Transportation Research Part C 178 (2025) 105205

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Advanced computer vision for extracting georeferenced vehicle trajectories from drone imagery

Robert Fonod [a,*], Haechan Cho [b], Hwasoo Yeo [b], Nikolas Geroliminis [a]

[a] School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, CH-1015, Switzerland

[b] Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, 34141, South Korea

ARTICLE INFO

Dataset link: Songdo Traffic, Songdo Vision

Keywords:
Traffic monitoring
Machine vision
Vehicle tracking
Video image processing
Georeferenced vehicle trajectories
Multi-drone data collection
Urban traffic

ABSTRACT

This paper presents a comprehensive framework for extracting georeferenced vehicle trajectories from high-altitude drone imagery, addressing key challenges in urban traffic monitoring and the limitations of traditional ground-based systems. Our approach integrates several novel contributions, including a tailored object detector optimized for high-altitude bird's-eye view perspectives, a unique track stabilization method that uses detected vehicle bounding boxes as exclusion masks during image registration, and an orthophoto and master frame-based georeferencing strategy that enhances consistent alignment across multiple drone viewpoints. Additionally, our framework features robust vehicle dimension estimation and detailed road segmentation, enabling comprehensive traffic dynamics analysis. Conducted in the Songdo International Business District, South Korea, the study utilized a multi-drone experiment covering 20 intersections, capturing approximately 12TB of ultra-high-definition video data over four days. The framework produced two high-quality datasets: the Songdo Traffic dataset, comprising approximately 700,000 unique vehicle trajectories, and the Songdo Vision dataset, containing over 5000 human-annotated images with about 300,000 vehicle instances categorized into four classes. Comparisons with high-precision sensor data from an instrumented probe vehicle highlight the accuracy and consistency of our extraction pipeline in dense urban environments. The public release of the Songdo Traffic and Songdo Vision datasets, along with the complete source code for the extraction pipeline, establishes new benchmarks in data quality, reproducibility, and scalability in traffic research. The results demonstrate the potential of integrating drone technology with advanced computer vision methods for precise and cost-effective urban traffic monitoring, providing valuable resources for developing intelligent transportation systems and enhancing traffic management strategies.

1. Introduction

Traditional traffic monitoring methods, such as loop detectors and manual counting, are becoming obsolete in the era of smart cities due to their inflexibility and limited scope (Zhang et al., 2018). Stationary camera surveillance, although effective, presents economic challenges due to high installation costs and limited field of view (FoV), leading to blind spots. Data from global navigation satellite system (GNSS) and connected vehicles also lack the versatility needed for comprehensive traffic monitoring. Several vehicle

* Corresponding author.
E-mail addresses: robert.fonod@ieee.org (R. Fonod), gkqkemwh@kaist.ac.kr (H. Cho), hwasoo@kaist.ac.kr (H. Yeo), nikolas.geroliminis@epfl.ch (N. Geroliminis).

https://doi.org/10.1016/j.trc.2025.105205
Received 4 November 2024; Received in revised form 17 March 2025; Accepted 23 May 2025
Available online 2 July 2025
0968-090X/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

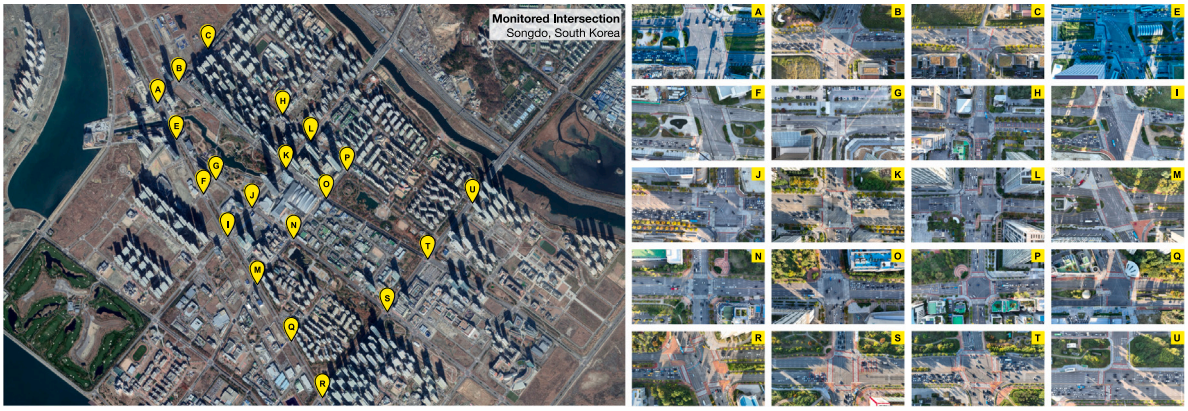


Fig. 1. Left: Map of the Songdo International Business District, indicating the locations of the 20 monitored intersections labeled A to U (excluding D, which denotes the drones). Right: Images of the 20 intersections as seen by our DJI Mavic 3 drones at 4K resolution.

detection and tracking technologies have emerged, including camera-based (Sivaraman and Trivedi, 2013; Yang and Pun-Cheng, 2018; Ghahremannezhad et al., 2023), LiDAR-based (Li, 2017; Zhang et al., 2019), magnetics-based (Sifuentes et al., 2011; Wang et al., 2017), and radar-based (Fang et al., 2007; Park et al., 2003) systems. Although each offers certain advantages, they still face limitations in adaptability and coverage. Accurate estimation of traffic states allows adjustments in traffic control strategies (Seo et al., 2017), improving network utilization (Kouvelas et al., 2023; Tsitsokas et al., 2023) and road user experiences, as well as delivering economic and ecological benefits.

The integration of unmanned aerial vehicles (UAVs), commonly known as drones, and computer vision (CV) techniques presents a promising solution to these challenges. Originally developed for military purposes (Hassanalian and Abdelkefi, 2017), UAVs have since found applications in various civilian domains, including urban traffic monitoring and management (Khan et al., 2017; Outay et al., 2020; Espadaler-Clapés et al., 2023b). Advances in drone technology, combined with sophisticated CV tools, enable the extraction of detailed vehicle trajectories from high-altitude drone footage, facilitating spatio-temporal analyses of dynamic entities such as vehicles, pedestrians, and cyclists (Gohari et al., 2022; Kim, 2020; Mahajan et al., 2023). Compared to traditional sensors, these trajectories offer new opportunities to study complex traffic phenomena, such as large-scale network modeling (Paipuri et al., 2021), on-street parking (Kim et al., 2024), as well as emission (Barmpounakis et al., 2021) and noise (Espadaler-Clapés et al., 2023a) estimation. This aerial perspective, free from the constraints of fixed infrastructure, provides an adaptable and cost-effective means of observing urban mobility patterns in congested environments, thus supporting responsive traffic management strategies in smart cities (Jian et al., 2019; Khan et al., 2020; Barmpounakis and Geroliminis, 2020).

