

Swarm intelligence freeway–urban trajectories (SWIFTraj) dataset—Part II: Graph-based approach for trajectory connection

Xinkai Ji¹, Pan Liu^{1,✉}, Ying Yang^{2,3}, Yu Han^{4,✉}

 **Cite this article:** Ji X, Liu P, Yang Y, et al. *Commun Transp Res* 2026, 6(3): 9640036. <https://doi.org/10.26599/COMMTR.2026.9640036>

ABSTRACT: In Part I of this companion study series, we introduced swarm intelligence freeway–urban trajectories (SWIFTraj), a new open-source vehicle trajectory dataset collected using an unmanned aerial vehicle (UAV) swarm. The dataset has two distinctive features. First, by connecting trajectories across consecutive UAV videos, it provides long-distance continuous trajectories, with the longest exceeding 4.5 km. Second, it covers an integrated traffic network consisting of both freeways and their connected urban roads. However, obtaining such long-distance continuous trajectories from a UAV swarm is challenging, due to the need for accurate time alignment across multiple videos and the irregular spatial distribution of the UAVs. To address these challenges, this study proposes a novel graph-based approach for connecting vehicle trajectories captured by a UAV swarm. An undirected graph is constructed to represent flexible UAV layouts, and an automatic time alignment method based on trajectory matching cost minimization is developed to estimate optimal time offsets across videos. To associate trajectories of the same vehicle observed in different videos, a vehicle matching table is established using the Hungarian algorithm. The proposed approach is evaluated using both real-world and simulated data. The results from real-world experiments show that the time alignment error is within three video frames, corresponding to approximately 0.1 s, and that vehicle matching achieves a consistently high F1-score. These findings demonstrate the effectiveness of the proposed method in addressing key challenges in UAV-based trajectory connection and highlight its potential for large-scale vehicle trajectory collection.

KEYWORDS: unmanned aerial vehicle (UAV) swarm; trajectory reconstruction; time alignment; undirected graph; vehicle trajectory dataset

1 Introduction

High-resolution trajectory data are invaluable in road traffic flow research. They have been extensively used to understand driving

behavior, analyze the mechanisms underlying puzzling traffic phenomena, calibrate (train) and validate traffic flow models, and optimize traffic flow dynamics. In recent years, the growing availability of open-source trajectory datasets has further advanced

¹School of Transportation, Southeast University, Nanjing 210089, China. ²School of Management, Shanghai University, Shanghai 200444, China. ³Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg 41296, Sweden. ⁴Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511455, China.

✉ Corresponding authors. E-mail: P. Liu, liupan@seu.edu.cn; Y. Han, yuhan2@hkust-gz.edu.cn

Received: February 24, 2026; Revised: April 26, 2026; Accepted: June 10, 2026

© The Author(s) 2026. This is an open access article under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), <http://creativecommons.org/licenses/by/4.0/>.

research in this field. In particular, unmanned aerial vehicles (UAVs), combined with advanced computer vision techniques, have enabled the extraction of high-resolution vehicle trajectories from aerial videos (Chen et al., 2021; Liu et al., 2023; Shi et al., 2021).

Despite these advances, most existing UAV-based trajectory datasets remain limited in spatial coverage. For instance, the trajectory data from US-101 in the NGSIM dataset (Alexiadis et al., 2004), captured from high-rise buildings, cover a maximum distance of only 640 m. Similarly, the highD dataset (Krajewski et al., 2018), recorded by a single UAV along German freeways, spans approximately just 420 m. Such restricted spatial extent hinders the comprehensive observation of the full life cycle of traffic phenomena, such as capacity drop and oscillations (Chen et al., 2014). Although datasets like I24-Motion (Gloudemans et al., 2023) cover a much broader spatial range, their trajectories are discontinuous, which limits their applicability to driving behavior analysis. Therefore, long-distance continuous trajectory datasets remain scarce in the transportation field.

To address this limitation, recent studies have explored the use of multiple UAVs to collect trajectory data over extended roadway segments (Barmounakis and Geroliminis, 2020; Ma et al., 2022). A key challenge in this context is connecting vehicle trajectories across multiple UAV videos. Existing approaches can be broadly categorized into two types. The first type stitches multiple videos into one before extracting trajectories (Chen et al., 2020, 2022). Although effective for small-scale scenarios (typically two videos), this strategy generally incurs high computational costs and becomes impractical as the number of videos increases. The second type first extracts trajectories independently for each video and then connects trajectories across videos (Coifman and Li, 2024; Long et al., 2024; Raju et al., 2022). This approach significantly reduces computational complexity, making it suitable for large-scale trajectory connection tasks. Therefore, this study focuses on this second type of approach.

Nevertheless, most existing methods of the second type suffer from the following limitations. First, time alignment across videos is often overlooked, which can lead to vehicle matching errors and discontinuities in the connected trajectories. Second, many existing approaches focus on connecting trajectories within the same lane and have limited capacity to handle lane-changing scenarios. Furthermore, current connection methods (Coifman and Li, 2024; Wang et al., 2024b) are primarily designed for scenarios where multiple UAVs are arranged in a line, as illustrated in Fig. 1b. As a result, these methods are unable to handle situations where UAVs are distributed in a scattered manner, as depicted in Fig. 1a. The irregular distribution of UAVs implies that vehicle trajectories in one video may need to be connected with trajectories from three or four surrounding videos, which significantly increases the complexity of trajectory connection. In summary, the challenges of large-scale UAV-based trajectory connections can be attributed to two key aspects: time alignment across multiple videos and the irregular distribution of UAVs.

To address these challenges, this study proposes a graph-based approach for connecting vehicle trajectories (GCVT) from UAV swarms, which consists of two subtasks: time alignment and vehicle re-identification (Re-ID). Time alignment, also known as time synchronization, aligns the timestamps of vehicle trajectories captured by different UAVs. Vehicle Re-ID refers to the task of identifying the same vehicle across multiple cameras (Li et al., 2023; Wang et al., 2024a). In computer vision, vehicle Re-ID typically relies on image features for matching. However, due to

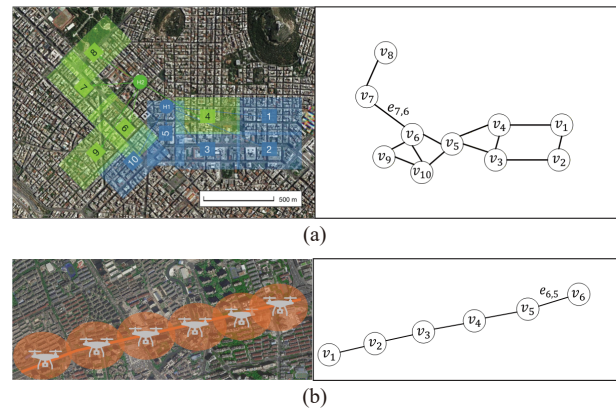


Fig. 1 Examples of video graphs. (a) Flight plan (left: reproduced with permission from Barmounakis and Geroliminis (2020), © 2019 Elsevier Ltd.) and corresponding video graph for the pNEUMA dataset. (b) Flight plan and corresponding video graph for the MAGIC dataset (Ma et al., 2022).

the high-flight altitude of UAVs and the resulting bird's-eye view perspective, vehicles in the imagery are often small, which makes image-based matching challenging. To overcome this limitation, we adopt a trajectory-based matching strategy instead of relying on image features. Specifically, vehicles are matched based on their trajectories across adjacent videos. The matching relationships are first established between vehicles in neighboring videos, and the associations are then extended to all videos. This process ensures that each vehicle maintains a consistent ID across the entire dataset.

To this end, the proposed GCVT comprises the following key steps. First, a graph is constructed to represent the UAV layout, where each video is treated as a node and edges are defined based on overlapping views, enabling the approach to handle irregular UAV distributions. Next, an automatic time alignment method based on trajectory matching cost minimization is applied to estimate pairwise time offsets, which are then accumulated to achieve global time synchronization. Finally, vehicle trajectories across adjacent videos are associated using the Hungarian algorithm (Kuhn, 1955), and continuous trajectories are generated through trajectory fusion. The effectiveness of the proposed approach is demonstrated through both simulation experiments and real-world UAV swarm experiments. Overall, the main contributions of this study are summarized as follows:

- 1) A graph-based approach is proposed for connecting vehicle trajectories captured by a UAV swarm, enabling the generation of continuous trajectories across multiple UAV videos under irregular spatial distributions.
- 2) A time offset optimization algorithm based on trajectory matching cost is proposed for the automatic time alignment of multiple videos, ensuring temporal synchronization of the trajectories across these videos.
- 3) The effectiveness of the proposed approach is evaluated through a real-world experiment conducted with a swarm of 16 UAVs. This evaluation assesses the accuracy of time alignment and vehicle matching. The results demonstrate the effectiveness of the GCVT approach in real-world scenarios. Additionally, the vehicle trajectory dataset from the 16 UAVs is available online for noncommercial research purposes.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 defines key concepts and presents the task of vehicle trajectory connection. Section 4 elaborates on the proposed GCVT approach, providing a step-by-step description of the trajectory connection process. Section 5 details the experimental setup and evaluates the performance of both time

alignment and vehicle matching. Finally, Section 6 concludes the study and outlines directions for future research.

2 Related work

2.1 Trajectory dataset

Methods for obtaining vehicle trajectories can be divided into two categories: vehicle-based methods and video-based methods (Kim and Cao, 2010). Vehicle-based methods rely on onboard sensors such as light detection and ranging (LiDAR), camera, global positioning system (GPS), and inertial measurement unit (IMU) to collect trajectories of both the ego vehicle and surrounding traffic participants (Chen et al., 2024; Coifman et al., 2016; Hu et al., 2022; Liu et al., 2025; Zhao et al., 2017). These methods are capable of capturing trajectories over long durations and long distances. However, the high cost of sensor-equipped vehicles and their low penetration rate limit their applicability for large-scale traffic flow analysis. Video-based methods, in contrast, extract vehicle trajectories from traffic videos (Chen et al., 2021; Liu et al., 2023; Shi et al., 2021; Xu et al., 2017). Compared with vehicle-based approaches, video-based methods are more cost-effective and can provide full-sample trajectory data. The emergence of UAV technology has further sparked research interest in extracting vehicle trajectories from UAV videos. UAV-based data collection offers several advantages, including a bird's-eye view for comprehensive observation of traffic dynamics, high flexibility in capturing diverse traffic scenarios, and lower deployment costs compared to vehicle-based sensing.

