp.701-716.

[44] You J, Wu A, Li X, Zheng W S. Top-push video-based person re-identification. In *Proc. the 2016 CVPR*, June 2016, pp.1345-1353.

Xin-Chen Liu received his Ph.D. degree at the Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing, in 2018, and B.E. degree in computer science from Northwest Agricultural and Forestry University, Xi'an, in 2011. He received the Best Student Paper Award at ICME in 2016. His research interests include multimedia content analysis and computer vision.

Wu Liu is currently a senior researcher in JD AI Research, Beijing. He was a lecturer in Beijing University of Posts and Telecommunications from 2015 to 2018. He received his B.E. degree in software engineering from Shandong University, Jinan, in 2009, and Ph.D. degree in computer application technology at the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, in 2015. His current research interests include video analytics and human behavior analysis. He has authored or co-authored over 30 papers in prestigious conferences and journals in computer vision and multimedia. He received CAS Outstanding Ph.D. Thesis Award in 2016, Best Student Paper Award at ICME in 2016, the Deans Special Award of CAS in 2015, etc.

Hua-Dong Ma received his Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 1995. He is currently a Chang Jiang Scholar Professor, the director of the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, and the executive dean of the Institute of Network Technology with Beijing University of Posts and Telecommunications, Beijing. He has published more than 200 papers and five books in his areas of interest. He was a recipient of National Funds for Distinguished Young Scientists in 2009. He is an associate editor of IEEE Transactions on Multimedia, ACM Transactions on Internet of Things, and IEEE Internet of Things Journal. He is a fellow of CCF and Outstanding Council Member. His current research focuses on Internet of Things, sensor networks, and multimedia system and networking.

Shuang-Qun Li is a Ph.D. candidate at Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing. He received his B.S. and M.S. degrees in computer science and computer application technology respectively, from the College of Computer and Information Engineering, Henan Normal University, Xinxiang, in 2001 and 2011, respectively. His research interests include gait recognition, person re-identification, computer vision, and deep learning.