Spatio-Temporal Vehicle Trajectory Recovery on Road Network Based on Traffic Camera Video Data

Fudan Yu*
Department of Electronic Engineering, Tsinghua University
Beijing, China

Wenxuan Ao*
Department of Electronic Engineering, Tsinghua University
Beijing, China

Huan Yan†
Department of Electronic Engineering, Tsinghua University
Beijing, China

Guozhen Zhang
Department of Electronic Engineering, Tsinghua University
Beijing, China

Wei Wu
SenseTime Research
Beijing, China

Yong Li
Department of Electronic Engineering, Tsinghua University
Beijing, China

ABSTRACT

Large-scale vehicle trajectories bring great benefits in understanding urban mobility, and can be used to promote a wide range of applications in building intelligent transportation systems. Traditional approaches cannot recover the trajectories of all the vehicles on the roads since they are based on partial trajectory data. To address it, we study the all-vehicle trajectory recovery based on traffic camera video data. However, there are two challenges in this study. First, the quality of the images captured by traffic cameras is unbalanced, so it is hard to identify the same vehicles. Second, the traffic camera observation data are sparse due to the incompleteness of the traffic cameras and possible vehicle miss from the traffic cameras. To deal with these challenges, we design a novel system to recover the vehicle trajectory with the granularity of the road intersection. In this system, we propose an iterative framework to jointly optimize the vehicle re-identification and trajectory recovery tasks. In the vehicle re-identification task, we propose an effective strategy to guide the vehicle clustering based on visual features and the spatio-temporal constraint features updated by the trajectory discovery task. In the trajectory recovery task, we model the spatial and temporal relations as well as the vehicle miss problem by a probabilistic approach to recover the trajectories. Extensive experiments demonstrate that our framework outperforms the existing state-of-art solutions. Finally, our system is deployed in practical applications of SenseTime, China, including traffic congestion analysis and traffic signal control.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; *Mobile information processing systems*; *Data mining*.

∗ Both authors contributed equally to this research.
† Corresponding author. Email: yanhuan@tsinghua.edu.cn.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539186

KEYWORDS

Vehicle trajectory recovery, spatio-temporal modeling, urban computing

ACM Reference Format:
Fudan Yu*, Wenxuan Ao*, Huan Yan†, Guozhen Zhang, Wei Wu, and Yong Li. 2022. Spatio-Temporal Vehicle Trajectory Recovery on Road Network Based on Traffic Camera Video Data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539186

1 INTRODUCTION

Large-scale vehicle trajectories play an important role in understanding urban mobility, which brings great benefits to many applications in intelligent transportation systems, such as route planning, traffic condition prediction, video surveillance, and traffic signal control. The development of GPS-enabled devices provides an opportunity to record vehicle trajectories anywhere and anytime. However, because of the constraints of devices, the quality of trajectory data has uncertainty. Meanwhile, it is hard to share data from different providers because of business and user privacy protection, so the trajectories are usually biased towards a small number of vehicles. Existing works [2, 20] study the trajectory recovery based on low-sampling GPS data, which cannot well address the above issues at the same time. Thus, it motivates us to design an effective approach to recover the trajectories of all the vehicles in the road network based on unbiased data.

Nowadays, the popularity of traffic cameras deployed at road intersections makes it possible to obtain the trajectory data of all the vehicles. To be specific, traffic cameras record all the vehicles passing by the intersections at different times in terms of videos or images. By utilizing large-scale video or image data from the city-wide camera network, it is expected to recover the full-amount vehicle trajectories. To sum up, our goal is to utilize the video data obtained from traffic cameras to recover the vehicle trajectories.

To achieve this goal, we collect one-day-worth video data from 441 traffic cameras in a metropolis. Intuitively, a vehicle re-identification module and a spatio-temporal recovery module should be introduced. The vehicle re-identification module is adopted to extract the vehicle visual features based on camera video data and perform the clustering algorithms to identify the same vehicles. Based on the results of vehicle re-identification, the spatio-temporal

module integrates the spatial and temporal constraints to recover the intersection-level vehicle trajectories. However, due to the influence of several factors like the environment and the camera coverage rate, there are two challenges in addressing the trajectory recovery problem.

- **Unbalanced quality of the captured images**. Many factors have an impact on the quality of the image data captured by traffic cameras. For example, poor illumination or low resolution makes the visual appearances of vehicles ambiguous. On the contrary, good illumination or high resolution enables clear visual appearances. These lead to the unbalanced quality of the captured images belonging to the same vehicles, which further influences the results of vehicle re-identification. In other words, the cluster of a vehicle may contain the records belonging to other vehicles. Meanwhile, the records of the same vehicle would fall into other clusters. More importantly, the inaccurate results of vehicle re-identification would degrade the performance of the trajectory recovery. Thus, it is challenging to accurately identify the same vehicles.
- **Sparse traffic camera observations.** On the one hand, because of high economic costs, many intersections may not have traffic cameras installed, which indicates that the trajectories of the same vehicle cannot be fully tracked. Hence, for two consecutive observations from the same vehicle, there is more likely to be more than one possible path between them. On the other hand, it is possible to miss capturing the vehicles due to some factors like low camera performance or high vehicle driving speed. These lead to the sparsity of the traffic camera observations. Thus, it is challenging to explore how to accurately recover the real trajectories based on the sparse traffic camera observations.

To tackle the above challenges, we propose an iterative framework to jointly optimize the vehicle re-identification and trajectory recovery tasks based on traffic camera video data. In this framework, a vehicle re-identification module and a spatio-temporal recovery module interact with each other to improve their own performance. To be specific, in the vehicle re-identification module, we utilize the results of trajectory recovery with the spatial and temporal constraints to guide the clustering process, which deals with the problem of the unbalanced quality of the captured images. A novel strategy is designed to dynamically adjust the feature representations of vehicles. In this strategy, the feature representations of vehicles are divided into static and dynamic parts. The static part is the vehicle visual feature representations, and the dynamic one contains dynamic vehicle feature representations updated according to the spatio-temporal constraints. In the spatio-temporal recovery module, we introduce a probabilistic approach that integrates both temporal and spatial relations to address the challenge of the sparsity of the camera observations. Considering that traffic cameras may fail to capture a certain amount of vehicle images, we model it in a probabilistic way under the condition that the candidate paths have cameras.