In a collaborative effort between KAIST and EPFL, a unique multi-drone experiment was conducted in the Songdo International Business District, South Korea, from October 4 to 7, 2022. The first planned day, October 3, was excluded due to severe weather conditions, including strong winds exceeding the drones' operational limits, rain, and fog, which disrupted flight operations. This adjustment ensured stable drone performance and consistent data quality. A fleet of 10 commercial-off-the-shelf drones, operated by certified pilots, synchronized their operations to monitor 20 busy intersections, as depicted in Fig. 1. Advanced flight plans, which combine the hover and transition phases, were executed to optimize coverage. The drones maintained an altitude of 150 m, some reduced to 140 m to mitigate collision risks, ensuring compliance with South Korean UAV regulations. Adopting a bird's-eye view (BEV) perspective, the drones captured footage at 4K ultra-high-definition (UHD) resolution (3840×2160 pixels) at a smooth frame rate of 29.97 frames per second (FPS). Data collection was structured into 10 daily flight sessions, with five morning sessions (AM1–AM5) and five afternoon sessions (PM1–PM5), capturing traffic flow during peak morning and afternoon hours. Each synchronized flight session lasted approximately 30 min, followed by a brief battery replacement, bringing the total cycle time to exactly 40 min. Over the course of the experiment, 400 flights were conducted, generating 12TB of raw video data. Thirteen real-time kinematic (RTK)-calibrated GCPs were strategically placed throughout the study area as reference points for a dedicated drone, which captured overlapping images at an altitude of 75 m. These images produced a high-resolution orthophoto that serves as a geometrically accurate reference for georeferencing the extracted trajectories and mapping.

This work introduces a comprehensive end-to-end trajectory extraction pipeline that employs advanced CV algorithms to derive detailed traffic data from high-altitude drone footage. Although a detailed experiment design, regulatory considerations, and an in-depth analysis of the extracted data are outside this paper's scope, our contributions significantly expand the resources available to the research community. Specifically, we introduce the *Songdo Traffic* dataset (Fonod et al., 2025a), comprising approximately 700,000 unique vehicle trajectories, making it one of the largest public traffic datasets collected from a smart city worldwide. This extensive collection provides unparalleled insights into urban mobility, serving as a valuable resource for traffic management and research. The dataset includes trajectory data at a high frequency of 29.97 points per second, complemented by various metadata. Songdo Traffic encompasses vehicle trajectories in multiple CSs, including global geographic, local Cartesian, and orthophoto CS. Each trajectory is further annotated with metadata such as vehicle identifier (ID), type, and, when feasible, vehicle dimension

estimates. Additionally, each trajectory point is tagged with a precise ISO-formatted timestamp, along with instantaneous speed and acceleration estimates, road section and lane number information, and a visibility flag, where available.

Complementing Songdo Traffic, the *Songdo Vision* dataset (Fonod et al., 2025b) addresses the scarcity of detailed vehicle annotations necessary for training object detection models, particularly from high-altitude BEV aerial perspectives, where smaller vehicles, like motorcycles, are often less discernible. This dataset contains over 5000 human-annotated video frames, featuring nearly 300,000 vehicle instances across four distinct classes: cars (including vans and light-duty vehicles), buses, trucks, and motorcycles. To facilitate broader use, annotations are provided in multiple formats compatible with various deep learning (DL) models, making the Songdo Vision dataset a valuable resource for improving and validating vehicle detection algorithms. In line with our commitment to open science, we have publicly released these datasets (Fonod et al., 2025a,b) alongside the source codes (Fonod, 2025a,b,c) for our extraction and stabilization routines. These resources are expected to set new benchmarks for data quality and reproducibility in traffic research leveraging CV and DL techniques.

Our extraction methodology introduces several innovative techniques that significantly enhance the accuracy and reliability of vehicle trajectory data from drone imagery. We developed a tailored YOLOv8-based object detector optimized for detecting small vehicles at high-altitude BEV perspectives. To address the challenges of drone movements, we proposed a novel track stabilization approach that applies homographic transformations directly to extracted tracks rather than raw video frames, allowing for the effective use of exclusion masks based on detected vehicle bounding boxes. This approach reduces both storage and computational overhead compared to full-frame video stabilization and is further optimized through a newly proposed metric designed to evaluate object-level stabilization accuracy. Additionally, we introduced a robust vehicle dimension estimation method that uses azimuth-based filtering and ratio-based criteria to ensure accurate measurements under varying conditions. Our methodology further includes speed and acceleration computations, incorporating visibility filtering and Gaussian smoothing to derive realistic traffic dynamics. To enhance consistency and minimize matching errors during georeferencing, we propose a novel master frame approach that serves as an intermediary CS, ensuring uniform transformation accuracy across multiple drone viewpoints. Advanced georeferencing techniques, employing a high-resolution orthophoto enhanced with 13 ground control points, further ensure precise mapping of trajectories into real-world coordinates. We optimized the stabilization routine and image registration through an extensive parameter search, resulting in highly accurate geospatial data for traffic analysis.

During the Songdo experiment, various traffic events, including severe congestion, excessive speeds, and post-accident traffic dynamics, were captured, along with data from an instrumented AV provided by Stanford Center at the Incheon Global Campus (SCIGC), which was equipped with high-precision RTK-GNSS sensors. Although the collision itself occurred during a battery swap and was not recorded, the prolonged presence of the involved vehicles enabled us to document the resulting traffic disruptions over time. The AV data provided a valuable resource for comparison with our drone-derived trajectories and speed profiles, demonstrating very high consistency between the two measurement methods. Additionally, our methodology was subjected to rigorous hyperparameter tuning for track stabilization and comprehensive benchmarks to evaluate image registration and georeferencing accuracy. For vehicle dimension estimation, the results showed close agreement with the known dimensions of specific vehicles, such as the AV and certain bus types, confirming the method's ability to produce realistic distributions.