In recent years, UAVs have become the primary means of obtaining vehicle trajectory data. The ika research team released a series of UAV-based trajectory datasets, including the HighD (Krajewski et al., 2018), exiD (Moers et al., 2022), inD (Bock et al., 2020), and roundD (Krajewski et al., 2020), each focusing on different types of roadway sections, such as basic highway sections, ramps, intersections, and roundabouts. Additionally, the INTERACTION dataset (Zhan et al., 2019) is specifically designed for research related to driver behavior and comprises trajectory data from four scenarios: Roundabouts, unsignalized intersections, signalized intersections, and merging areas. To analyze the driving behavior of Chinese drivers and the characteristics of Chinese roads, many scholars have published vehicle trajectory data collected in China. For instance, the SIND dataset (Xu et al., 2022) provides vehicle trajectories at two-phase intersections in Tianjin, China. Chen et al. (2021) published vehicle trajectories on various urban expressway segments during both peak and off-peak hours on weekdays in Nanjing, China. Zhang et al. (2023) focused on traffic congestion scenarios and proposed vehicle trajectory data for congested highways in China, aiming to support the development of traffic jam pilot systems. Despite their value, these datasets are primarily collected by a single UAV, which inherently limits the spatial coverage of the resulting vehicle trajectories.

To obtain longer vehicle trajectories, some studies have adopted multi-UAV data collection schemes. In practical applications, such deployments require careful operational planning and safety management, including routing, scheduling, airspace coordination, and flight safety (De Luca et al., 2025; Zhen et al., 2024). The pNEUMA experiment (Barmounakis and Geroliminis, 2020) deployed a swarm of ten drones to capture a large-scale urban traffic dataset in the central business district of Athens, Greece. This effort provided a high-resolution, multi-modal traffic dataset, enabling in-depth analysis of congestion phenomena and offering a valuable resource for global researchers to develop and test traffic models. Ma et al. (2022) proposed the MAGIC dataset,

which was collected on an expressway in Shanghai, China, using six UAVs. The dataset covers a total road length of 4000 m in both directions. However, in these studies, the processing of UAV videos relies on commercial platforms such as DataFromSky. The underlying processing pipelines and algorithms are not publicly disclosed, which limits the reproducibility and transparency of the resulting datasets. In addition, the datasets produced in these studies typically cover a limited range of road types.

2.2 Trajectory connection

Methods for obtaining continuous vehicle trajectories from multiple UAV videos can generally be categorized into two types: Early connection (Chaudhari et al., 2025; Chen et al., 2020, 2022; Rajput et al., 2026; Zheng et al., 2024), in which the videos are first stitched together, and late connection (Coifman and Li, 2024; Long et al., 2024; Raju et al., 2022), in which trajectories extracted from individual videos are subsequently linked to form complete vehicle trajectories.

In early connection approach, multiple UAV videos are aligned and stitched into a single composite video, after which conventional single-video trajectory extraction methods are applied (Berghaus et al., 2024; Chen et al., 2020, 2022). This process typically relies on affine transformations to align and merge video frames into a unified video stream. While effective for scenarios involving a small number of videos, the early connection method is generally limited to stitching only two or three videos, as the computational cost increases rapidly with the number of videos. Additionally, the process of video stitching faces the significant challenge of video time alignment, also known as video time synchronization. Since different UAVs may have variations in their internal clock systems, the timestamps carried by each video may exhibit temporal discrepancies. Therefore, manual time alignment is necessary to ensure accurate synchronization of all the videos. However, as the number of videos increases, the workload for manual time alignment will significantly increase. As the number of videos grows, the manual effort required for time alignment becomes substantially, making the process increasingly complex, time-consuming, and resource-intensive.

The late connection strategy extracts vehicle trajectories independently from each video and subsequently connects them across videos. Compared with the early connection method, this strategy offers several advantages. First, it requires fewer computing resources, making it more feasible and practical for processing a larger number of videos. Second, trajectory extraction can be performed independently for each video, enabling parallel processing and significantly accelerating the overall pipeline. For these reasons, the late connection strategy is adopted in this study.

In the late connection method, a car-following-based vehicle trajectory connection approach was proposed to link broken vehicle trajectories (Long et al., 2024; Shi et al., 2021). This method frames trajectory connections across multiple videos as a problem analogous to trajectory interruptions caused by bridge occlusions. However, bridge-induced occlusions occur within a single video and do not involve time misalignment between different videos. Moreover, the method encounters difficulties when attempting to connect vehicle trajectories in scenarios where vehicles change lanes within the overlapping area of two videos, mainly due to its reliance on the car-following model. Raju et al. (2022) proposed a stitching-based algorithm to generate continuous long trajectories. However, this approach relies solely on the two endpoint data of each trajectory, which leads to poor reliability in vehicle matching. Coifman and Li (2024) introduced a partial trajectory method to

align views from successive fixed cameras, thereby facilitating video-based vehicle tracking across multiple camera views. Nevertheless, these methods do not account for time synchronization among multiple videos. When significant time offsets exist between trajectories captured by adjacent videos, achieving robust trajectory connection becomes challenging. Therefore, to overcome the limitations of existing late connection method and to meet the requirements of large-scale, multi-UAV trajectory data collection, a new trajectory connection approach that explicitly considers time alignment and complex deployment scenarios is needed.

3 Problem statement

The problem of connecting vehicle trajectories collected by multiple UAVs focuses on generating a unified and continuous trajectory representation from multiple UAV videos. In the proposed GCVT framework, vehicle trajectories extracted from individual videos constitute the primary input data. It is worth noting that video stabilization and trajectory extraction are not the focus of this study; for specific methods, please refer to our related research (Ji et al., 2025). For successful vehicle trajectory connection, two prerequisites must be fulfilled. (1) All vehicle trajectories have the same time interval. (2) Any pair of videos to be connected must have overlapping image regions, within which the vehicle trajectories are observable. Moreover, we assume that all videos are recorded at a frame rate of $1/\Delta t$, where Δt represents the time interval between consecutive frames. The vehicle trajectories are extracted at a time interval of $A\Delta t$, where A is a constant factor that determines the sampling rate of the trajectories. In this study, Δt is set to $1/30$ s, and $A = 3$. Based on these assumptions, we subsequently formulate the problem using a graph-theoretical framework to enable a rigorous mathematical representation. Next, some key definitions and the task are stated as follows.

Definition 1: Video graph. The spatial relationships among videos captured from different locations are defined as a video graph. Specifically, an undirected graph $G = (V, E)$ is constructed to represent the topological structure of the video layout. Each video is treated as a node, and the node set is defined as $V = \{v_m \mid m = 1, 2, \dots, M\}$, where M denotes the total number of videos. The edge set is defined as $E = \{e_{i,j} \mid i, j \in \{1, 2, \dots, M\}, i \neq j\}$, where an edge $e_{i,j}$ exists if and only if videos v_i and v_j have overlapping image regions.

To illustrate the structure of the video graph, the pNEUMA (Barnpounakis and Geroliminis, 2020) and MAGIC (Ma et al., 2022) datasets are used as representative examples, and their corresponding video graphs are shown in Fig. 1. Using Fig. 1a as an example, the construction of the video graph is described as follows. First, the video recorded at each collection location is represented as a node. For example, there are 10 collection locations in Fig. 1a, and the corresponding video nodes are denoted as $v_1 - v_{10}$. Next, an edge is established between two nodes if their corresponding videos share overlapping image regions. For example, since there is an overlap between v_6 and v_7 , edge $e_{7,6}$ is included in the video graph.

Definition 2: Video timestamp. The starting timestamp of video v_m is denoted as T_m . Due to the discrepancies between the UAV's onboard clock and real-world time, timestamps across different videos may not be temporally synchronized. The actual starting timestamp is denoted as $\hat{T}_m = T_m + \varepsilon_m$, where ε_m is the unknown time error of video v_m . In practice, the time error is not directly available. Instead, the measurable quantity is the pairwise time offset between two videos. For an edge connecting videos v_i

and v_j , the time offset is defined as $\delta_{i,j} = \varepsilon_i - \varepsilon_j$. When $\delta_{i,j} = 0$, the two videos are temporally aligned.

Definition 3: Main video and subvideo. In the vehicle trajectory connection problem formulated on the undirected graph $G = (V, E)$, a reference node is designated to provide a unified temporal and spatial reference across all videos. We define this reference node as the main video, denoted as v_a ($a \in [1, 2, \dots, M]$). All remaining nodes in the graph are referred to as subvideos. As the temporal reference, the timestamp of the main video is considered to represent real-world time, i.e., its time error is assumed to be zero ($\varepsilon_a = 0$). As the spatial reference, the pixel coordinate system of the main video is used as the coordinate system for the final connected trajectories.

Definition 4: Vehicle trajectory. A vehicle trajectory is defined as a time-ordered sequence of spatial positions that describes a vehicle's movement over time. Let $C^m = \{c_i^m \mid i = 1, 2, \dots, N_m\}$ denote the set of vehicles observed in video v_m , where N_m is the total number of vehicles in v_m . The trajectory of vehicle c_i^m is represented as $J_i^m = \{(p_k^i, l_k^i) \mid k = 1, 2, \dots, K\}$, where $p_k^i = (x_k^i, y_k^i)$ denotes the pixel coordinates of the vehicle center at the k -th sampled trajectory frame, and l_k^i denotes the corresponding lane ID at frame k . Trajectories are extracted at a sampling interval of $A\Delta t$, such that the k -th trajectory frame corresponds to the $(A(k-1) + 1)$ -th frame of the original video.

Definition 5: Vehicle matching table. For each edge $e_{i,j} \in E$, the goal of trajectory connection is to establish correspondences between vehicle trajectories observed in videos v_i and v_j . These correspondences are represented by a vehicle matching table $R_{i,j}$. Specifically, $R_{i,j}$ is defined as a set of matched vehicle pairs, consisting of three columns: vehicle ID in video v_i , vehicle ID in video v_j , and the trajectory matching cost. The trajectory matching cost quantifies the dissimilarity between two vehicle trajectories and is formally defined in Section 4.

Task formulation. Based on the above definitions, the vehicle trajectory connection problem is formulated as follows. **Input:** The vehicle trajectories from all videos, denoted as $J = \{J_i^m \mid m = 1, 2, \dots, M, i = 1, 2, \dots, N_m\}$, the starting timestamp of each video $T = \{T_m \mid m = 1, 2, \dots, M\}$, the video graph $G = (V, E)$, and the selected main video v_a . **Output:** The time error for each video, $\varepsilon = \{\varepsilon_m \mid m \in [1, 2, \dots, M]\}$, and the vehicle matching tables for edges, $R = \{R_{i,j}\}$. Based on the estimated time errors and vehicle matching tables, individual trajectories are temporally aligned and fused to generate continuous vehicle trajectories.