Finally, we summarize our contributions as follows:

- We propose an iterative framework for the vehicle trajectory recovery, which tackles two critical tasks including vehicle re-identification and trajectory recovery. The vehicle re-identification task utilizes the recovered trajectories with the constraints of spatial and temporal information to improve the performance of vehicle clustering. With the enhanced vehicle clusters, the trajectory recovery task can further obtain more accurate recovered trajectories.
- For the vehicle re-identification, we design a novel strategy that guides the clustering process by dynamically adjusting the input of the vehicle clustering algorithm, based on the visual features and the spatio-temporal constraint features updated by the trajectory recovery task.
- For the trajectory recovery, we propose a probabilistic spatio-temporal vehicle trajectory recovery model, which considers the case that traffic cameras fail to capture the vehicle images with some probability.
- We conduct several experiments based on the real-world data from traffic cameras. The results show that the performance of our framework is superior to that of the state-of-art baselines.
- We deploy our system in the practical applications of SenseTime, China, which can accurately identify the vehicles and recover the vehicle trajectories.

2 PRELIMINARIES

We describe the key definitions and problem statement in this section.

2.1 Definition

Road Network: we denote the road network as $G = (V, E, W)$, where V and E represent the set of nodes and edges respectively. $v \in V$ is the road intersection. $e \in E$ denotes the connection relationship between intersections, whose weight $w(e_i, e_j)$ is defined as the geological distance between intersection i and j.

Traffic Camera Record: each record of traffic camera is represented as $< c, t, p, v >$, which means that camera c captured vehicle p at the road intersection v at time t. This record can be extracted from the raw video data in terms of the vehicle image. Note that camera c is located at the intersection v, which is associated with geographical coordinates of intersection i in terms of longitude and latitude.

Intersection-level Vehicle Trajectory: a vehicle trajectory W_p is denoted as a time sequence $[(v_1, t_1), (v_2, t_2), ..., (v_n, t_n)]$ at the intersection level, where each element represents that a vehicle p passes the intersection v_i at time t_i. It is noted that the our work focus on the continuous vehicle trajectory, where intersection v_i and v_{i+1} is adjacent.

GPS-based Vehicle Trajectory: we define a GPS-based vehicle trajectory X_p as a time sequence $[(d_1, t_1), (d_2, t_2), ..., (d_n, t_n)]$, where each element represents a vehicle record with the GPS position d_i and timestamp t_i.

2.2 Problem Statement

Given the vehicle image data S from N traffic cameras, as well as the historical GPS-based vehicle trajectories X, our goal is to recover the intersection-level vehicle trajectory W_p based on the

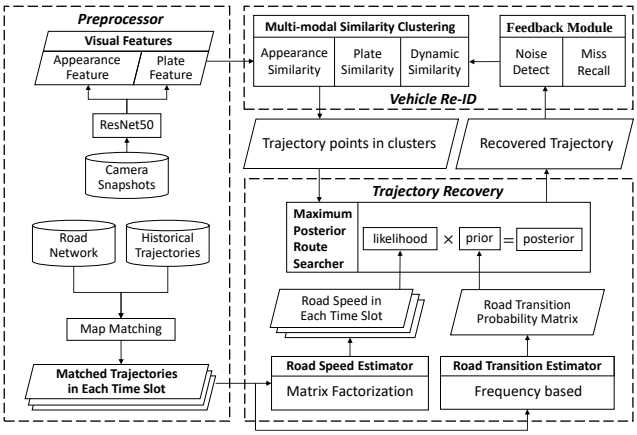

Figure 1: The overall framework of our system.

road network G, which can be expressed as:

$$W_p \leftarrow \mathcal{F}(S, X, G), \tag{1}$$

where $\mathcal{F}(\cdot)$ is the function that learns to recover the vehicle trajectories.

3 OUR APPROACH

3.1 Overall Framework

As displayed in Figure 1, the overall framework consists of a preprocessor, a vehicle re-identification module, and a trajectory recovery module. The three kinds of raw data are first preprocessed: visual features including appearance features and plate features are extracted from camera snapshots following the same way as [7], and a modern map-matching algorithm [23] is used to match the historical trajectories with the road network.

In the vehicle re-identification module, a multi-modal similarity clustering algorithm takes not only the visual features as the static part of input but also a self-supervised embedding as the dynamic part of input which is updated by the feedback module.

In the trajectory recovery module, a road speed estimator calculates the speed distribution of each road in each time slot (1h) from the map-matched trajectories where a matrix factorization approach is taken to tackle the sparseness issue. Meanwhile, a road transition estimator proposes a transition probability matrix based on the transition frequencies from the map-matched trajectories at each node on the road network depicting the prior probabilities from its predecessor roads to its successor roads. Then, a maximum posterior route searcher takes the clustering results as input and searches the maximum a posterior route to recover the trajectory between every two consecutive records in a cluster, where the travel time between the two records is considered in the likelihood part and the road transition is considered in the prior part.

As an iterative framework, the feedback module calculates the spatio-temporal feasibility score of the recovered trajectories, based on which it can detect noisy records and complement missing records. The dynamic embeddings of the records are updated accordingly so that the clustering results will be improved in the next iteration.

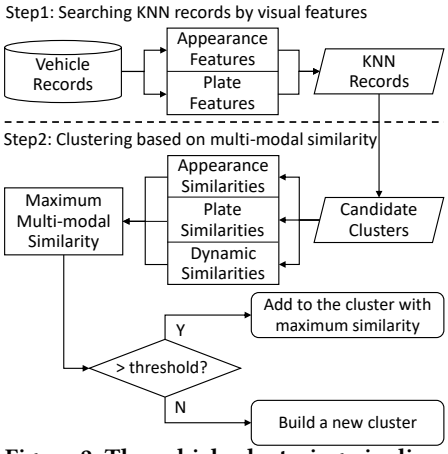

Figure 2: The vehicle clustering pipeline.

3.2 Vehicle Re-identification

3.2.1 Vehicle Feature Representation. The raw video data captured by traffic cameras are sampled and cropped to form a collection of individual vehicle images. We use two separately pretrained and finetuned ResNet-50 model to extract 256-dimensional vehicle appearance features and 256-dimensional vehicle plate features from the images. However, we remark that the plate feature is not always available due to the low quality of the image, the obstruction of other vehicles, or the viewing angle.