By addressing key challenges in drone-based traffic monitoring through CV, as discussed in Section 2, this research presents a comprehensive end-to-end framework for extracting georeferenced trajectories from high-altitude drone footage. We enhance detection and tracking algorithms to handle unique perspectives and dynamic conditions, surpassing traditional ground-based methods. Our innovative track stabilization, accurate vehicle dimension estimation, speed and acceleration computations, and precise georeferencing contribute to the extraction of high-quality trajectory data. Comparison with AV data shows that our methods produce realistic speed, acceleration, and dimension distributions. By disseminating these methods through open-source datasets and code, we establish a new resource for data quality, facilitating more responsive, scalable, and cost-effective traffic management solutions in smart cities.

2. Related work

This section reviews the state-of-the-art (SOTA) in drone-based traffic data extraction, including object detection, tracking, video stabilization, georeferencing, and dataset availability. Traditional CV methods for fixed cameras, such as background subtraction (Mandellos et al., 2011), optical flow (OF) (Gokul et al., 2023), template matching (Choi et al., 2006), feature matching (Coifman et al., 1998), and histogram-based methods (Cao et al., 2011; Wu et al., 2013), face challenges in complex environments (Khan et al., 2018) and under varying illumination conditions (Kim and Jung, 2017; Zhang and Jin, 2019; Ghahremannezhad et al., 2023).

2.1. Object detection

The rise of DL techniques for CV has significantly improved vehicle detection. Leading architectures include single shot detector (SSD) (Liu et al., 2016), region-based convolutional neural network (R-CNN) (Girshick et al., 2014), and you only look once (YOLO) (Redmon et al., 2016). R-CNN variants, such as Faster R-CNN (Ren et al., 2017), have been used for vehicle detection in various studies (Hsu et al., 2018; Zhan et al., 2019). YOLO models (v3–v10) (Redmon and Farhadi, 2018; Bochkovskiy et al., 2020; Jocher, 2020; Li et al., 2022; Wang et al., 2023; Jocher et al., 2023; Wang et al., 2024,a) have also been adapted for accuracy in traffic surveillance (Mao et al., 2020; Zhang et al., 2022a; Espadaler-Clapés et al., 2023b).

Despite their effectiveness, these models often rely on non-maximum suppression (NMS), a post-processing technique that removes redundant bounding boxes (BBs) by retaining only the one with the highest confidence score for each detected object.

Recent transformer-based detectors (DETRs)(Carion et al., 2020; Zhu et al., 2021a; Liu et al., 2021; Zhao et al., 2024) aim to eliminate NMS but struggle with small object detection and require high computational resources. Specialized detectors, such as YOLO for occluded vehicles (He et al., 2024) and the Butterfly detector (Adaimi et al., 2023), address challenges specific to aerial perspectives. To improve small vehicle detection in drone footage, Novikov et al. (2024) incorporated the computationally expensive Sliced Aided Hyper Inference (SAHI) technique (Akyon et al., 2022) into their YOLOv8-based detector, enabling enhanced detection capabilities in high-resolution aerial imagery. Integration of multimodal data, like combining RGB with LiDAR (Wu et al., 2024) or thermal (Ha et al., 2017) imaging, has also shown promise in enhancing detection in complex urban environments.

2.2. Object tracking

Accurate tracking is essential for reliable movement data. Traditional methods (Yilmaz et al., 2006), like Kalman filtering (Kalman, 1960) and the Hungarian algorithm (Kuhn, 1955), have been widely used in traffic analysis (Hsieh et al., 2006; Kim and Cao, 2010; Zhan et al., 2019; Chen et al., 2020). However, these approaches often struggle with drone-captured footage complexities, especially in dynamic urban environments where occlusions, varying object sizes, and diverse motion patterns pose significant challenges.

Current challenges in aerial tracking include persistent track loss in occluded areas, object similarity, high computational costs, and the additional complexity of camera-induced motion. These issues are particularly pronounced in dense urban environments. To address the challenge of camera-induced motion, ego motion compensation (EMC) emerges as an effective solution by isolating object motion, thereby enhancing tracking accuracy in dynamic drone-captured scenarios. While benchmarks like MOTChallenge (Milan et al., 2016; Dendorfer et al., 2020, 2021) have advanced pedestrian tracking, vehicle tracking in BEV contexts remains underrepresented.

Newer multi-object tracking (MOT) algorithms, such as simple online and realtime tracking (SORT) (Bewley et al., 2016), observation-centric SORT (OC-SORT) (Cao et al., 2023), ByteTrack (Zhang et al., 2022b), and "bag of tricks" SORT (BoT-SORT) (Aharon et al., 2022), follow the "separate detection and tracking" paradigm. Hybrid models, including DeepSORT (Wojke et al., 2017), StrongSORT (Du et al., 2023), and BoT-SORT-ReID (Aharon et al., 2022), incorporate appearance information to reduce identity switches and handle occlusions more effectively. End-to-end tracking approaches like FairMOT (Zhang et al., 2021) combine detection and tracking but face challenges in BEV perspectives due to limited ground truth data and high computational demands. Moreover, FairMOT's joint detection-ReID architecture requires extensive domain-specific fine-tuning, which is impractical given the lack of annotated aerial tracking data.

2.3. Stabilization

Unlike larger drones equipped with high-performance gimbals, video stabilization is crucial for accurate traffic data extraction with smaller commercial drones that are more susceptible to environmental disturbances like wind (Mingkhwan and Khawsuk, 2017; Barmpounakis et al., 2019). Stabilization, a form of video registration (Kumar et al., 1998), involves aligning consecutive frames by matching keypoints to correct unwanted motion. Various keypoint detectors such as scale-invariant feature transform (SIFT) (Lowe, 2004), root SIFT (RSIFT) (Arandjelović and Zisserman, 2012), binary robust invariant scalable keypoints (BRISK) (Leutenegger et al., 2011), KAZE (Alcantarilla et al., 2012), accelerated-KAZE (A-KAZE) (Alcantarilla and Solutions, 2013), and oriented FAST and rotated BRIEF (ORB) (Rublee et al., 2011) are commonly used. OF methods, like the Kanade-Lucas-Tomasi (KLT) tracker (Shi et al., 1994), are faster but can be less reliable (Ke et al., 2016). Recent DL-based methods such as SuperPoint (DeTone et al., 2018) and SuperGlue (Sarlin et al., 2020) enhance keypoint detection and matching, although they often require fine-tuning with ground truth data.