4 Methodology

4.1 Overview

The process of trajectory connection can be divided into five steps: selection of alignment edges, spatial transformation, time alignment, vehicle matching, and trajectory fusion. The overall framework of the GCVT approach is depicted in Fig. 2, which illustrates how these five steps are integrated to achieve trajectory connection from multiple UAV videos. The overall process is described as follows:

Step 1: Construct the alignment edge set, S_a , based on the video graph.

Step 2: Estimate the affine transformation matrix for each edge in S_a , and compute the cumulative transformation matrix from each subvideo to the main video, denoted as $H_{i,a}$.

Step 3: Estimate the pairwise time offsets for edges in S_a , and derive the time error for each subvideo, denoted as ε_i .

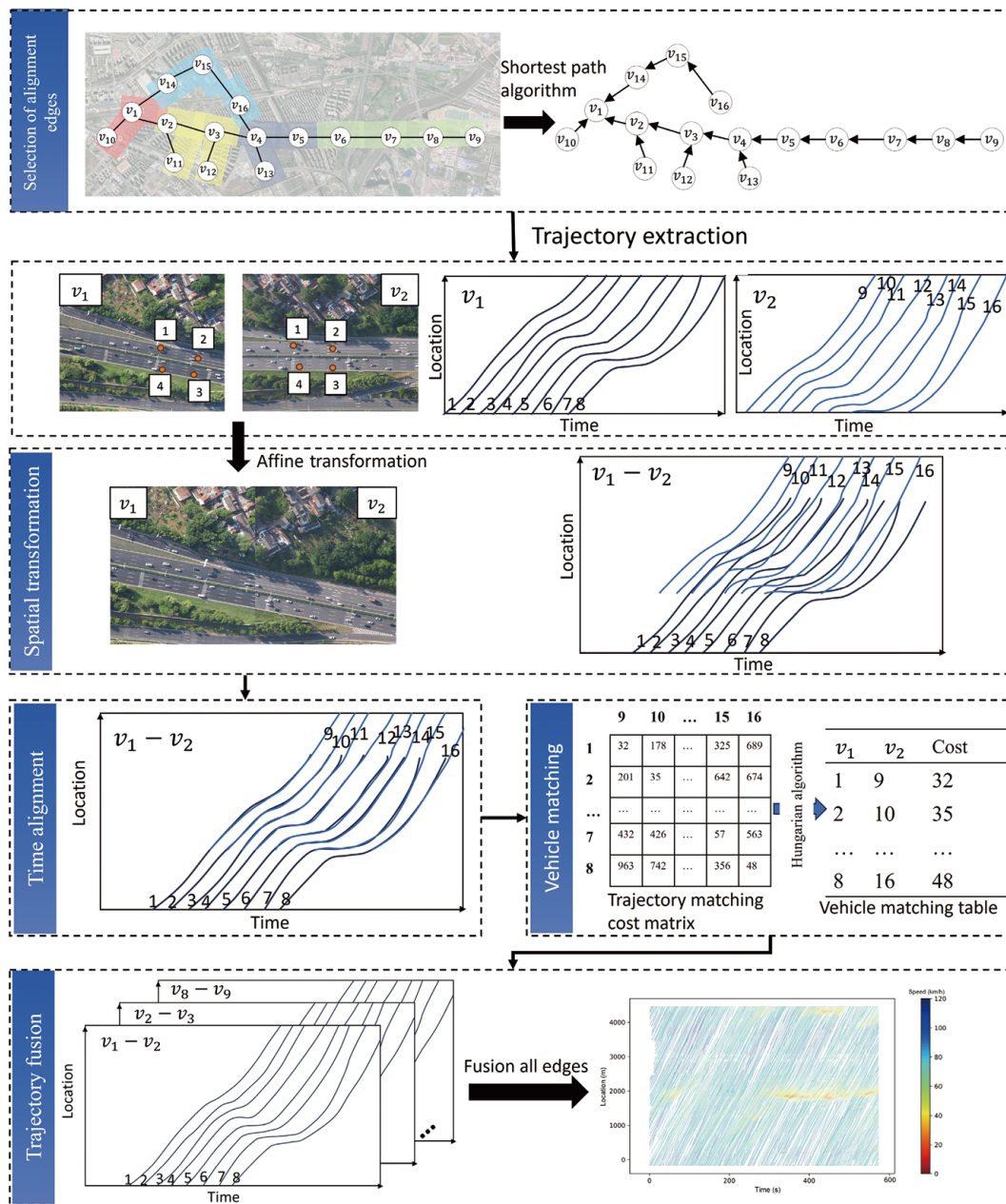


Fig. 2 GCVT framework. The subsets of trajectory data from videos v_1 and v_2 are selected to demonstrate the trajectory connection process.

Step 4: Apply spatial transformation and time alignment to the vehicle trajectories in the subvideos based on $H_{i,a}$ and ϵ_i , and subsequently compute the corresponding vehicle matching table $R_{i,j}$ for each edge in S_a .

Step 5: Use the matching table to link the vehicle trajectories and generate continuous trajectories.

Detailed formulations and algorithms for alignment edge construction, spatial transformation, time alignment, and vehicle matching are provided in specify.

4.2 Selection of alignment edges

Errors are inevitable in both time alignment and spatial transformation between video pairs. As alignment parameters are propagated along longer paths from a subvideo to the main video, pairwise errors tend to accumulate, resulting in amplified global synchronization deviations.

To mitigate error accumulation, a shortest-path-based strategy is adopted to select a compact and reliable subset of alignment edges. In this context, alignment edges refer to the graph edges

along which spatial transformation and time alignment are explicitly performed. Specifically, the video graph is treated as an unweighted directed graph with unit edge weights. For each subvideo, the shortest path to the main video is computed using a standard shortest-path algorithm, such as Dijkstra's algorithm. All directed edges that appear along these shortest paths are collected and merged to form the alignment edge set S_a . By restricting subsequent spatial and temporal alignment to this selected subset of edges, alignment parameters are propagated along the shortest paths. This approach effectively reduces cumulative errors and enhances the robustness of the overall multi-video synchronization process.

4.3 Spatial transformation

Accurate spatial transformation is critical for reliable trajectory connection. In UAV-based road scenes, automatic feature matching may become unstable due to non-coplanar structures, such as bridges or roadside objects. In such cases, background regions can be aligned while the roadway itself remains

misaligned. To ensure the reliability of spatial transformation, a manual feature selection strategy is adopted in this study. Although the proposed approach involves manual annotation, it only requires annotating the first frame of each video, resulting in a manageable annotation workload. Specifically, for each alignment edge, a set of eight feature points (four points per video) is manually selected on the road surface within the overlapping region of the two videos, as illustrated in Fig. 3. For a directed edge $e_{i,j}$, the feature points selected in video v_i are denoted by $\text{pts}_{i,j}^i$. After annotating feature points for all alignment edges in S_a , the affine transformation matrices are estimated using the procedure described below.

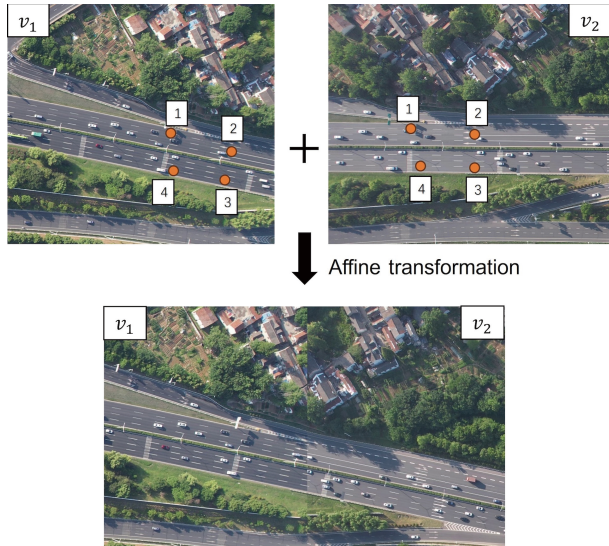


Fig. 3 Spatial transformation of images. The four orange points indicate the selected feature points.

The proposed spatial transformation algorithm consists of two main steps. First, for each space-time edge, the corresponding affine transformation matrix is estimated using the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981), which provides robustness against potential annotation noise and outliers. Second, the affine transformation matrix from each subvideo to the main video is obtained by cumulatively composing the pairwise transformation matrices along the shortest path in the video graph. The detailed procedure is summarized in Algorithm 1.

Algorithm 1 Spatial transformation

Input: Alignment edges S_a , feature point sets $\{\text{pts}_{i,j}^i\}$, video graph $G = (V, E)$, main video v_a

Output: Affine transformation matrices from all subvideos to the main video $H_{i,a}$

- 1 Initialize the affine transformation matrix estimation function $\text{getAffine}(\cdot)$;
 - 2 **foreach** edge $d_{i,j} \in S_a$ **do**
 - 3 $H_{i,j} \leftarrow \text{getAffine}(\text{pts}_{i,j}^i, \text{pts}_{i,j}^j)$;
 - 4 **end**
 - 5 **foreach** $v_i \in V \setminus \{v_a\}$ **do**
 - 6 Compute the shortest path: $P_{i \rightarrow a} \leftarrow (d_{i,u_1}, d_{u_1,u_2}, \dots, d_{u_m,a})$;
 - 7 Obtain the affine transformation by cumulative composition:
 $H_{i,a} \leftarrow H_{u_m,a} \cdots H_{u_1,u_2} H_{i,u_1}$;
 - 8 **end**
-

4.4 Time alignment

Time alignment is a critical component of the proposed GCVT framework, as its objective is to estimate the temporal offset of each subvideo relative to the main video. Accurate time offset estimation enables the synchronization of all UAV videos into a unified temporal reference. To this end, we propose an automatic time alignment algorithm based on trajectory matching cost minimization. The algorithm is built on the principle that when two videos are correctly time-aligned, vehicle trajectories within their overlapping spatial region exhibit the minimum trajectory matching cost. Owing to its sole reliance on trajectory data extracted from UAV videos, the proposed method is computationally efficient, resource-light, and broadly applicable. It can be readily extended beyond UAV swarms to other multi-camera traffic surveillance scenarios. Algorithm 2 presents an efficient numerical solution to this optimization problem.