3.2.2 Vehicle Clustering. As mentioned above, the multi-modal similarity clustering algorithm takes as input the appearance feature $\boldsymbol{f}_a$ and plate feature $\boldsymbol{f}_p$ of each vehicle, as well as a dynamic embedding $\boldsymbol{f}_d$ which is initialized the same as the appearance feature and updated by the feedback module in our iterative framework. For two records i and j, we define their similarity as the weighted sum of the *cosine* similarity of their features. When the plate feature is not available for either of the two records, the overall similarity is calculated without the plate feature similarity. Assuming that the features are normalized, the total similarity is

$$S_{i,j} = \begin{cases} \dfrac{w_a \boldsymbol{f}_a^i \cdot \boldsymbol{f}_a^j + w_p \boldsymbol{f}_p^i \cdot \boldsymbol{f}_p^j + w_d \boldsymbol{f}_d^i \cdot \boldsymbol{f}_d^j}{w_a + w_p + w_d}, & \text{if } \boldsymbol{f}_p^i \text{ and } \boldsymbol{f}_p^j \text{ available,} \\ \dfrac{w_a \boldsymbol{f}_a^i \cdot \boldsymbol{f}_a^j + w_d \boldsymbol{f}_d^i \cdot \boldsymbol{f}_d^j}{w_a + w_d}, & \text{if } \boldsymbol{f}_p^i \text{ or } \boldsymbol{f}_p^j \text{ unavailable.} \end{cases} \tag{2}$$

The weights w_a, w_p, w_d are hyper-parameters. We generally assign the plate feature with a larger weight than the appearance feature, since vehicle appearance can be rather confusing. It is common for different vehicles to have similar appearances while the same vehicle can appear quite different under various lighting conditions.

The clustering algorithm is two-fold, as shown in Figure 2. First, we search each vehicle record's top k nearest neighbors among all the vehicle records by the appearance features and the plate features respectively, and the similar records are gathered to form its *KNN records set*. Second, we go through the records one by one and decide whether to add them to existing clusters or to build new clusters based on multi-modal similarity. Specifically, for each *current record*, we calculate its multi-modal similarities with *candidate clusters* that contain its KNN records. If the maximum similarity is greater than a predefined threshold, the record is added to the cluster with the

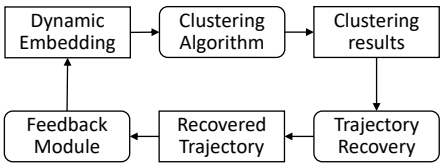

Figure 3: The iteration pipeline.

maximum similarity; otherwise, a new cluster containing only this record is built.

Note that clustering algorithms like K-Means and HDBSCAN are not suitable for our task. The K-Means algorithm requires the number of clusters as a hyper-parameter, which in our task is the number of vehicles that is unknown and hard to tune. Likewise, the demanding requirement of time and memory of HDBSCAN algorithm renders it implausible to be used in the case of millions of inputs and hundreds of thousands of clusters.

3.3 Spatio-temporal Vehicle Trajectory Recovery

In the real-world transportation systems, traffic cameras are not installed at all the intersections, and vehicles are not captured every time they pass by a camera. Thus, using the records of the vehicle alone is not sufficient to fully determine its trajectory. To address it, we use a spatio-temporal vehicle trajectory recovery module to recover the most probable trajectory between consecutive records.

Given the start point r_s, start time t_s, end point r_e and end time t_e, we denote the trajectory connecting the two points as $p = \{s_1, ..., s_n\}$, where s_i represents the road segment. Let $\Delta t = t_e - t_s$. The posterior probability of the trajectory given the above information can be factorized into two parts:

$$\begin{aligned}\Pr(p|r_s, t_s, r_e, t_e) &\propto \Pr(p, \Delta t|r_s, r_e, t_e) \\ &= \Pr(p|r_s, r_e, t_e)\Pr(\Delta t|p, t_e) \\ &\approx \Pr(p|r_s, r_e)\Pr(\Delta t|p, t_e). \end{aligned} \tag{3}$$

The approximation is due to the weak dependence of the choice of trajectory with the time of the day given r_s and r_e.

Intuitively, the first factor is a prior probability that drivers who intend to move from r_s to r_e will choose this trajectory as their route, which accounts for the general popularity of the trajectory. The second factor is the likelihood that Δt is taken to travel along this trajectory in a certain time slot of a day, which accounts for the consistency between the actual travel time and the expected travel time determined by the real-time traffic condition.

For the first factor, we assume that the transition from one road segment to another is independent from the start point r_s and satisfies Markov property given the end point r_e, which is both intuitive and widely adopted in other works [6, 7, 14, 20].

$$\begin{aligned}\Pr(p|r_s, r_e) &= \Pr(s_1, ..., s_n|r_s, r_e) \\ &= \Pr(s_1|r_s, r_e)\prod_{i=1}^{n-1}\Pr(s_{i+1}|s_i, r_e). \end{aligned} \tag{4}$$

We refer to $\Pr(s|r, r_e)$ as start segment probability and $\Pr(s'|s, r_e)$ as segment transition probability. To obtain the probability values, we use a uniform Dirichlet prior and fit the model with the data of 147,661 vehicle GPS trajectories collected in 24 hours.

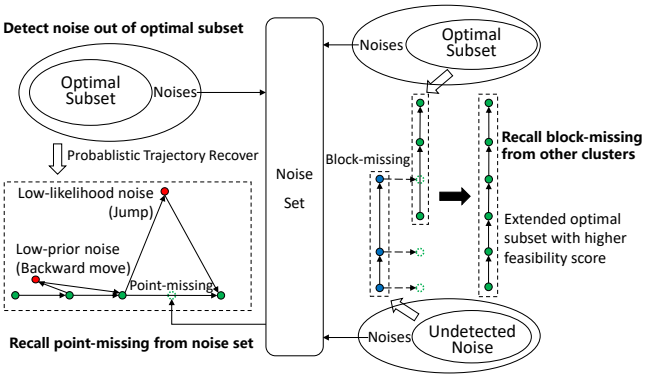

Figure 4: The feedback module for detecting noises and recalling missing records.