Keyframe correspondences established through these features allow for the computation of either perspective or affine transformations using the least squares method, optimized by variants of the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981) for outlier rejection. Feature matching algorithms, such as brute force (BF) or fast library for approximate nearest neighbors (FLANN) (Muja and Lowe, 2009), compute similarity scores between extracted keypoints to form correspondences. Preprocessing steps, like grayscale conversion and contrast limited adaptive histogram equalization (CLAHE), are implemented to enhance keypoint detection in varying lighting conditions.

Various methods can be combined to align video frames, allowing for smoother vehicle tracking. For instance, (Mingkhwan and Khawsuk, 2017) utilized speeded-up robust features (SURF) features for keypoint detection to ensure smooth traffic analysis. However, the main challenge in stabilization is achieving an optimal balance between speed and accuracy while mitigating distortions caused by moving objects.

2.4. Georeferencing

Georeferencing, in the context of our study, is the process of transforming vehicle trajectories from image coordinates into real-world geographic coordinates (e.g., latitude and longitude). This process is essential for integrating trajectory data with other geospatial datasets, enabling accurate traffic analysis and infrastructure assessment. Accurate georeferencing often requires a precise georeferenced image of the area (Kumar et al., 1998, 2001; Barber et al., 2006). An orthophoto is a geometrically corrected (orthorectified) image that eliminates distortions caused by camera tilt, lens distortion, and topographic relief, producing a uniform scale throughout the image. Unlike standard aerial photographs, orthophotos allow for precise measurements of distances, angles,

and areas directly on the image, making them indispensable for traffic monitoring and urban planning (Toutin, 2004). Nonetheless, previous studies often omit meticulous consideration of accurate georeferencing, relying on simplistic methods based on ground sample distance (GSD) for estimating only a subset of traffic parameters, such as vehicle speed and location derived from drone's location and GSD-scaled image coordinates. However, these methods lack the accuracy required for traffic monitoring, particularly since commercial-off-the-shelf drones often exhibit positional errors greater than claimed (Ekaso et al., 2020).

More sophisticated georeferencing involves the use of geographic information system (GIS) software (Espadaler-Clapés et al., 2023b) for manual alignment of imagery, which, while more accurate than basic GSD methods, is time-consuming and demands significant expertise. Alternatively, RTK-GNSS can significantly enhance accuracy by refining both the GCPs placement and the drone's position during image capture. By providing centimeter-level precision through corrections from a nearby base station, RTK-GNSS facilitates the generation of highly accurate orthophotos for the surveyed area. Moreover, integrating these orthophotos with automatic image matching not only reduces the need for manual interventions but also addresses the limitations of GSD and manual GIS-based methods, enabling more reliable mapping of vehicle movements in real-world coordinates.

2.5. Availability of datasets

Recent datasets such as HighD (Krajewski et al., 2018), HARPY (Makrigiorgis et al., 2020), pNEUMA (Barmpounakis and Geroliminis, 2020), HIGH-SIM (Shi et al., 2021), and CitySim (Zheng et al., 2024) have primarily targeted specific traffic scenarios, such as single intersections or highway segments, often relying on relatively outdated CV techniques. For example, HighD, HARPY, and HIGH-SIM employed U-Net, YOLOv2, and YOLOv3 for detection, respectively, but lacked robust mechanisms for trajectory validation. The pNEUMA dataset, generated by static drone swarms, provides insights into traffic flow but exhibits imprecise BBs and trajectory anomalies (Kim et al., 2023; Mahajan et al., 2023), potentially due to visual limitations or suboptimal CV tools. Meanwhile, CitySim has not fully disclosed its georeferencing methods, and the validation of its traffic data remains uncertain.

In contrast, infrastructure-based approaches such as the I-24 MOTION (Gloudemans et al., 2023), while addressing similar challenges, capture comprehensive traffic dynamics over extended freeway segments using a dense network of 276 high-resolution, pole-mounted cameras spanning approximately 4.2 miles. The system utilizes a curvilinear coordinate system for accurate lane-level positioning and an automated post-processing pipeline to correct detection artifacts such as dropped frames, occlusions, and trajectory fragmentation (Gloudemans et al., 2023). More recently, efforts have been made to enhance the accuracy of I-24 MOTION trajectories by employing automated data reconciliation techniques (Wang et al., 2024b), leveraging vehicle dynamics and physical constraints to associate fragmented tracks, remove outliers, and impute missing data. However, such systems are inherently limited in adaptability and spatial coverage compared to UAV-based systems, highlighting the distinctive advantage of UAV flexibility in capturing detailed, dynamic urban environments.

Vehicle annotation datasets, such as CARPK (Hsieh et al., 2017), CyCAR (Kouris et al., 2019), HARPY (Makrigiorgis et al., 2022), RAI4VD (Puertas et al., 2022), UAVDT (Yu et al., 2020), UIT-ADrone (Tran et al., 2023), and VisDrone (Zhu et al., 2021), serve as valuable training resources for object detection models. Nonetheless, they frequently utilize basic annotation methods, cover a restricted or incoherent range of vehicle classes, and exhibit varying degrees of annotation accuracy, thereby potentially affecting the reliability and consistency of DL models trained on these datasets. To mitigate these limitations, we incorporated these datasets in conjunction with our custom Songdo Vision dataset (Fonod et al., 2025b) and employed a multi-phase object detector training pipeline, which is described in Section 3.3, where Table 1 presents the relevant metadata for these datasets.

The datasets provided in this study, Songdo Traffic and Songdo Vision, are intended to significantly expand the scope and quality of existing traffic datasets by offering high-frequency georeferenced trajectories and extensive annotated data, thereby supporting advanced research in traffic monitoring and computer vision. For readers seeking detailed comparisons of existing datasets, we recommend referring to recent reviews such as (Gloudemans et al., 2023; Butilă and Boboc, 2022; Wang et al., 2024b) for traffic datasets, and Ding et al. (2021), Abdel-Aty et al. (2023), Leng et al. (2024) for vehicle annotation-oriented datasets. These works provide extensive analyses, highlight existing limitations, and discuss future directions for dataset collection and methodological improvements.