Specifically, the algorithm begins by using the affine transformation matrix obtained in the spatial transformation step to extract vehicle trajectories within the overlapping region of the videos, thereby performing spatial filtering of the trajectories. Next, within a search window, it iterates through various frame offsets starting from 0 to find the optimal time offset. To reduce computational complexity, only vehicle trajectories within the overlapping temporal interval of the two videos—limited to the preceding 60 s—are considered. Let n_i and n_j denote the numbers of selected vehicle trajectories in videos v_i and v_j , respectively. For a given pair of videos v_i and v_j , consider trajectory i extracted from v_i and trajectory j extracted from v_j . For a candidate time offset δ , the trajectory matching cost is defined as

$$L_{i,j}(\delta) = \frac{1}{K} \sum_{k=1}^K [\|p^i(t_k) - p^j(t_k + \delta)\|_2^2 + \alpha |l^i(t_k) - l^j(t_k + \delta)|] \quad (1)$$

where $t_k = (k-1)A\Delta t$, δ denotes the candidate time offset; K is the number of frames; and $p = [x, y]$ represents the pixel coordinates of the vehicle center. Additionally, l^i and l^j denote the lane IDs of vehicles i and j , respectively. The parameter α controls the weight of the lane ID difference and is set to 100 in this study. To avoid matching vehicles that move in opposite directions, a motion direction check is applied. Specifically, the displacement between the first and last sampled positions of each trajectory is used to approximate its motion direction. If $(p_K^i - p_1^i)(p_K^j - p_1^j) \leq 0$, the matching cost is set to $+\infty$ to prevent matching spatially proximate but oppositely moving vehicles. It is worth noting that the proposed trajectory matching cost is defined directly in the two-dimensional pixel coordinate space, rather than relying on longitudinal car-following relationships. This design enables the method to handle general traffic scenarios, including lane-changing and merging.

Pairwise trajectory matching costs are computed for all selected trajectories in v_i and v_j , resulting in a cost matrix of size $n_i \times n_j$. The Hungarian algorithm is then applied to obtain the optimal trajectory matching. Matched pairs with costs exceeding a threshold θ^{td} are discarded to filter out unreliable associations. Let $\mathcal{M}$ denote the set of valid matched trajectory pairs after threshold filtering under offset δ . The optimal time offset is determined by minimizing the mean matching cost over the search window w .

$$\delta_{i,j} = \underset{\delta}{\text{argmin}} L_{\text{mean}}(\delta) = \underset{\delta}{\text{argmin}} \frac{1}{|\mathcal{M}|} \sum_{(i,j) \in \mathcal{M}} L_{i,j}(\delta) \quad (2)$$

To accelerate convergence, we adopt a bidirectional search strategy that starts from zero offset and explores both positive and negative directions. We employ an early stopping criterion to

Algorithm 2 Time offset optimal algorithm

Input: Alignment edges S_a , affine transformation matrices $H_{i,j}$, video graph $G = (V, E)$, main video v_a

Output: Time error of sub videos ε_i

- 1 Initialize search window w , $L_{\min} \leftarrow +\infty$, thresholds $\theta^{\min}$, θ^{td} , scale factor θ^{scale} , and sampling interval $A\Delta t$;
- 2 **for** each $d_{ij} \in S_a$ **do**
- 3 Transform the first frame F_j and trajectories C^j of video v_j based on the affine transformation matrix $H_{i,j}$, $\hat{F}_j \leftarrow H_{i,j} \cdot F_j$, $\hat{C}^j \leftarrow H_{i,j} \cdot C^j$;
- 4 Obtain the overlapping region $O_{i,j}$ between v_i and v_j , $O_{i,j} \leftarrow F_i \cap \hat{F}_j$;
- 5 Filter the trajectories C^i and $\hat{C}^j$ within $O_{i,j}$ to get the new trajectories C_o^i and $\hat{C}_o^j$;
- 6 **for** $f = 0$ to w **do**
- 7 **for** $d \in [-1, 1]$ **do**
- 8 $\Delta f \leftarrow fd$ and $\delta_{i,j} \leftarrow \Delta f A\Delta t$;
- 9 Update the frame index of the trajectory $\hat{C}_o^j$, $\hat{k} \leftarrow \lceil k + (T_i - T_j)/(A\Delta t) + \Delta f \rceil$;
- 10 Select trajectories C_o^i and $\hat{C}_o^j$ within the overlapping temporal window of length 60 s;
- 11 Compute cost matrix D using Eq. (1);
- 12 Apply Hungarian matching on D and discard pairs with cost $> \theta^{\text{td}}$ to obtain $\mathcal{M}$;
- 13 Compute $L_{\text{mean}}(\delta) = \frac{1}{|\mathcal{M}|} \sum_{(i,j) \in \mathcal{M}} L_{i,j}(\delta)$;
- 14 **if** $L_{\min} > L_{\text{mean}}(\delta)$ **then**
- 15 $L_{\min} \leftarrow L_{\text{mean}}(\delta)$; $d_{\min} \leftarrow d$; $\delta_{i,j}^* \leftarrow \delta_{i,j}$;
- 16 **end**
- 17 **if** $d = d_{\min}$ and $L_{\min} < \theta^{\min}$ and $L_{\text{mean}}(\delta) > \max(\theta^{\text{scale}} L_{\min}, \theta^{\min})$ **then**
- 18 Break out of two nested loops; // Early stopping strategy
- 19 **end**
- 20 **end**
- 21 **end**
- 22 **end**
- 23 **for** $v_i \in V$ where $v_i \neq v_a$ **do**
- 24 Compute the shortest path $P \leftarrow (d_{i,u_1}, d_{u_1,u_2}, \dots, d_{u_m,a})$;
- 25 $\varepsilon_i \leftarrow \delta_{i,u_1}^* + \delta_{u_1,u_2}^* + \dots + \delta_{u_m,a}^*$;
- 26 **end**

terminate the search once the cost increases significantly relative to the historical minimum. Finally, we obtain the time error for each subvideo by cumulatively summing the pairwise time offsets along the shortest path from the subvideo to the main video in the video graph.

4.5 Vehicle matching

The purpose of vehicle matching is to generate a vehicle matching table. In the time alignment step, a partial vehicle matching table has already been obtained. However, for computational efficiency, only vehicle trajectories within a 60 s window were selected for matching. To obtain the complete vehicle matching table, it is necessary to compute the trajectory matching cost matrix for all vehicle trajectories within the overlapping region after time alignment. We employ a two-step matching approach based on the Hungarian algorithm to improve the accuracy of the matching results and to prioritize vehicle pairs with lower trajectory matching costs. The specific steps are as follows:

Step 1: Set the elements in the trajectory matching cost matrix D that have values greater than a threshold θ^{matching} to infinity, resulting in the new matrix D^1 . This step ensures that smaller trajectory matching costs are given higher priority in the matching process.

Step 2: Apply the Hungarian algorithm to the new matrix D^1 to obtain the first-step matching table R^1 .

Step 3: Remove the already matched vehicles from the trajectory matching cost matrix D , resulting in an updated matrix D^2 . Then, perform the second-step matching by applying the Hungarian algorithm to D^2 to obtain the second-step matching table R^2 .

Step 4: Combine R^1 and R^2 to obtain the final vehicle matching table R .

5 Experiment

In this section, we present two experiments conducted to evaluate the effectiveness of the proposed GCVT approach. These experiments include both numerical simulations and real-world applications, thereby providing a comprehensive evaluation of the method.

Experiment 1 (Exp. 1): This experiment uses the Swarm Intelligence Freeway–Urban Trajectories (SWIFTraj) dataset, which was collected by a swarm of 16 UAVs equipped with 5.4 K-resolution cameras in a hybrid road environment that includes urban expressways, intersections, and multiple on- and off-ramps. This multi-UAV setup enables large-scale spatial coverage and provides a practical basis for graph construction, where each UAV

video is treated as a node and the overlap between neighboring videos defines the edges. The 5.4 K-resolution cameras also allow each UAV to maintain sufficient vehicle observability at a relatively high-flight altitude, which helps improve single-video trajectory quality and extend the effective coverage of each video for cross-video trajectory connection. Although these hardware settings are adopted in the present experiment, the framework itself is also applicable to other multi-UAV video data that feature overlap and reliable trajectories.

The objective is to evaluate the effectiveness of the proposed method under realistic traffic conditions. Data were collected along a segment of the G42 Hulong Expressway and its adjacent intersections in Nanjing, China, as illustrated in Fig. 4. Traffic flow was recorded during the morning peak hours (7:00 am–9:00 am) on June 16 and 17, 2022. To ensure synchronized recording and reduce battery consumption during deployment, the UAVs were organized into five groups, with each group taking off from the same location. As shown in Fig. 4, UAVs with the same color belong to the same group.

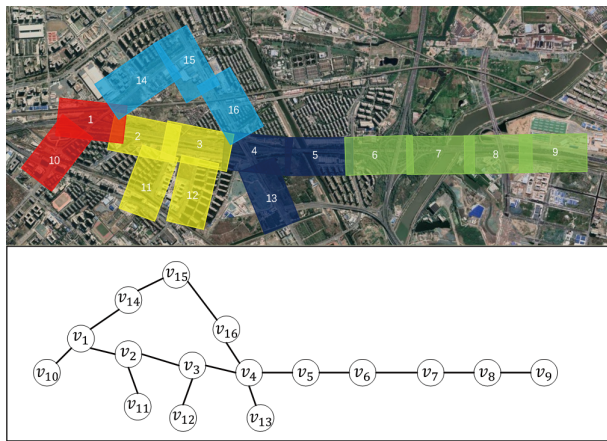


Fig. 4 Study area and video graph. (a) Satellite map of the study area and recording areas of each UAV. (b) Video graph of the UAV swarm.

Video v_1 is designated as the main video. Videos v_1 – v_9 were captured along the expressway mainline, while the remaining videos were recorded on adjacent urban roads. In the experiment, videos v_1 – v_9 , collected along the expressway mainline, were primarily used to evaluate the reliability of the proposed method in constructing long-distance vehicle trajectories on urban expressways. In addition, four video nodes—specifically, the expressway videos v_4 and v_5 and the urban-road videos v_{13} and v_{16} —were selected to assess the applicability of the proposed GCVT method in scenarios involving irregularly distributed UAVs.

Experiment 2 (Exp. 2): In this experiment, we used the microsimulation software simulation of urban mobility (SUMO) to generate synthetic trajectory data as the ground truth. This simulation enabled us to collect vehicle trajectories across various regions, each corresponding to the recording area of a UAV. The purpose of using synthetic trajectory data in this study is to evaluate the accuracy of time alignment and vehicle matching under different time offsets and spatial transformation errors. By manually adjusting these time offsets and spatial transformation errors, we can systematically test and assess the performance of the alignment and matching methods under diverse conditions, thereby ensuring their robustness and effectiveness in real-world scenarios.