For the second factor, we quantize t_e into 24 time slots with the length of 1 hour and calculate the average driving speed on each road by time slot. We also adopt the matrix factorization method proposed in [20] to solve the data sparsity issue. Assuming that the relative deviation of average time follows a normal distribution, the likelihood of the total traveling time can be expressed as

$$\Pr(\Delta t|p, t_e) = \exp(-(\overline{\Delta t}/\Delta t - 1)^2/2\sigma^2), \tag{5}$$

where $\overline{\Delta t}$ is the sum of the average traveling time of p at time slot t_e, and σ is a hyper-parameter finetuned around the empirical standard deviation of the dataset.

To find optimal p^* that maximizes Equation 3, we use a greedy search algorithm that starts at r_s, expands outward, and maintains a set of at most k best trajectories until r_e is reached.

3.4 Co-optimization of Vehicle Re-identification and Trajectory Recovery

The trajectory recovery module takes the clustering results as input and recovers the maximum a posteriori trajectories. On the one hand, the best achievable accuracy of the recovered trajectories is bounded by the quality of the clustering results. Specifically, the mis-clustered records (noises) should be as few as possible for the recovered trajectory to faithfully reflect the true trajectory of a single vehicle. The more records correctly recalled, the less uncertain the possible trajectories are, leading to a more accurate recovering result. On the other hand, the infeasibility of the recovered trajectory can serve as a clue to underlying noises and missing records.

Therefore, an iterative pipeline is designed to achieve the co-optimization of vehicle re-identification and trajectory recovery, which is shown in Figure 3. In the following part, we introduce how spatio-temporal information is incorporated in the feedback module to detect noises and recall missing records to refine and complement clustering results, as well as how to update the dynamic embedding part of the record feature accordingly. An illustration of how the feedback module works is shown in Figure 4.

3.4.1 Denoising. The intuition is that if there are noises in a cluster, the probability score of the recovered trajectory will be low. For example, the trajectory suffering from noisy records may consecutively pass two distant records in a short time, or demonstrates

abnormal behaviors like driving in opposite directions. These anomalies are reflected in our probabilistic model as low speed probability (deviation from mean road speed) and low road transition probability. Therefore, we can use the spatio-temporal trajectory recovery method described in Section 3.3 to find an optimal subset of records that achieves the highest trajectory feasibility score so that the records outside of the optimal subset are detected as noises.

Specifically, given a set of records in chronological order, $R = \{r_1, ..., r_n\}$, we denote the set of all possible trajectories that pass through the records in R sequentially as $\mathcal{P}(R)$. For each trajectory $P \in \mathcal{P}(R)$, its score is calculated using a trajectory scoring function $f(\cdot)$. We aim to find the optimal subset $R^* = \{r_1^*, ..., r_{m^*}^*\} \subset R$ that maximizes the score of the highest-scored trajectory, i.e.

$$R^* = \arg\max_{R' \subset R} \max_{P \in \mathcal{P}(R')} f(P). \tag{6}$$

For the outer maximization, we adopt a greedy approach that enumerates the k largest subsets of R instead of examining all $2^n - n - 1$ possible subsets for the sake of efficiency as the number of all possible subsets grows exponentially. Besides, it is reasonable to search only large subsets since the number of underlying noises generally takes up only a small portion.

As for the inner maximization, note that for $R' = \{r'_1, ..., r'_{m'}\}$, each trajectory P is divided into sub-trajectories $p_1, ..., p_{m'-1}$ by the m' records. The scoring function is defined as compensated geometric mean of the probability of each sub-trajectory.

$$f(P) = \exp \frac{1}{m' + \alpha} \sum_{i=1}^{m'} \log \Pr(p_i), \tag{7}$$

here $\alpha > 0$ is a hyper-parameter that compensates for the size of R' so that a subset with more records is favored.

Finally, the records outside of the optimal subsets are recognized as noises, and the dynamic embedding of noise records $\boldsymbol{f}^d_{noise}$ will be update away from the average embedding of the non-noise records $\overline{\boldsymbol{f}^d_{non-noise}}$ in the next iteration, i.e.

$$\boldsymbol{f}^d_{noise,t+1} = \boldsymbol{f}^d_{noise,t} + \lambda(\boldsymbol{f}^d_{noise,t} - \overline{\boldsymbol{f}^d_{non-noise,t}}), \tag{8}$$

where λ is a hyper-parameter determining how much the noises are moved away from non-noise records.

3.4.2 Complement. The complement step tries to add missing records that mistakenly clustered into other clusters back to the cluster that they truly belong to. Practically, there are two major missing cases and two complement methods are designed respectively.

The first case is called point-missing, where the recovered trajectory passes through a camera but there is no corresponding record in this cluster at that camera. Hypothesis H_0 is that the vehicle is not captured by the camera. Hypothesis H_1 is that the vehicle is captured but the record is not in the cluster. Either way, we search for the record r_i with the highest visual and plate similarity to the cluster center in all the records at the camera that are marked as noise of other clusters in the denoising step. We denote the previous record and next record in the trajectory as r_{i-1} and r_{i+1}, and denote the capture rate of the camera as p_c, which is calculated from the dataset. The decision rule is

$$\Pr(r_{i+1}|r_{i-1})(1 - p_c) \underset{H_1}{\overset{H_0}{\gtrless}} \Pr(r_i|r_{i-1}) \Pr(r_{i+1}|r_i) p_c. \tag{9}$$

If H_1 is accepted, record r_i will be added to the cluster in the next iteration $t + 1$, and its dynamic embedding will be updated as the cluster mean $\overline{\boldsymbol{f}^d_t}$ at current iteration t.

$$\boldsymbol{f}^d_{r_i,t+1} = \overline{\boldsymbol{f}^d_t}. \tag{10}$$

The second case is called block-missing or batch-missing, where the records of the same vehicle are scattered into several clusters in the form of record blocks. It is very likely that the large blocks are included in the optimal subset of their clusters and thus not detected as noises. To recall these records, for each cluster, we search for the clusters with high multi-modal similarities and test one by one if merging the two clusters yields an optimal subset with a higher score. If such an optimal subset is found, the records in the new optimal subset from other clusters will be added to this cluster, and their dynamic embeddings will be updated as the cluster mean the same way as in Equation 10.