2.6. Summary

In summary, while the potential of drones and CV for urban traffic data extraction is evident, challenges such as varying object scales, occlusions, and the scarsity of comprehensive datasets persist (Bouguettaya et al., 2021; Ghahremannezhad et al., 2023). Most existing approaches are optimized for static cameras and lower altitudes, restricting their effectiveness in BEV perspectives, where objects appear smaller and less distinctive. Video stabilization and georeferencing are critical for reliable trajectory extraction from high-altitude drone imagery. Common methods, including GSD-based and manual GIS georeferencing, often lack accuracy, while the incorporation of RTK-GNSS with GCPs and advanced stabilization techniques shows promise in mitigating these limitations. Expanding the scope and quality of available datasets, as demonstrated in this study, can further support advanced traffic monitoring and urban planning.

3. Methodology

Our methodology for transforming raw drone footage into detailed traffic data, which is compiled into the Songdo Traffic dataset, involves several key steps, as shown in Fig. 2. First, in the data wrangling step, the raw videos are processed into meaningful

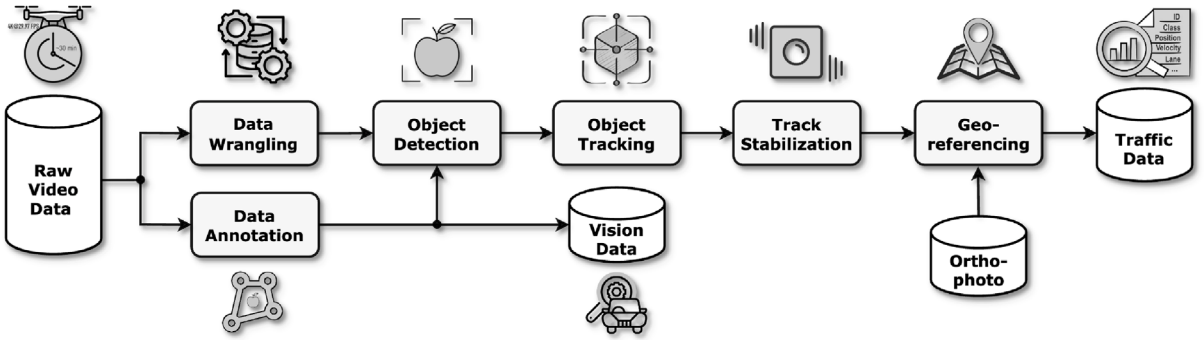


Fig. 2. Diagram illustrating the vehicle trajectory extraction process from raw drone footage to traffic data, with auxiliary human annotation and vision data generation.

segments. In parallel, selected frames are annotated to identify vehicle classes, which are used to train and validate an object detection algorithm, and are released as part of the Songdo Vision dataset. The detection algorithm processes the segmented videos to detect and locate vehicles. Next, a MOT algorithm assigns unique IDs and tracks vehicles across frames. Image registration methods stabilize the footage and transform vehicle coordinates into a fixed reference frame. Georeferencing then converts these coordinates into real-world values using homographic and affine transformations, along with an orthophoto. Finally, the extracted trajectories are used to derive useful metadata, including vehicle dimensions, speed, acceleration, road segment, and lane number. By applying track stabilization after object tracking, we are able to bypass the limitations of traditional video stabilization and achieve more accurate results.

3.1. Data wrangling

During the data wrangling (pre-processing) phase, we segmented 12TB of raw UHD drone videos, saved in .mp4 format at 29.97 FPS, along with their corresponding flight logs, into distinct segments. Each video segment corresponds to a stable hovering period of an individual drone over a monitored intersection (see Fig. 1). In addition to manual data wrangling, automated post-processing checks were performed to ensure data consistency and correct any remaining video or flight-log anomalies, as detailed in Appendix C.

Due to the drone's built-in file size limitation of 4 GB per video file, the fragmented raw videos and corresponding flight logs were first merged into a single file per drone per flight session, resulting in approximately 30 min segments on average. These merged files contained stationary footage at intersections and dynamic clips of drones transitioning between intersections or during take-offs and landings. Due to the lack of yaw measurements in the flight logs, manual segmentation by human annotators was necessary to identify the start and end frames of stable hoverings above intersections, as well as to identify any unintentional movements caused by technical glitches or human error, including significant drone movements during hoverings, recording issues, camera tilt, or zoom. This ensured that only stable and high-quality footage was retained.

After manual annotation of the intended cut locations, an automated process adjusted them to the nearest I-frame (intra-coded frame, a key reference point in video compression) to avoid re-encoding, significantly speeding up the editing process. The resulting video segments were then organized using an automated naming system, where each filename was generated based on the intersection label (see Fig. 1), determined by matching the average latitude and longitude from the flight logs with the nearest intersection center, and incorporated a unique sequence number appended to each hovering event, with the sequence resetting at the start of each flight session. Due to the optimized flight paths, video lengths varied, typically ranging from 2 to 4 min, depending on the drone and intersection.

3.2. Data annotation

The unique challenges of the Songdo study, particularly the BEV aerial perspective at 140–150 m altitudes, necessitated the creation of a specialized vehicle annotation dataset: *Songdo Vision* (Fonod et al., 2025b). Existing public datasets often exhibit varying perspectives, resolutions, and object sizes incompatible with our study's specific conditions. Additionally, annotations in these datasets are frequently inaccurate, containing poor-quality BBs that compromise detection performance. To overcome these limitations, we constructed a tailored dataset categorizing vehicles into four standardized classes: cars (including vans and light-duty vehicles), buses, trucks, and motorcycles. Annotation classes were consolidated to ensure consistency with public datasets used during training. Specifically, trucks were annotated as a single class without subdivision into subclasses (e.g., small vs. heavy trucks), while vans and light-duty vehicles, variously classified in external datasets, were uniformly grouped under the car class. This standardization prevented extensive re-annotation, which was beyond the scope of this study.