In the simulation, the road network and the coverage of each virtual UAV video collection point are designed to be consistent with the real-world conditions observed in Exp. 1. The simulation

duration is set to 3600 s. Vehicle trajectories from two adjacent UAVs near the on-ramp area were selected to evaluate time alignment and vehicle matching. Since vehicles change IDs when moving across different UAV videos, vehicle matching becomes necessary. To test the accuracy of time alignment and vehicle matching under spatial transformation inaccuracies, Gaussian noise with a mean μ and standard deviation σ is added to the ground-truth position of the trajectory in one of the videos.

The GCVT approach was implemented in Python. The experimental platform used in this study is a workstation equipped with an Intel Xeon W-2255 CPU@3.9 GHz and 64 GB of memory.

5.1 Evaluation metrics

Time alignment and vehicle matching are two critical components of the proposed trajectory connection framework, and their performance directly affects the accuracy and reliability of the final connected trajectories.

The performance of time alignment is evaluated using the time offset error, which is defined as the absolute difference between the estimated time offset δ_{est} and the ground-truth time offset $\delta_{\text{gt}} = |\delta_{\text{est}} - \delta_{\text{gt}}|$.

Vehicle matching performance is assessed using standard classification metrics, including precision, recall, and F1-score. These metrics are computed based on the following definitions:

- True Positive (TP): The number of correctly matched vehicle pairs.
- False Positive (FP): The number of incorrectly matched vehicle pairs.
- False Negative (FN): The number of ground-truth matches that were not identified.

The evaluation metrics are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Higher values of Precision, Recall, and F1-score reflect greater accuracy and completeness in vehicle matching results.

5.2 Evaluation of time alignment

In this subsection, we evaluate the performance of the proposed time offset optimization algorithm using both real-world UAV video data (Exp. 1) and numerical simulation data (Exp. 2). The algorithm parameters are set to $\theta^{\min} = 100$, $\theta^{\text{td}} = 1000$, and $\theta^{\text{scale}} = 2$.

In Exp. 1, we first analyze the variation in trajectory matching cost under different candidate time offsets, using real-world trajectory data to examine the identifiability of the optimal time offset in practical scenarios. Fig. 5 illustrates the change in trajectory matching cost during the time alignment of six video pairs captured along the main road. In each subplot, the central double-headed arrow indicates the minimum trajectory matching cost, obtained through a bidirectional search that starts from zero time offset and explores both positive and negative directions. The blue solid line represents the search process with the early stopping strategy, while the orange dashed line corresponds to the search process without this strategy. The early stopping strategy significantly reduces the computational workload and improves

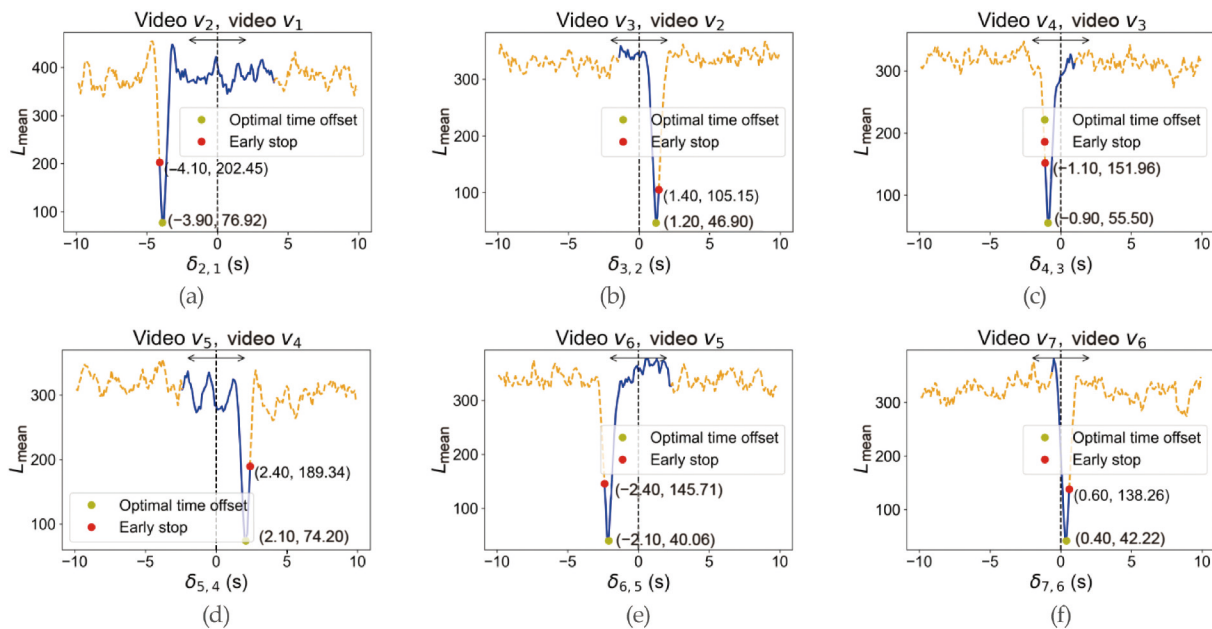


Fig. 5 Variation of the mean trajectory matching cost (defined in Eq. (1)) with different alignment edges during the search process.

the processing speed. All subplots exhibit a pronounced global minimum that is significantly lower than neighboring values, indicating that the trajectory matching cost is highly sensitive to temporal misalignment. Even small time deviations lead to noticeable increases in the trajectory matching cost, which ensures that minimizing this cost can reliably recover the optimal time offset. Moreover, the minimum trajectory matching costs across different video pairs fall within a relatively narrow range (approximately 40–80). This consistency suggests that key algorithm parameters, such as $\theta^{\min}$ and θ^{cl} , do not require scenario-specific tuning, thereby demonstrating the robustness and practical applicability of the proposed method.

Table 1 summarizes the time alignment results from Exp. 1. As shown in the table, the time offsets vary considerably across different video pairs, underscoring the necessity of explicit time alignment. The absolute error $|\delta_{\text{est}} - \delta_{\text{gt}}|$ remains small, ranging from 0 to 0.07 s. Given a video frame rate of 30 frames per second (FPS), the maximum temporal error of 0.07 s corresponds to approximately three frames, which translates to a longitudinal displacement of about 1.94 m for a vehicle traveling at 100 km/h. These results indicate that the proposed time alignment method effectively reduces the temporal offsets between adjacent videos. In practical terms, such a temporal error may lead to a meter-level longitudinal displacement in the stitching region under high-speed freeway conditions. However, this effect is localized to the connection area between adjacent videos and does not change the relative vehicle positions within each individual video. Therefore, its influence on most subsequent traffic flow and driving behavior analyses is expected to be limited.

To further validate the alignment performance, we apply the estimated time offsets to video stitching for qualitative assessment. Fig. 6 presents representative stitched images after time alignment.

Table 1 Time alignment evaluation results from Exp. 1

Alignment edge (video pair)	$\delta_{\text{est}}(\text{s})$	$\delta_{\text{gt}}(\text{s})$	$ \delta_{\text{est}} - \delta_{\text{gt}} (\text{s})$
$d_{2,1}(v_2, v_1)$	-3.90	-3.87	0.03
$d_{3,2}(v_3, v_2)$	1.20	1.17	0.03
$d_{4,3}(v_4, v_3)$	-0.90	-0.90	0.00
$d_{5,4}(v_5, v_4)$	2.10	2.10	0.00
$d_{6,5}(v_6, v_5)$	-2.10	-2.07	0.03
$d_{7,6}(v_7, v_6)$	0.40	0.33	0.07
$d_{8,7}(v_8, v_7)$	-1.50	-1.43	0.07
$d_{9,8}(v_9, v_8)$	1.70	1.67	0.03

The yellow dashed lines indicate the stitching seams between adjacent videos. By observing the vehicles near the seam, it is clear that the left and right videos are well time-aligned, demonstrating high accuracy in the estimated time offsets.

In Exp. 2, Fig. 7 illustrates the time alignment error under varying ground-truth time offsets and spatial transformation errors. Spatial transformation errors are simulated by adding Gaussian perturbations with mean μ and a fixed standard deviation of 6 to the ground-truth vehicle positions, thereby mimicking imperfect spatial alignment between overlapping UAV views. As shown in Fig. 7, for a fixed spatial offset, the time alignment error remains nearly constant across different ground-truth time offsets, demonstrating the robustness of the proposed method to varying temporal discrepancies between UAV videos. For a fixed time offset, the alignment error increases approximately linearly with the spatial offset. When the spatial offset approaches zero, the time alignment error correspondingly approaches zero, indicating that accurate spatial transformation is

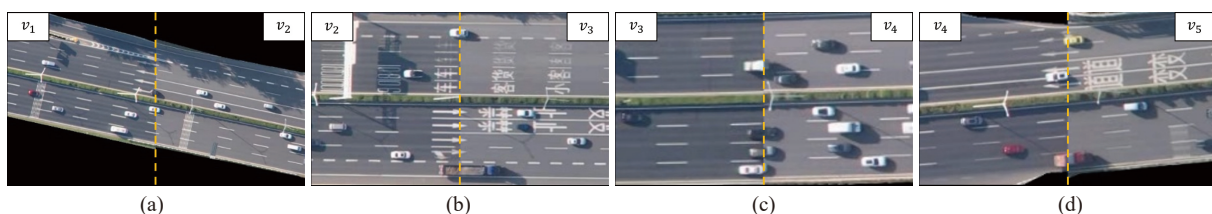


Fig. 6 Video stitching results after time alignment. The yellow dashed lines indicate the seams between adjacent stitched videos.

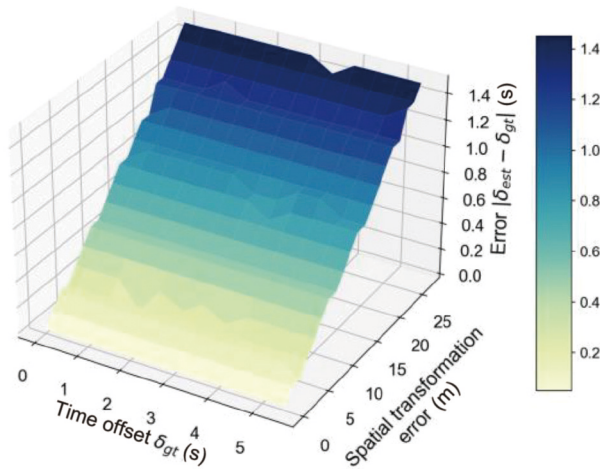


Fig. 7 Time alignment results from Exp. 2.

a prerequisite for precise temporal synchronization. These results demonstrate that the proposed time alignment algorithm achieves high accuracy under various time offsets, while also revealing a strong dependency of time alignment accuracy on the quality of spatial transformation. In this GCVT, high spatial transformation accuracy is achieved through manual annotation of feature points on the first frame of each video, which contributes to improved robustness and precision in the subsequent time alignment process.