4 EXPERIMENTS

We conduct extensive experiments to evaluate the performance of the proposed framework based on real-world data from traffic cameras. In the experiments, we make efforts to answer the following research questions:

- **RQ1**: How is the performance of our model compared with different baselines in vehicle clustering and trajectory recovery tasks?
- **RQ2**: How do different important modules influence the performance of our model?
- **RQ3**: How do the parameter settings affect the model performance?
- **RQ4**: How does the iterative approach play the role in practical scenarios?
- **RQ5**: How is our model used in practical applications in the transportation system?

4.1 Dataset

We collect video footages in a day (8 a.m. to 8 p.m.) from 441 cameras in a metropolis. The graph constructed from the map of the area contains 2,966 edges and 1,263 nodes. The raw video footages are preprocessed into cropped images of individual vehicles, based on which 256-dimensional visual features and 256-dimensional plate features are extracted. In total, the dataset contains 4,000,000 vehicle records, which are quadruples of visual feature, plate feature, camera ID, and timestamp. The re-identification ground truth is a set of 4,760 records as one of 197 vehicles. The trajectory recovery ground truth is the GPS trajectory of the 197 vehicles map-matched onto the road network.

We also sample a smaller dataset of 1 million records from the complete dataset for a more comprehensive comparison of model performance in terms of varying dataset sizes. The full dataset is referred to as dataset D4M and the smaller one as dataset D1M.

4.2 Experimental Settings

4.2.1 Metrics. For the evaluation of trajectory recovery, we use common metrics including Longest Common SubSequence (LCSS), Edit Distance on Real sequence (EDR), and Spatio-Temporal Linear Combine distance (STLC) [15].

In addition, as the recovered trajectory is strongly affected by clustering, we also evaluate the precision, recall, F1-score and expansion of clustering results. We denote the set of all clusters as $C = \{c_1, ..., c_n\}$, and the set of ground truth vehicles as V. For each ground truth vehicle $v \in V$, we denote the set of records belonging to v as $R(v)$ and define $C(v) = \arg\max_{c \in C} |R(v) \cap c|$. Then the average precision, recall, F1-score and expansion are formulated as

$$\text{Precision} = \frac{1}{|V|} \sum_{v \in V} \frac{|R(v) \cap C(v)|}{|C(v)|}. \tag{11}$$

$$\text{Recall} = \frac{1}{|V|} \sum_{v \in V} \frac{|R(v) \cap C(v)|}{|R(v)|}. \tag{12}$$

$$\text{F1-score} = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

$$\text{Expansion} = \frac{1}{|V|} \sum_{v \in V} \sum_{c \in C} \mathbb{I}_{|R(v) \cap c| \neq 0}. \tag{14}$$

4.2.2 Baselines. The task of recovering vehicle trajectories directly from vehicle records captured by traffic cameras is a relatively new problem, and only a few existing works have been done on this problem.

Traditional re-identification models are not well designed or optimized for their outputs to be used for trajectory recovery. And without joint training or proper feedback mechanism, the re-identification models cannot utilize information like the feasibility of recovered trajectory to correct clustering results. As for trajectory recovery models, they usually take as input the sparse or noisy GPS trajectory of a *single* vehicle, while in our problem the noises are misidentified records of other vehicles, which are fundamentally different from the GPS sampling noise.

Therefore, for the baselines of this problem, we choose one re-identification model and two representative high-performance models that can recover vehicle trajectories from vehicle records.

BNN [11] It is a strong baseline for deep person re-identification with a novel batch normalization neck structure. We tailor this method to our problem setting for vehicle feature extraction and re-identification, and use the shortest path algorithm to recover trajectories from clustering results.

VeTrac [17] It builds a weighted graph based on the visual similarities of vehicle snapshots and employs a graph convolution process that iteratively updates the representation of vehicle snapshots according to the spatio-temporal similarities of the snapshots. We replace the HDBSCAN clustering algorithm with K-Means because even the most efficient HDBSCAN implementation[1] we manage to find cannot finish running on the dataset in a reasonable time.

MMVC [7] It uses an iterative framework to combine both vehicle clustering and trajectory recovery tasks. It proposes a visual-feature-based vehicle clustering process and then adopts an HMM map-matching algorithm to recover vehicle trajectory given the clustering results.

[1] https://github.com/scikit-learn-contrib/hdbscan

Dataset	Method	Precision	Recall	F1-score	Expansion
D1M	BNN	0.5561	0.4269	0.4830	9.0355
	VeTrac	0.7369	0.5624	0.6379	4.2944
	MMVC	0.8621	0.7971	0.8283	3.8071
	Ours	**0.8890**	**0.8254**	**0.8560**	**3.7513**
	Gain	3.1%	3.5%	3.3%	1.5%
D4M	BNN	0.3613	0.4183	0.3877	9.3959
	VeTrac	0.7093	0.5605	0.6262	5.3249
	MMVC	0.8258	0.7705	0.7972	4.2335
	Ours	**0.8557**	**0.8158**	**0.8353**	**4.0203**
	Gain	3.6%	5.9%	4.8%	5.0%

Table 1: Performance comparison of our method and baselines in terms of clustering output.

Dataset	Method	LCSS	EDR	STLC
D1M	BNN	0.7590	33.52	0.5158
	VeTrac	0.6984	21.62	0.5802
	MMVC	0.6210	16.95	0.6478
	Ours	**0.5879**	**15.22**	**0.6753**
	Gain	5.3%	10.2%	4.2%
D4M	BNN	0.8301	57.72	0.4670
	VeTrac	0.7091	27.94	0.5605
	MMVC	0.6251	17.98	0.6381
	Ours	**0.5828**	**15.52**	**0.6665**
	Gain	6.8%	13.7%	4.5%

Table 2: Performance comparison of our method and baselines in terms of trajectory recovery.

Setting	Precision	Recall	F1-score	Expansion
CSP	0.8621	0.7971	0.8283	3.8071
CDSP	0.8973	0.7977	0.8446	4.1574
CRSP	0.8580	0.8214	0.8393	3.6345
CMSP	0.8823	0.8232	0.8517	3.6142
CFSP/FULL*	0.8890	0.8254	0.8560	3.7513

*FULL and CFSP only differ in trajectory recovery, not clustering.

Table 3: The clustering performance of different settings.