From the raw drone footage, we selected a representative subset of 5,419 video frames. Of these, 5,274 frames were randomly sampled from merged videos (described in Section 3.1), covering stationary hovering and transitioning sequences. The remaining 145 frames were purposefully chosen to capture underrepresented yet critical scenarios, such as motorcycles at pedestrian crossings,



Fig. 3. Left: Examples of human-annotated frames. Right: Examples of underrepresented motorcycle instances.

bicycle lanes, traffic signals, and distinctive road markers, as illustrated in Fig. 3 (right). Frames captured at unsuitable altitudes (e.g., drone take-off or landing sequences) or unfavorable camera angles were excluded, maintaining dataset integrity.

Annotations were conducted by trained annotators using Microsoft Azure ML Studio, requiring approximately 300 cumulative hours. Initially, bounding boxes were annotated manually. However, as annotations progressed, the platform's automated labeling functionality was gradually activated, increasingly proposing candidate bounding boxes, thus accelerating the annotation process. To ensure high annotation accuracy and consistency, each frame underwent three iterative rounds of careful validation and refinement. Overall, nearly 300,000 vehicle instances were captured with tight, axis-aligned BBs, as illustrated in Fig. 3 (left) and summarized in Table 2.

To facilitate reliable benchmarking, we established an 80 and 20 percent split of the Songdo Vision dataset for training and testing, respectively, ensuring an unbiased distribution of vehicle instances across all classes. By preserving this division in our public release, we enable other researchers to directly compare their custom-trained detectors against our results using the same test set. The detailed breakdown of this split is provided in Table 2, showing an almost perfect 80 to 20 percent ratio across all annotated class labels. The Songdo Vision dataset, enriched with human-verified annotations, is designed to significantly improve vehicle detection algorithms by providing a reliable ground truth for model validation. The annotations are released in widely used formats such as COCO, YOLO, and Pascal VOC, ensuring compatibility with a broad range of applications beyond traffic analysis, including autonomous drone delivery, surveillance, safety systems, and environmental studies.

3.3. Object detection

Object detection is a pivotal component of our vehicle extraction framework. We employ a DL-based object detector that predicts BBs encoded as $(x,\ y,\ w,\ h)$. Here, $(x,\ y)$ represent the center coordinates of the BB, while w and h denote its width and height, respectively. Each kth image frame of video v, denoted F_v^k, is a $w_I \times h_I$ matrix of pixel intensities, where w_I and h_I represent the frame's width and height in pixels, respectively.

The object detector processes each frame F_v^k to identify potential vehicles by predicting BBs. Following this initial detection step, our pipeline employs a class-agnostic NMS algorithm with an intersection over union (IoU) threshold $\eta_{\text{IoU}} > 0$. This crucial step effectively eliminates overlaps among the BBs, thereby refining the overall detection quality in densely populated urban settings. The refined set of detections, $\mathcal{D}_v[k]$, includes only the most reliable vehicle detections and is defined as follows:

$$\mathcal{D}_v[k] = \left\{ \left(x^{(i)}[k],\ y^{(i)}[k],\ w^{(i)}[k],\ h^{(i)}[k],\ c^{(i)}[k],\ s^{(i)}[k]\right) \;\middle|\; s^{(i)}[k] \geq \tau_{\text{d}} \right\}_{i=1}^{N_v^d[k]}, \tag{1}$$

where $N_v^d[k]$ represents the number of vehicle detections in the kth frame after processing through NMS. For the ith detection, $x^{(i)}$ and $y^{(i)}$ are the normalized center coordinates of the BB, $w^{(i)}$ and $h^{(i)}$ are its normalized width and height, $c^{(i)}$ is the class label restricted to $c^{(i)} \in C = \{0, 1, 2, \ldots, C-1\}$, with $C > 0$ representing the total number of distinct object classes recognized by the detector, and $s^{(i)}$ is the confidence score constrained within $(0, 1]$. Detections are retained in $\mathcal{D}_v[k]$ only if their confidence scores exceed the detection threshold $\tau_{\text{d}} > 0$. All BB coordinates and dimensions are normalized with respect to the input image size, ensuring that $0 \leq x^{(i)},\ y^{(i)},\ w^{(i)},\ h^{(i)} \leq 1$.

For our study, we employed the anchor-free YOLOv8 detector architecture (Jocher et al., 2023). Initially, we explored anchor-based YOLOv5 (Jocher, 2020) and transformer-based real-time DETR (RT-DETR) (Zhao et al., 2024) as alternatives, but found that YOLOv8 offered superior accuracy and faster inference speeds, particularly for smaller objects, making it more suitable for our operational requirements. Our choice of YOLOv8 was also influenced by the practical challenges associated with transformer-based architectures like DETR or RT-DETR, which require extensive computational resources for training and inference due to their reliance on self-attention mechanisms. In addition, achieving satisfactory detection performance for smaller vehicles, such as motorcycles, necessitated increasing the input resolution beyond the default settings, which would further amplify DETR's

Table 1
Vision datasets used in multi-stage object detector training.

Dataset	BASE	FINE	Consolidated class labels	Present but unlabeled
Songdo Vision (ours) (Fonod et al., 2025b)	✓	✓	Car, bus, truck, motorcycle	Pedestrian, bicycle
CARPK (Hsieh et al., 2017)	✓	✗	Car	Motorcycle, pedestrian
PUCPR+ (Hsieh et al., 2017)	✓	✗	Car	Motorcycle, pedestrian
CyCAR (Kouris et al., 2019)	✓	✓	Car	
UAVDT (Yu et al., 2020)	✓	✗	Car, bus, truck	Motorcycle, pedestrian
HARPY (Makrigiorgis et al., 2022)	✓	✗	Car, bus, truck	
RAI4VD (Puertas et al., 2022)	✓	✓	Car, bus, truck, motorcycle	Pedestrian
UIT-ADrone (Tran et al., 2023)	✓	✓	Car, bus, truck, motorcycle, pedestrian, bicycle	
VisDrone (Zhu et al., 2021)	✓	✗	Car, bus, truck, motorcycle, pedestrian, bicycle	

Table 2
Annotation counts and class distributions across training and test sets for the BASE and FINE dataset groupings, with our Songdo Vision dataset also shown separately.