5.3 Evaluation of vehicle matching

Vehicle matching is a critical component of the trajectory connection framework. Its accuracy depends not only on the matching algorithm itself but also on the preceding spatial transformation and time alignment stages, which ensure spatial and temporal consistency across videos. Therefore, the evaluation results reflect the overall effectiveness of the proposed GCVT framework. In this subsection, we evaluate the performance of the vehicle matching module using both numerical simulation data and real-world datasets. In addition to quantitative evaluation metrics, we visualize the connected vehicle trajectories to provide qualitative evidence of matching accuracy. The matching threshold parameter in the algorithm is set to $\theta^{\text{matching}} = 1000$.

In Exp. 1, three representative alignment edges, corresponding to a total of six videos, were selected to evaluate vehicle matching performance. The evaluation metrics, including precision, recall, and F1-score—as defined in Section 5.1—were used to

quantitatively assess vehicle matching performance. We further compared the proposed method with three baseline methods: (1) Stitched-video trajectory extraction (Chaudhari et al., 2025; Chen et al., 2020, 2022; Rajput et al., 2026; Zheng et al., 2024), in which adjacent videos are first stitched and vehicle trajectories are then extracted from the stitched videos. For this method, the time offset obtained from the time alignment step was used for synchronization; (2) appearance-based association, which performs cross-video association using only vehicle appearance embeddings extracted with the simple online and realtime tracking with a deep association metric (DeepSORT) feature extractor (Wojke et al., 2017); and (3) history-aware transformation (HAT) (Gao et al., 2025), which dynamically projects Re-ID features into a sequence-specific discriminative space based on historical trajectory features to enhance multi-object association.

The comparison results are summarized in Table 2. Overall, the proposed GCVT framework consistently outperforms the three baseline methods across all evaluated alignment edges, demonstrating its strong effectiveness and robustness for vehicle matching. Compared with the stitched-video trajectory extraction method, the proposed approach achieves better overall performance, particularly in terms of recall and F1-score on the more challenging edges. Although the stitched-video trajectory extraction attains high precision due to prior spatial-temporal alignment, its relatively low recall indicates that many true matches are still missed. This is likely because video stitching is highly sensitive to video stability and alignment quality; slight camera jitter or local misalignment near stitching boundaries can disrupt vehicle detection and tracking continuity, resulting in interrupted trajectories and missed associations. In addition, its computational cost increases substantially with the number of videos, limiting its scalability for multi-UAV trajectory extraction. The inferior performance of the appearance-based association and HAT baselines further suggests that appearance-based methods are insufficient for reliable vehicle matching in UAV-based multi-video traffic scenes, even when historical trajectory features are incorporated. One likely reason is that aerial videos provide only limited discriminative appearance information, as vehicles are primarily observed from top-down views and many of them appear highly similar. Moreover, the performance of these methods varies substantially across different alignment edges, likely because the degree of appearance distinguishability differs from one edge to another, thereby affecting the robustness of appearance-based matching. Overall, these results verify the

Table 2 Comparison of vehicle matching results

Method	Alignment edge	Precision	Recall	F1-score
Stitched-video trajectory extraction	$d_{2,1}$	0.999	0.630	0.773
	$d_{3,2}$	0.998	0.975	0.987
	$d_{4,3}$	1.000	0.699	0.823
Appearance-based association	$d_{2,1}$	0.449	0.451	0.450
	$d_{3,2}$	0.912	0.939	0.925
	$d_{4,3}$	0.896	0.900	0.898
HAT	$d_{2,1}$	0.527	0.529	0.528
	$d_{3,2}$	0.951	0.979	0.965
	$d_{4,3}$	0.960	0.965	0.963
Proposed GCVT	$d_{2,1}$	0.999	0.982	0.990
	$d_{3,2}$	0.984	0.968	0.976
	$d_{4,3}$	0.999	0.984	0.991

effectiveness and robustness of the proposed GCVT framework for multi-UAV vehicle matching.

To further assess the computational efficiency of the proposed matching framework, we measured the runtime of the Hungarian algorithm—which constitutes the main computational cost in the vehicle matching step—on each alignment edge. The results show that the average runtime is approximately 0.082 s per edge. For the eight alignment edges corresponding to the nine video nodes along the mainline expressway, the total runtime of the Hungarian algorithm is approximately 0.658 s. These results indicate that the computational burden is low under the current experimental settings.

Fig. 8 illustrates a comparison between the original trajectory and the connected trajectory. In each of Figs. 8a–8d, the trajectories in the left panel represents the result of spatial transformation alone, while the trajectories in the right panel demonstrate the connected trajectory obtained by applying the proposed GCVT approach. The color of each trajectory indicates the vehicle speed. The black double-headed arrow marks the overlapping area between the two videos. In the magnified portion of the trajectory shown in Fig. 8a, a distinct misalignment is visible between the upper and lower trajectories. The horizontal distance between these trajectories corresponds to the time offset, δ .

From the comparison of trajectories before and after connection, it is evident that the proposed method effectively

eliminates sudden changes caused by time misalignment, resulting in smoother trajectories. The colors of the trajectories in Figs. 8a–8d represent variations in speed, covering both free-flowing and congested traffic. This observation underscores the method's applicability for trajectory stitching under different traffic flow conditions. Each subplot includes lane-changing events within the video overlap area, indicating that the GCVT approach can successfully connect trajectories for vehicles that change lanes in the overlap area, without relying on car-following behavior. The approach's ability to accurately connect trajectories in such complex traffic situations highlights its robustness and adaptability to a wide range of driving behaviors.

After connecting vehicles across each pair of videos, the complete set of vehicle trajectories captured by the UAV swarm can be obtained, as shown in Fig. 9. Fig. 9 illustrates the vehicle trajectories on lane 1 of the mainline segment, covering videos v_1 – v_9 (Fig. 4), where different colors indicate vehicle speeds. The trajectories extend over 4000 m, and a fixed bottleneck can be observed around the 2000 m mark, corresponding to a merging area. Beyond expressway trajectories, the proposed GCVT method is capable of connecting vehicle trajectories recorded by irregularly distributed UAVs, enabling the reconstruction of trajectories across complex network structures involving expressways and urban intersections. Fig. 10 presents the spatial trajectories obtained by stitching videos v_4 , v_5 , v_{13} , and v_{16} . Fig. 10a shows all

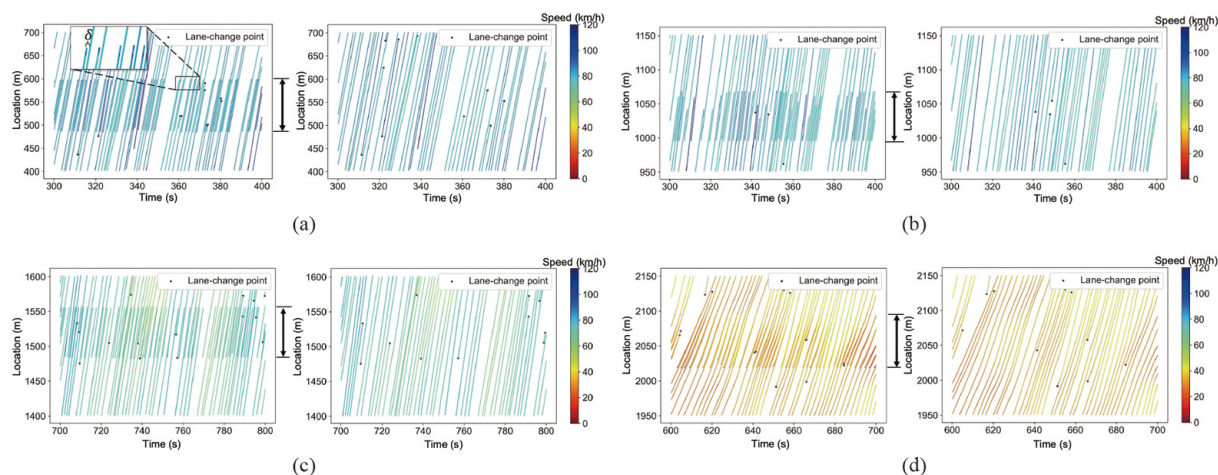


Fig. 8 Original trajectory and connected trajectory.

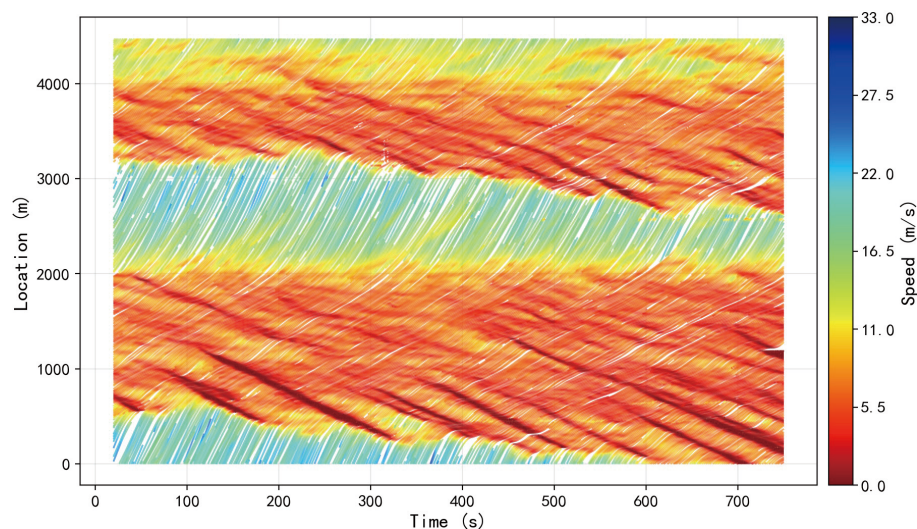


Fig. 9 Space-time diagram of vehicle trajectories after connection, for videos v_1 – v_9 .

stitched trajectories during the first three minutes, while Fig. 10b highlights three representative trajectories. The dashed boxes indicate the spatial coverage of each video, and the rectangular labels denote vehicle IDs. As illustrated, video v_4 is connected not only to the expressway segment captured in v_5 but also to two urban intersections recorded in v_{13} and v_{16} . Among the representative trajectories, Vehicle 1 travels along the expressway, whereas Vehicles 2 and 3 travel from the expressway into urban roads. These results demonstrate that the proposed method can effectively stitch trajectories captured by irregularly distributed UAVs, enabling the reconstruction of continuous vehicle trajectories across heterogeneous road types.