Setting	LCSS	EDR	STLC
CSP	0.6210	16.95	0.6478
CDSP	0.6153	16.91	0.6513
CRSP	0.6191	18.52	0.6494
CMSP	0.6137	16.88	0.6564
CFSP	0.6137	15.81	0.6571
FULL	**0.5879**	**15.22**	**0.6753**

Table 4: The recovery performance of different settings.

4.3 Overall Performance (RQ1)

For a more comprehensive comparison, we evaluate both the clustering results and final trajectory recovery results on datasets D1M and D4M. The results are displayed in Table 1 and Table 2 respectively. We have the following observations:

- Our method consistently outperforms all the baselines in both the vehicle re-identification task and the trajectory recovery task across various metrics on both the full-sized dataset D4M and the sampled dataset D1M. The relative gain compared to the best baseline is shown in the tables.
- As D4M introduces a huge amount of vehicles and their records, the vehicle re-identification task becomes more challenging. An obvious drop in performance can be seen in all the methods. However, our method achieves an even greater gain in performance compared to that of D1M, which indicates that our method is more robust to noise and can better handle large-scale datasets and thus more suitable for real-world applications.

Setting	Cluster Module	Feedback Module			Trajectory Recovery	
		Denoising	Complement	Miss-capture	Shortest Path	Spatio-temporal
CSP	✓				✓	
CDSP	✓	✓			✓	
CRSP	✓		✓		✓	
CMSP	✓	✓	✓		✓	
CFSP	✓	✓	✓	✓	✓	
FULL	✓	✓	✓	✓		✓

Table 5: Settings for ablation study.

4.4 Ablation Study (RQ2)

To demonstrate the importance and contribution of each module to the final performance of our method, we test our method in several cases as summarized in Table 5.

The clustering performance of these settings on dataset D1M is shown in Table 3 and the trajectory recovery performance is shown in Table 4. We have the following observations:

- Comparing CSP and CDSP, we can see that the denoising step effectively removes noisy records and improves clustering precision by 4.1%, and F1-score by 2.0%. The trajectory recovery results are also improved as there is less interference from inaccurate records.
- Compared with CSP, CRSP has a higher recall and F1-score. The recall is improved by 3.0% and F1-score is improved by 1.3%. But its precision is slightly lower. This makes sense as the act of merging clusters may introduce some noises that not only are visually similar but also have a high spatio-temporal likelihood score.
- Among CSP, CDSP, CRSP and CMSP, CMSP achieves the highest recall and F1-score. It combines the strength of denoising and complement so that the noises introduced in the complement step are removed in the next iteration of denoising. And through the probabilistic modeling of the cameras' miss-capture, CFSP further improves the results.
- FULL achieves the best performance of all the above settings in the trajectory recovery results. Compared with CFSP, the incorporation of spatio-temporal information in trajectory recovery greatly improves the accuracy of the results as the maximum likelihood trajectories can better reflect real-life driving behavior and drivers' preference in road selection.

4.5 Parameter Analysis (RQ3)

One of the major parameters that influence the overall performance is the multi-modal similarity threshold in the vehicle re-identification module which determines how the clustering algorithm makes the trade-off between the precision and recall. Another important parameter is the number of iterations which decides how many times the spatio-temporal feedback is drawn from the recovered trajectory and the dynamic embedding is updated accordingly. We explore the effects of the two major parameters by fixing one of them and adjusting another and comparing the clustering result. The experiments are set on the D1M with all other parameters consistent with the experiments carried out in the overall performance part above.

Similarity Threshold. We fix the number of iterations as 3 and vary the clustering similarity threshold from 0.6 to 0.95. The result is shown in Figure 5. The precision and expansion monotonically increase with the similarity threshold and recall monotonically decreases as expected. As a result, the F1-score first rises and then

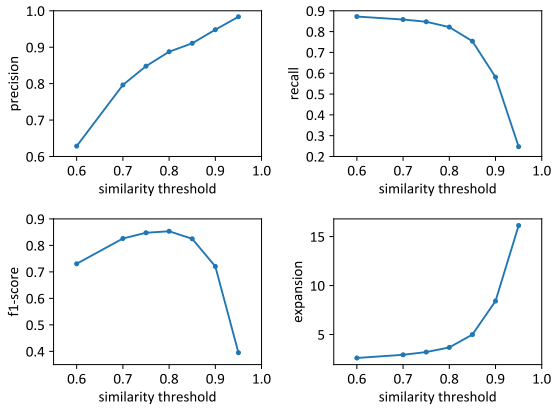

Figure 5: The influence of similarity threshold.

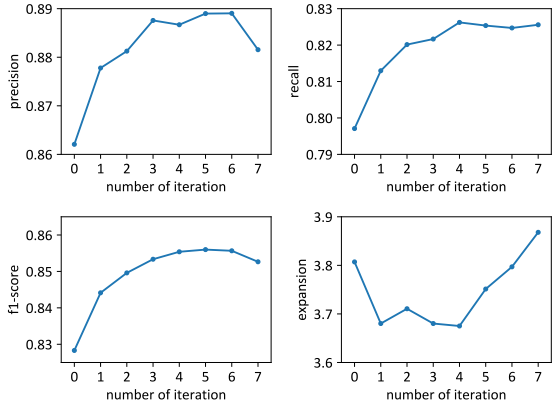

Figure 6: The influence of the number of iterations.

drops. Therefore, a moderate similarity threshold helps achieve a suitable trade-off between precision and recall for a better F1-score.

Number of Iteration. We fix the similarity threshold as 0.8 and vary the number of iterations from 0 to 7. From Figure 6, it can be seen that the precision, F1-score, and recall overall keep increasing during the first 5 iterations except for the subtle drops of precision in the 4th iteration and recall in the 5th iteration. The trend of the expansion curve also suggests that the first few iterations are beneficial. This demonstrates the effectiveness of the proposed feedback module and iterative framework. The reason why precision or recall can experience a subtle drop while F1-score rises is that both noise detection and missing complement step can make some mistakes. In general, the noise detection step increases precision at cost of recall and the missing complement step increases recall at cost of precision, while both of them boost the F1-score anyway. For example, there is a major rise in recall during the 4th iteration so that the precision drops a little. Another observation is that all the metrics go worse when the number of iterations goes too large, and this is because some hard cases exist so that the corresponding noises or missing records cannot be detected and those detectable records are already fixed during the first few iterations.