Vehicle class	BASE dataset			FINE dataset			Songdo Vision dataset		
	Training	Test	Test ratio	Training	Test	Test ratio	Training	Test	Test ratio
Car	561,666	148,375	20.9%	266,745	52,260	16.4%	195,539	49,508	20.2%
Bus	15,587	2,202	12.4%	8,047	1,793	18.2%	7,030	1,759	20.0%
Truck	28,830	4,869	14.4%	14,305	3,452	19.4%	11,779	3,052	20.6%
Motorcycle	44,512	11,478	20.5%	30,925	10,309	25.0%	2,963	805	21.4%
Pedestrian	24,239	1,983	7.6%	1,260	243	16.2%			
Bicycle	4,472	276	5.8%	86	45	34.4%			
Total Vehicles	679,306	169,183	19.9%	321,368	68,102	17.5%	217,311	55,124	20.2%

computational demands, as self-attention scales the computational complexity quadratically with the input resolution. While our 140–150 m flight altitude ensured wide coverage, it inherently limited the detection of smaller objects such as pedestrians and bicycles. Lower altitudes and higher-resolution imaging improve small-object detection (Ren et al., 2018), but such configurations reduce spatial coverage and introduce operational trade-offs, such as the need for a larger drone fleet to maintain constant monitoring of the same geographical area.

After extensive trials, the 's' model scale with standard output heads was selected for its optimal balance of accuracy and inference speed. While we experimented with an additional P2/4 output layer, as utilized in our previous YOLOv5 setup (Espadaler-Clapés et al., 2023b), no significant performance gains were observed within the YOLOv8 architecture. Our object detector was trained using a multi-stage approach. Initially, the model was trained on the large-scale "BASE" dataset, which consists of a diverse collection of images from various sources, offering broad coverage of vehicle types and scenarios. This training commenced with weights pre-trained on the COCO dataset (Lin et al., 2014), establishing a solid foundation for transfer learning. Subsequently, the model was fine-tuned on the "FINE" dataset, a curated, high-quality subset of BASE, specifically chosen for its accurate annotations and higher-resolution images, facilitating fine-grained model refinement.

The datasets used for training, comprising images and annotations from eight public datasets and our Songdo Vision dataset (detailed in Section 3.2), are summarized in Table 1, which also specifies their usage across the BASE and FINE training stages. The public datasets include a variety of aerial images, characterized by varying altitudes, camera angles, perspectives, resolutions, illumination conditions, image quality, and train/test split ratios. For the public datasets, we adhered to the original train/test splits whenever available. In cases where no such split was provided, we applied an 80/20 division to maintain uniformity. Furthermore, datasets containing a high proportion of repetitive images were proportionally down-sampled to maintain a balanced representation across sources. This ensured that no single dataset dominated the training process, enhancing the model's ability to generalize across diverse traffic conditions.

To ensure consistency with our 140 m–150 m BEV scenes in Songdo, we performed extensive preprocessing and filtering of these datasets. Although a comprehensive description of these steps is omitted here for brevity, we focused on consolidating the diverse class labels found in the public datasets to streamline our trajectory extraction process, which utilizes four primary vehicle categories and two auxiliary categories: $C = 0, 1, 2, 3, 4, 5$, where 0 corresponds to cars (including vans and light-duty vehicles), 1 to buses, 2 to trucks, 3 to motorcycles, 4 to pedestrians, and 5 to bicycles. While only the four primary vehicle categories were used during inference, the auxiliary categories (pedestrians and bicycles) were retained during training to improve overall classification performance. To ensure that our FINE dataset meets stringent training specifications, we excluded certain datasets, such as VisDrone and UAVDT. For example, VisDrone's annotations primarily originate from non-BEV environments at lower altitudes, resulting in inconsistent object size representations and varied camera settings that complicated integration. Additionally, this dataset exhibited mediocre annotation accuracy, further justifying its exclusion from our FINE dataset.

Although pedestrians and bicycles were excluded during inference due to their underrepresentation in the training data (see Table 2) and diminutive sizes, we included them in the training phase with the aim of reducing false positives from similarly sized objects like motorcycles and improving the model's ability to distinguish between closely related categories. To ensure consistency and enhance reliability, certain object classes in the public datasets were either disregarded or consolidated into the categories

Table 3
Detection performance metrics on the Songdo Vision test set. Values before the slash (/) exclude the Songdo Vision train set from training, while **bold** values after the slash (/) include it.

Class	Precision	Recall	mAP@50	mAP@50-95
All	0.766/**0.911**	0.690/**0.935**	0.742/**0.951**	0.503/**0.711**
Car	0.934/**0.979**	0.971/**0.981**	0.970/**0.992**	0.702/**0.835**
Bus	0.862/**0.952**	0.853/**0.977**	0.906/**0.988**	0.690/**0.826**
Truck	0.737/**0.887**	0.410/**0.916**	0.593/**0.935**	0.421/**0.722**
Motorcycle	0.532/**0.827**	0.527/**0.866**	0.499/**0.888**	0.197/**0.463**

defined by C. For instance, in the VisDrone dataset (Zhu et al., 2021), where "pedestrian" and "people" are distinct labels, these categories were merged into a single "pedestrian" label, while classes like tricycles (including awning-tricycles) were excluded.

To optimize our object detection for the complex and variable conditions of drone-based surveillance, we employed a comprehensive suite of data augmentation strategies. Specifically, the training set was subjected to random scaling (±50%) and translations (±10% of the image dimensions), as well as horizontal flips applied with a probability of 0.5. In addition, we introduced random perturbations to image hue, saturation, and brightness by scaling these channels with multiplicative factors of 0.015, 0.7, and 0.4, respectively. Mosaic augmentation was applied deterministically (probability = 1.0) by combining four distinct training images into a single composite image, thereby enhancing the model's ability to generalize across varying spatial contexts and object scales (Hao and Zhili, 2020). Furthermore, the Albumentations framework (Buslaev et al., 2020) was employed to introduce additional perturbations, including Gaussian and median blur, grayscale conversion, and CLAHE, each applied with a probability of 0.01, to improve robustness against variations in texture and illumination. All these strategies were essential for adapting the model to real-world variations in vehicle appearance caused by changing lighting conditions, weather, and diverse urban landscapes.