In real-world baseline comparisons, performance variation across alignment edges reflects overall matching difficulty, which is caused by multiple coupled factors rather than by a single variable. Therefore, Exp. 2 further used simulation experiments to examine key factors under controlled conditions. Since the time alignment analysis in Exp. 2 showed similar time alignment errors across different ground-truth time offsets under the same spatial offset, the vehicle matching analysis focused on two practical factors: Spatial transformation error and overlap length. Spatial transformation error characterizes the geometric misalignment resulting from imperfect spatial calibration, whereas overlap length denotes the length of the road segment jointly captured by two adjacent videos, which determines the amount of shared trajectory information available for matching.

Fig. 11a shows the F1-score under different spatial transformation errors, while Fig. 11b shows the F1-score under different overlap lengths. In Fig. 11a, the F1-score remains close to 1.0 when the spatial transformation error is below approximately 6 m. Since a 6 m error is roughly equivalent to the width of two traffic lanes, this result demonstrates that the proposed method remains highly robust even under considerable geometric misalignment. In Fig. 11b, the F1-score increases substantially with overlap length, rising from nearly zero at 10 m to approximately 0.64 at 15 m, reaching almost 1.0 at 20 m and above. This result indicates that a sufficient overlap length is essential for accurate and stable vehicle matching, while also demonstrating that the proposed method does not require an excessively long overlap region in practical deployment. Under the current experimental settings, an overlap length of at least 20 m is recommended for stable cross-video vehicle matching, although the required length may vary under different deployment conditions.

5.4 Ablation study

To quantify the contribution of the key modules in the proposed framework, we conducted an ablation study on vehicle matching performance using the SWIFTraj dataset. In this study, we removed or simplified one component at a time, resulting in the following four variants: (1) The proposed GCVT framework, which includes both the time alignment module and the original

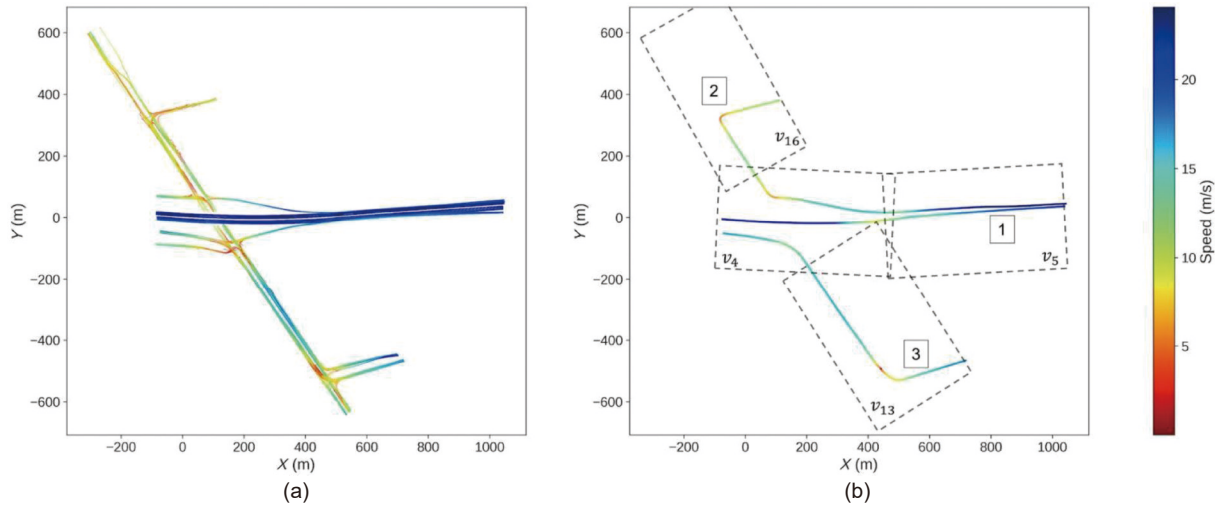


Fig. 10 Spatial trajectories illustrating connections across videos v_4 , v_5 , v_{13} , and v_{16} .

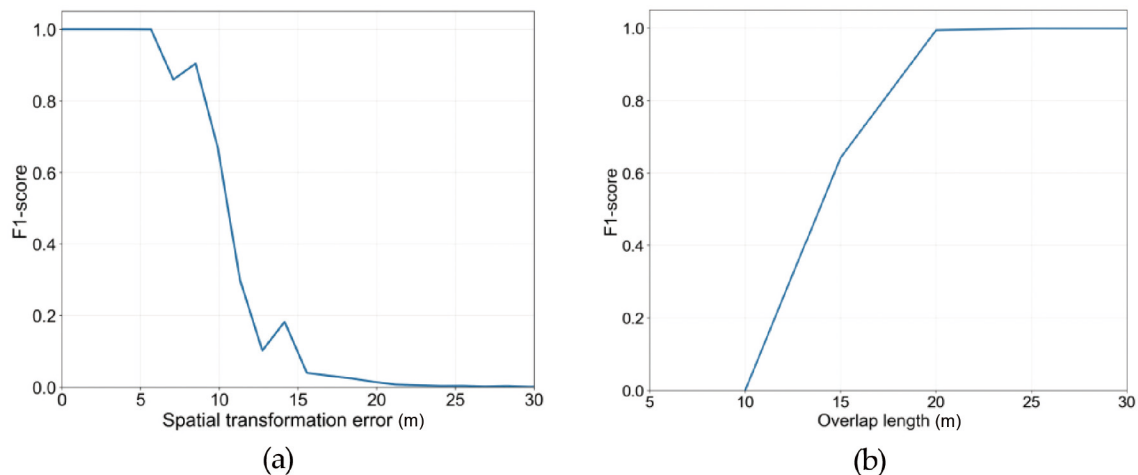


Fig. 11 Vehicle matching results in Exp. 2. (a) F1-score under different spatial transformation errors; (b) F1-score under different overlap lengths.

two-step Hungarian matching design; (2) a variant without time alignment, where vehicle matching is performed directly after spatial transformation without correcting time offsets between adjacent videos; (3) a variant without the lane-related term in the trajectory matching cost $L_{i,j}$, where the cost is computed solely based on pixel coordinates; and (4) a variant without the two-step matching strategy, where the Hungarian algorithm is applied only once, omitting the second matching step used in the proposed framework.

Table 3 presents the ablation study results. As shown, the proposed GCVT framework achieves the best overall performance, with an F1-score of 0.973. Removing the time alignment module causes the most significant degradation, reducing the F1-score to 0.890, which confirms the importance of correcting time offsets between adjacent videos. This result also suggests that relying solely on the UAV's internal clocks is insufficient for reliable vehicle matching. Additionally, removing the lane-related term from $L_{i,j}$ and replacing the original two-step Hungarian matching with a single-step strategy also leads to performance drops, demonstrating that both components contribute to the final trajectory connection accuracy. Overall, the ablation results verify the effectiveness of the main modules in the proposed framework.

Table 3 Ablation study results

Variant	Precision	Recall	F1-score
Proposed GCVT	0.976	0.971	0.973
w/o time alignment	0.880	0.901	0.890
w/o lane term in $L_{i,j}$	0.957	0.983	0.970
w/o two-step matching	0.931	0.957	0.944

6 Conclusions

To achieve automatic time alignment of multi-UAV videos and seamless connection of vehicle trajectories, this study proposes a novel graph-based approach for vehicle trajectory connection. The framework models UAV layouts as a graph, where each video corresponds to a node and overlap regions define edges, enabling flexible handling of irregular UAV deployments. A key strength of our approach lies in its ability to perform automatic time alignment by minimizing the trajectory matching cost, thereby ensuring accurate temporal synchronization of vehicle trajectories across videos.

Both real-world and simulation experiments were conducted to validate the proposed framework, with a focus on time alignment and vehicle matching. In the real-world experiment involving 16 UAVs, the absolute time alignment error was within three video frames. Quantitative evaluation based on precision, recall, and F1-score shows that the proposed method achieves reliable vehicle matching performance and outperforms representative baseline methods. The reconstructed trajectories further demonstrate the framework's capability to handle diverse traffic states, lane-changing scenarios, large spatial coverage, and irregular UAV layouts. In addition, the ablation study indicates that time alignment plays the most important role in the proposed framework, while the lane-related term in the matching cost and the two-step Hungarian matching design further improve the final trajectory connection accuracy.

Simulation experiments with varying time offsets and spatial transformation errors further reveal that time alignment remains stable under different time offsets when spatial transformation is accurate. Vehicle matching performance gradually degrades as

spatial error increases and drops sharply beyond a certain threshold. These findings confirm the robustness of the proposed approach to temporal discrepancies and its tolerance to spatial transformation errors within a reasonable range.

For future research, the proposed GCVT framework can be extended to broader multi-surveillance scenarios. Since the method operates directly on trajectory data without restricting its source, it can be applied to integrate trajectories from multiple road monitoring systems, thereby generating continuous vehicle trajectories. Several promising directions can be pursued to enhance the proposed GCVT approach. First, the current trajectory matching cost calculation for vehicle matching is based solely on pixel coordinates and lane ID. Incorporating additional vehicle attributes, such as vehicle category and appearance features, may further improve matching accuracy and robustness. Second, the current method relies on manually marking feature points to ensure the accuracy of spatial transformation. Future work may explore hybrid strategies that combine automatic feature detection with manual verification, thereby improving efficiency while maintaining spatial accuracy.

Author contributions

Xinkai Ji: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Pan Liu:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Ying Yang:** Writing – review & editing, Formal analysis. **Yu Han:** Writing – review & editing, Formal analysis, Data curation, Conceptualization.

Replication and data sharing

The proposed GCVT approach has been incorporated into our vehicle trajectory extraction framework, OpenVTER. The source code is available at <https://github.com/XinkaiJi/OpenVTER>, and the dataset can be accessed at <https://swiftraj.com>. The data and code used in this study are available at <https://doi.org/10.26599/ETSD.2026.9190022>.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 52525204 and 52232012).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Yu Han is the Replication Editor member of this journal.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT for language polishing. After utilizing this tool, the authors thoroughly reviewed and edited the content as necessary and take full responsibility for the final content of the published article.