4.6 Case Study (RQ4)

To showcase the effectiveness of our iterative framework, we track the clustering result and the corresponding recovered trajectory

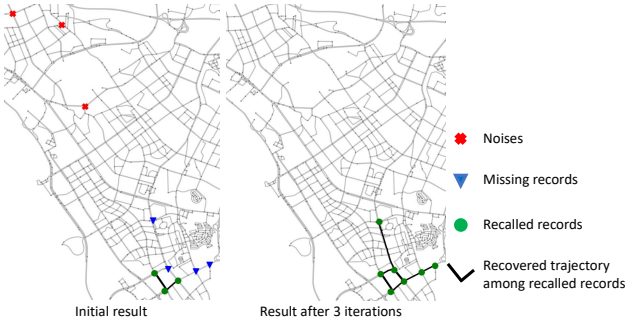

Figure 7: The initial result and the result after 3 iterations.

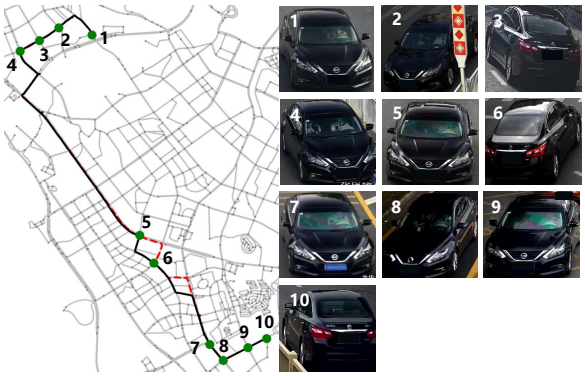

Figure 8: An example of final output of the system, including the recovered trajectory and related image records.

of a vehicle with ground truth clustering labels. As shown in Figure 7, the initial clustering result, which is fully based on visual features, suffers from some noises and missing records. However, during 3 iterations, where the feedback module detects noises and complements missing records based on spatio-temporal constraints contained in the recovered trajectory, 3 noises are removed and 5 missing records are complemented. Specifically, the 3 noises are far away from true records so that the recovered trajectory between noises and true records has a small feasibility score because of the small speed likelihood and transition prior, and the block of 5 missing records is complemented when other clusters are merged into this cluster because the extended optimal subset corresponds to a more feasible recovered trajectory.

4.7 Practical Deployment (RQ5)

The system is deployed in a district of a city in China, processing vehicle records from 673 cameras that cover an area of 76 km^2. The output trajectories are used by downstream applications, including an intelligent traffic light control system which analyzes the vehicle trajectories and adjust the phases of traffic lights. In average, the vehicle speed is improved by 20%, and the travelling time is reduced by 15.3%.

We showcase one of the recovered trajectories and the corresponding captured images in Figure 8. All the 10 captured images are retrieved from the massive dataset with millions of records. Even though there are far-away records, the trajectory is mostly correctly recovered, as can be seen by comparing the ground truth (dashed red line) and the recovered trajectory (solid black line).

5 RELATED WORK

5.1 Vehicle Re-identification

Vehicle re-identification (ReID) is an important task in intelligent transportation systems, which distinguishes the same vehicle from images or videos. Accurate ReID benefits many applications like vehicle trajectory recovery. Several existing works focus on ReID to distinguish the vehicle appearance features [1, 3, 10, 12, 27]. For instance, Zapletal et al. [25] use 3D bounding box to extract key vehicle features represented as color histogram and histogram of oriented gradients. Wang et al. [19] propose an orientation invariant feature embedding module to extract local region features of different orientations, and a spatial-temporal regularization module to model the spatial-temporal constraints. Recently, some works incorporate spatio-temporal information into the task of vehicle ReID. Liu et al. [9] exploit the spatio-temporal relations to re-rank the vehicles to improve the performance of the vehicle ReID. Authors in [14] propose to use visual-spatial-temporal path information for vehicle identification based on a two-stage framework, which employs a chain Markov random fields model to generate visual-spatio-temporal path proposals, and then adopts a Siamese-CNN+Path-LSTM model to calculate the similarity scores between paths. Unlike them, in this work, an iterative framework is designed to jointly optimize the vehicle ReID task and the vehicle recovery task, where spatial and temporal constraints in the vehicle recovery task are explored to guide the ReID process.

5.2 Vehicle Trajectory Recovery

The task of vehicle trajectory recovery is to recover high-sampling trajectories from sparsely-sampled trajectories. Most works propose their approaches to solve vehicle trajectory recovery based on GPS data from GPS-enabled mobile devices [24, 26]. Liao et al. [6] synthesize routes for low sampling trajectories based on an Absorbing Markov Chain model. Authors in [5] adopt the logit model to infer the route traveled by vehicles based on the hidden Markov model. Wu et al. [20] propose a route recovery system based on probabilistic models that integrate both spatial and temporal constraints. Banerjee et al. [2] propose a network mobility model to infer the vehicle trajectory by learning the mobility patterns that capture spatial patterns and temporal properties from historical trajectories. Recent works adopt deep learning techniques to tackle the complex factors in the vehicle trajectory recovery [13, 18, 21]. For example, Ren et al. [13] propose a map-constrained trajectory recovery model to recover the trajectories by utilizing the sequence-to-sequence multi-task learning. Different from them, our work studies the vehicle trajectory recovery based on the traffic camera data, which records all the vehicles passed by in the road network. Moreover, because of the low quality of videos or images captured by traffic cameras, it is hard to identify the sparsely-sampled trajectories for each vehicle. In other words, there are many noises in the trajectory of a vehicle, which brings the challenge to recover the trajectories. Thus, we propose to jointly optimize the vehicle ReID and the vehicle recovery tasks. Although our previous work [7] attempts to address it, it only performs the de-noise and complement process based on the clustering results, which fails to implement the iterative optimization systematically.

5.3 Multi-camera Vehicle Tracking

Multi-camera vehicle tracking aims to track the same vehicle among massive vehicles based on the vehicle images captured by cameras. Several existing works focus on tracking the targeted vehicles across multiple cameras [4, 8, 16, 22]. For example, Tang et al. [16] build the benchmark for multi-camera vehicle tracking based on a city-scale traffic camera dataset. Yan et al. [22] adopt the multi-grain ranking constraints to accurately search the vehicles with visually similar appearances of vehicle images. In our work, we recover the vehicle trajectories based on the intersection level no matter if there are cameras at the intersections, which is the main difference compared with the problem of multi-camera vehicle tracking.