References

- Alexiadis, V., Colyar, J., Halkias, J., Hranac, R., McHale, G., 2004. The next generation simulation program. *ITE J*, 74, 22–26.
- Barmounakis, E., Geroliminis, N., 2020. On the New Era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. *Transp Res Part C Emerg Technol*, 111, 50–71.
- Berghaus, M., Lamberty, S., Ehlers, J., Kalló, E., Oeser, M., 2024. Vehicle trajectory dataset from drone videos including off-ramp and congested traffic – Analysis of data quality, traffic flow, and accident risk. *Commun Transp Res*, 4, 100133.

- Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., Eckstein, L., 2020. The InD dataset: A drone dataset of naturalistic road user trajectories at German intersections. In: 2020 IEEE Intelligent Vehicles Symposium (IV), 1929–1934.
- Chaudhari, A. A., Treiber, M., Okhrin, O., 2025. MiTra: A drone-based trajectory data for an all-traffic-state inclusive freeway with ramps. *Sci Data*, **12**, 1174.
- Chen, D., Ahn, S., Laval, J., Zheng, Z., 2014. On the periodicity of traffic oscillations and capacity drop: The role of driver characteristics. *Transp Res Part B Methodol*, **59**, 117–136.
- Chen, P., Guan, W., Huang, F., Zhong, Y., Wen, W., Hsu, L. T., et al., 2024. ECMD: An event-centric multisensory driving dataset for SLAM. *IEEE Trans Intel Veh*, **9**, 407–416.
- Chen, Q., Wang, H., Dong, C., 2020. Modeling lane-changing behaviors in merging areas of urban expressways in Nanjing, China. *Transp Res Rec J Transp Res Board*, **2674**, 480–493.
- Chen, Q., Wang, H., Dong, C., 2022. Empirical analysis of vehicle tracking algorithms for extracting integral trajectories from consecutive videos. *PROMET-ZAGREB*, **34**, 551–565.
- Chen, X., Li, Z., Yang, Y., Qi, L., Ke, R., 2021. High-resolution vehicle trajectory extraction and denoising from aerial videos. *IEEE Trans Intel Transp Syst*, **22**, 3190–3202.
- Coifman, B., Li, L., 2024. Partial trajectory method to align and validate successive video cameras for vehicle tracking. *Transp Res Part C Emerg Technol*, **158**, 104416.
- Coifman, B., Wu, M., Redmill, K., Thornton, D. A., 2016. Collecting ambient vehicle trajectories from an instrumented probe vehicle. *Transp Res Part C Emerg Technol*, **72**, 254–271.
- De Luca, V., Pascarelli, C., Colucci, M., Afrune, P., Corallo, A., Avanzini, G., 2025. A platform for safe operations of unmanned aircraft systems in critical areas. *Engineering*, **49**, 314–331.
- Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM*, **24**, 381–395.
- Gao, R., Wang, Y., Liu, C., Wang, L., 2025. History-Aware Transformation of ReID Features for Multiple Object Tracking. <https://doi.org/10.48550/arXiv.2503.12562>
- Gloudemans, D., Wang, Y., Ji, J., Zachár, G., Barbour, W., Hall, E., et al., 2023. I-24 MOTION: An instrument for freeway traffic science. *Transp Res Part C Emerg Technol*, **155**, 104311.
- Hu, X., Zheng, Z., Chen, D., Zhang, X., Sun, J., 2022. Processing, assessing, and enhancing the Waymo autonomous vehicle open dataset for driving behavior research. *Transp Res Part C Emerg Technol*, **134**, 103490.
- Ji, X., Han, Y., Mao, P., Huang, Y., Yu, H., Liu, P., 2025. OpenVTER: An open vehicle trajectory extraction framework based on rotated bounding boxes. *IEEE Trans Intel Transp Syst*, **26**, 59–76.
- Kim, Z., Cao, M., 2010. Evaluation of feature-based vehicle trajectory extraction algorithms. In: 13th International IEEE Conference on Intelligent Transportation Systems, 99–104.
- Krajewski, R., Bock, J., Kloeker, L., Eckstein, L., 2018. The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2118–2125.
- Krajewski, R., Moers, T., Bock, J., Vater, L., Eckstein, L., 2020. The rounD dataset: A drone dataset of road user trajectories at roundabouts in Germany. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 1–6.
- Kuhn, H. W., 1955. The Hungarian method for the assignment problem. *Nav Res Logist Q*, **2**, 83–97.
- Li, Z., Zhang, X., Tian, C., Gao, X., Gong, Y., Wu, J., et al., 2023. TVG-ReID: Transformer-based vehicle-graph re-identification. *IEEE Trans Intel Veh*, **8**, 4644–4652.
- Liu, Y., Li, Q., Kim, I., 2025. Enhanced trajectory reconstruction from sparse and noisy GPS data: A progressive chunked transformer approach. *Commun Transp Res*, **5**, 100200.
- Liu, Z., He, J., Zhang, C., Yan, X., Wang, C., Qiao, B., 2023. Vehicle trajectory extraction at the exit areas of urban freeways based on a novel composite algorithms framework. *J Intel Transp Syst*, **27**, 295–313.
- Long, K., Shi, X., Li, X., 2024. Physics-informed neural network for cross-dynamics vehicle trajectory stitching. *Transp Res Part E Logist Transp Rev*, **192**, 103799.
- Ma, W., Zhong, H., Wang, L., Jiang, L., Abdel-Aty, M., 2022. MAGIC dataset: Multiple conditions unmanned aerial vehicle group-based high-fidelity comprehensive vehicle trajectory dataset. *Transp Res Rec J Transp Res Board*, **2676**, 793–805.
- Moers, T., Vater, L., Krajewski, R., Bock, J., Zlocki, A., Eckstein, L., 2022. The exiD dataset: A real-world trajectory dataset of highly interactive highway scenarios in Germany. In: 2022 IEEE Intelligent Vehicles Symposium (IV), 958–964.
- Rajput, S., Venkateshappa, S., Kanagaraj, V., Asaithambi, G., Treiber, M., 2026. SPT: Obtaining long trajectory data of disordered traffic using a swarm of unmanned aerial vehicles. *Transp Res Part C Emerg Technol*, **182**, 105431.
- Raju, N., Arkatkar, S., Easa, S., Joshi, G., 2022. Developing extended trajectory database for heterogeneous traffic like NGSIM database. *Transp Lett*, **14**, 555–564.
- Shi, X., Zhao, D., Yao, H., Li, X., Hale, D. K., Ghiasi, A., 2021. Video-based trajectory extraction with deep learning for High-Granularity Highway Simulation (HIGH-SIM). *Commun Transp Res*, **1**, 100014.
- Wang, S., Yang, D., Sheng, H., Shen, J., Zhang, Y., Ke, W., 2024a. A blockchain-enabled distributed system for trustworthy and collaborative intelligent vehicle re-identification. *IEEE Trans Intel Veh*, **9**, 3271–3282.
- Wang, Y., Gloudemans, D., Ji, J., Teoh, Z. N., Liu, L., Zachár, G., et al., 2024b. Automatic vehicle trajectory data reconstruction at scale. *Transp Res Part C Emerg Technol*, **160**, 104520.
- Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP), 3645–3649.
- Xu, Y., Shao, W., Li, J., Yang, K., Wang, W., Huang, H., et al., 2022. SIND: A drone dataset at signalized intersection in China. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), 2471–2478.
- Xu, Y., Yu, G., Wu, X., Wang, Y., Ma, Y., 2017. An enhanced viola-Jones vehicle detection method from unmanned aerial vehicles imagery. *IEEE Trans Intel Transp Syst*, **18**, 1845–1856.
- Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., et al., 2019. INTERACTION Dataset: An INTERNATIONAL, Adversarial and Cooperative motion Dataset in Interactive Driving Scenarios with Semantic Maps. <https://doi.org/10.48550/arXiv.1910.03088>
- Zhang, Y., Wang, C., Yu, R., Wang, L., Quan, W., Gao, Y., et al., 2023. The AD4CHE dataset and its application in typical congestion scenarios of traffic jam pilot systems. *IEEE Trans Intel Veh*, **8**, 3312–3323.
- Zhao, H., Wang, C., Lin, Y., Guillemard, F., Geronimi, S., Aioun, F., 2017. On-road vehicle trajectory collection and scene-based lane change analysis: Part I. *IEEE Trans Intel Transp Syst*, **18**, 192–205.
- Zhen, L., Yang, Z., Laporte, G., Yi, W., Fan, T., 2024. Unmanned aerial vehicle inspection routing and scheduling for engineering management. *Engineering*, **36**, 223–239.
- Zheng, O., Abdel-Aty, M., Yue, L., Abdelraouf, A., Wang, Z., Mahmoud, N., 2024. CitySim: A drone-based vehicle trajectory dataset for safety-oriented research and digital twins. *Transp Res Rec J Transp Res Board*, **2678**, 606–621.



Xinkai Ji received the M.S. degree in transportation engineering from Southeast University, Nanjing, China, in 2019. He worked as a Traffic Engineer at Zhejiang Lab from 2020 to 2021. He is currently pursuing the Ph.D. degree in transportation engineering at the School of Transportation, Southeast University, Nanjing, China. His research interests include connected and automated vehicles and traffic flow.



Pan Liu received the Ph.D. degree in civil engineering from the University of South Florida, Tampa, USA, in 2006. He is the Professor with the School of Transportation, Southeast University, Nanjing, China. His research interests include traffic operations and safety, and intelligent transportation systems. He was supported by the National Natural Science Foundation of China (NSFC) Excellent Young Scientist Fund in 2013 and the National Science Fund for Distinguished Young Scholars of NSFC in 2019.



Ying Yang received the Ph.D. degree in psychology from Beijing Normal University, China, in 2012. She is currently a Professor at the School of Management, Shanghai University, China, and an Affiliated Professor at Chalmers University of Technology, Sweden. Her recent research has primarily focused on the management of sustainable transport behavior. She applies behavioral economics theory, sociocultural models, and decision-making theories to better understand the social, psychological, and economic factors that contribute to the adoption of intelligent and sustainable transport modes (e.g., on-demand buses, electronic vehicles, bicycle-sharing, and automated vehicles). She was awarded an Endeavour Research Fellowship funded by the Australian Department of Education.



Yu Han received the Ph.D. degree in traffic operation and management from Delft University of Technology, the Netherlands, in 2017. He is currently an Associate Professor in the Thrust of Intelligent Transportation at The Hong Kong University of Science and Technology (Guangzhou), China. Prior to this, he was an Associate Professor at the School of Transportation, Southeast University, Nanjing, China. His research focuses on developing traffic management and control strategies across multiple modes and scales through interdisciplinary approaches, including traffic flow theory, control theory, optimization, and artificial intelligence. He serves as an Associate Editor for *IEEE Transactions on Intelligent Transportation Systems* and as a Replication Editor for *Communications in Transportation Research*.