6 CONCLUSION

We design a novel system to recover the vehicle trajectories based on the video data from widely deployed traffic cameras. The core of our system is an iterative framework to co-optimize both the vehicle re-identification and trajectory recovery tasks. Specifically, the vehicle re-identification task provides basic trajectory points at the intersection level for trajectory recovery based on vehicle visual features and dynamic spatio-temporal constraint features. The trajectory recovery task adopts a probabilistic approach to model spatio-temporal dependencies and vehicle miss problems for the trajectory recovery, and provides the spatio-temporal information for the vehicle re-identification task. We conduct extensive experiments to evaluate the effectiveness of our framework, and the results demonstrate that the performance of the proposed model is superior to the state-of-the-art methods. Importantly, we also deploy our system in the practical applications of SenseTime, China. This system can provide accurate results of both vehicle re-identification and intersection-level vehicle trajectory recovery, which benefits many important applications including traffic signal control and congestion analysis.

7 ACKNOWLEDGEMENTS

This work was supported in part by The National Key Research and Development Program of China under grant 2020YFB2104005, The National Nature Science Foundation of China under U20B2060, 61971267, U21B2036.

REFERENCES

[1] Yan Bai, Yihang Lou, Feng Gao, Shiqi Wang, Yuwei Wu, and Ling-Yu Duan. 2018. Group-sensitive triplet embedding for vehicle reidentification. *IEEE Transactions on Multimedia* 20, 9 (2018), 2385–2399.

[2] Prithu Banerjee, Sayan Ranu, and Sriram Raghavan. 2014. Inferring uncertain trajectories from partial observations. In *2014 IEEE International Conference on Data Mining*. IEEE, 30–39.

[3] Bing He, Jia Li, Yifan Zhao, and Yonghong Tian. 2019. Part-regularized near-duplicate vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3997–4005.

[4] Zhiqun He, Yu Lei, Shuai Bai, and Wei Wu. 2019. Multi-Camera Vehicle Tracking with Powerful Visual Features and Spatial-Temporal Cue.. In *CVPR Workshops*. 203–212.

[5] George Rosario Jagadeesh and Thambipillai Srikanthan. 2014. Robust real-time route inference from sparse vehicle position data. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 296–301.

[6] Chengxuan Liao, Jiaheng Lu, and Hong Chen. 2011. Synthesizing routes for low sampling trajectories with absorbing Markov chains. In *International Conference on Web-Age Information Management*. Springer, 614–626.

[7] Zongyu Lin, Guozhen Zhang, Zhiqun He, Jie Feng, Wei Wu, and Yong Li. 2021. Vehicle Trajectory Recovery on Road Network Based on Traffic Camera Video Data. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 389–398.

[8] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. 2016. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2167–2175.

[9] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. 2016. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European conference on computer vision*. Springer, 869–884.

[10] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. 2017. Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia* 20, 3 (2017), 645–658.

[11] Hao Luo, Wei Jiang, Youzhi Gu, Fuxu Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. 2019. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia* 22, 10 (2019), 2597–2609.

[12] Bogdan C Matei, Harpreet S Sawhney, and Supun Samarasekera. 2011. Vehicle tracking across nonoverlapping cameras using joint kinematic and appearance features. In *CVPR 2011*. IEEE, 3465–3472.

[13] Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. 2021. MTrajRec: Map-Constrained Trajectory Recovery via Seq2Seq Multi-task Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1410–1419.

[14] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. 2017. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *Proceedings of the IEEE International Conference on Computer Vision*. 1900–1909.

[15] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. 2020. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal* 29, 1 (2020), 3–32.

[16] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. 2019. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8797–8806.

[17] Panrong Tong, Mingqian Li, Mo Li, Jianqiang Huang, and Xiansheng Hua. 2021. Large-scale vehicle trajectory reconstruction with camera sensing network. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 188–200.

[18] Jingyuan Wang, Ning Wu, Xinxi Lu, Wayne Xin Zhao, and Kai Feng. 2019. Deep trajectory recovery with fine-grained calibration using kalman filter. *IEEE Transactions on Knowledge and Data Engineering* 33, 3 (2019), 921–934.

[19] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. 2017. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *Proceedings of the IEEE international conference on computer vision*. 379–387.

[20] Hao Wu, Jiangyun Mao, Weiwei Sun, Baihua Zheng, Hanyuan Zhang, Ziyang Chen, and Wei Wang. 2016. Probabilistic robust route recovery with spatio-temporal dynamics. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1915–1924.

[21] Tong Xia, Yunhan Qi, Jie Feng, Fengli Xu, Funing Sun, Diansheng Guo, and Yong Li. 2021. Attnmove: History enhanced trajectory recovery via attentional network. *arXiv preprint arXiv:2101.00646* (2021).

[22] Ke Yan, Yonghong Tian, Yaowei Wang, Wei Zeng, and Tiejun Huang. 2017. Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In *Proceedings of the IEEE international conference on computer vision*. 562–570.

[23] Can Yang and Gyozo Gidofalvi. 2018. Fast map matching, an algorithm integrating hidden Markov model with precomputation. *International Journal of Geographical Information Science* 32, 3 (2018), 547–570. https://doi.org/10.1080/13658816.2017.1400548 arXiv:https://doi.org/10.1080/13658816.2017.1400548

[24] Yu Yang, Xiaoyang Xie, Zhihan Fang, Fan Zhang, Yang Wang, and Desheng Zhang. 2020. Vemo: Enabling transparent vehicular mobility modeling at individual levels with full penetration. *IEEE Transactions on Mobile Computing* (2020).

[25] Dominik Zapletal and Adam Herout. 2016. Vehicle re-identification for automatic video traffic surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 25–31.

[26] Kai Zheng, Yu Zheng, Xing Xie, and Xiaofang Zhou. 2012. Reducing uncertainty of low-sampling-rate trajectories. In *2012 IEEE 28th international conference on data engineering*. IEEE, 1144–1155.

[27] Yi Zhou and Ling Shao. 2018. Aware attentive multi-view inference for vehicle re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6489–6498.