Our training strategy employed stochastic gradient descent (SGD) optimization with YOLOv8's default hyperparameters, including an initial learning rate of 0.01, a final learning rate factor of 0.01, a momentum of 0.937, and a weight decay of 0.0005 to mitigate overfitting. Despite exploring various configurations, the default settings yielded the best performance. A batch size of 8 was used, with early stopping applied using a patience of 50 epochs, monitored on the test sets detailed in Table 2. Automatic mixed precision was enabled to improve efficiency without compromising accuracy. The loss function was calibrated with box, classification, and direction field loss gains of 7.5, 0.5, and 1.5, respectively. Training and inference were conducted at a fixed resolution of 1920×1920 pixels using letterbox padding to preserve aspect ratio. This higher resolution was critical for detecting small objects like motorcycles, while a confidence threshold of $\tau_d = 0.25$ and an IoU threshold of $\eta_{IoU} = 0.7$ ensured a balanced trade-off between precision and recall.

Songdo Vision, comprising 5,419 annotated images, was split into training and test sets, with 20% (1,084 images) designated as the test set, allocated for both model validation during training and final testing of our top-performing object detector. Table 2 shows the class distribution across the sets. Our top-performing model was selected based on its highest mean Average Precision (mAP) at an IoU of 0.5 (mAP@50) across all classes. As shown in Table 3, this model demonstrated remarkable performance across all vehicle types, achieving an overall mAP@50 of 0.951 and an exceptional mAP@50 of 0.888 for motorcycles, despite their smaller pixel size. Beyond conventional detection metrics, we also assessed the positional accuracy of BB centers, revealing a mean Euclidean error of approximately 2.21 ± 1.99 pixels (12.2 ± 10.9 cm), further reinforcing the reliability of our detection step for trajectory extraction. These results highlight the model's robustness in detecting small objects in BEV drone videos. To briefly define the key metrics: Precision indicates the proportion of correctly identified vehicles out of the total detected, while Recall reflects the ratio of correct detections to the actual number of vehicles in a class. The mAP@50 measures detection accuracy using an IoU threshold of 0.5, and mAP@50-95 evaluates performance over a range of IoU thresholds (0.5 to 0.95), with a step size of 0.05, capturing variability in BB overlaps.

The confusion matrix in Fig. 4 (left) illustrates the model's classification performance, highlighting accuracy and misclassification patterns across vehicle classes. The precision–recall curve in Fig. 4 (right) further demonstrates strong detection performance, with low false positive and false negative rates. While smaller objects like pedestrians and bicycles are occasionally misclassified as motorcycles, particularly near pedestrian crossings or bike lanes, these instances were not filtered due to the inherent challenges in accurate differentiation. Despite this, the model's robust performance on motorcycles, though slightly lower than for larger vehicles, highlights the challenges of detecting smaller objects in drone footage.

We also conducted an ablation study to assess the impact of our annotation efforts on detector performance. Specifically, we repeated the training procedure without the Songdo Vision dataset and observed substantial performance improvements across all metrics when it was included. As shown in Table 3, the mAP@50 improved by +28.15% and the mAP@50-95 by +41.35%, highlighting the importance of high-quality annotations. Notably, the inclusion of Songdo Vision led to significant gains in detecting motorcycles, with a +78.16% increase in mAP@50 and a +135.03% increase in mAP@50-95, demonstrating the value of this dataset in enhancing detection performance for smaller objects.

In conclusion, our object detection system has demonstrated robust performance in identifying various vehicle types in drone-based surveillance. While our current model faces challenges in detecting smaller objects like pedestrians and cyclists, we are actively exploring strategies to address these limitations. By incorporating data augmentation techniques, investigating advanced algorithms, and leveraging newly emerging datasets with more comprehensive annotations, we aim to further enhance the system's capabilities.

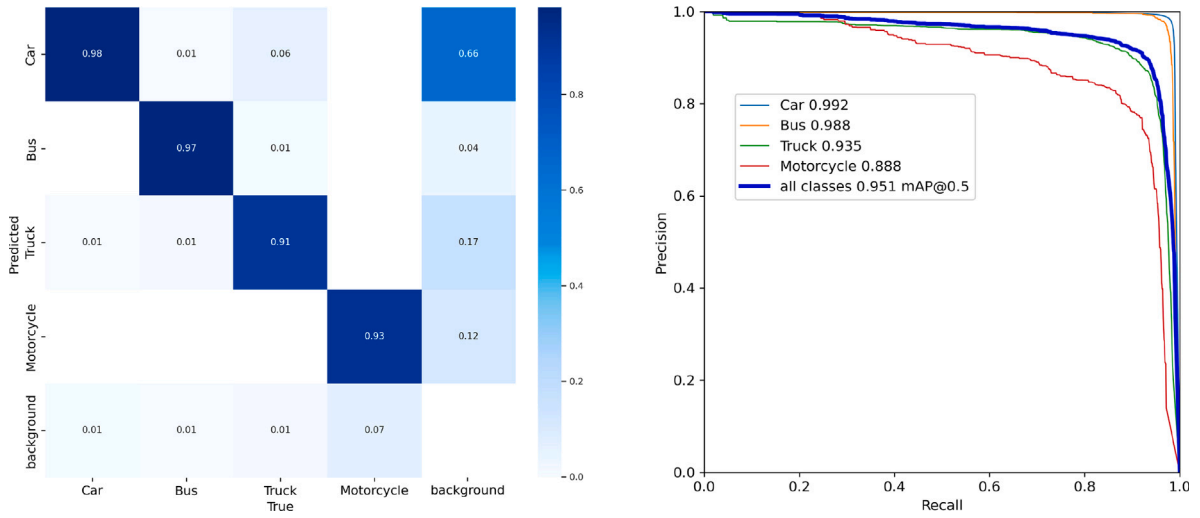


Fig. 4. Left: Normalized confusion matrix. Right: Precision–recall curve.

3.4. Object tracking

Object tracking, specifically MOT, is a crucial component of our framework, assigning unique IDs to each vehicle and consistently tracking their movements across video frames. This enables continuous tracking, even during temporary occlusions, providing valuable insights into vehicle behavior and interactions over time.

For the kth frame of video v, the MOT algorithm processes the detection set $\mathcal{D}_v[k]$ to generate a set $\mathcal{T}_v[k]$ of $N_v^t[k]$ tracks, each represented as:

$$\mathcal{T}_v[k] = \left\{ \left(id^{(i)}[k],\ x^{(i)}[k],\ y^{(i)}[k],\ w^{(i)}[k],\ h^{(i)}[k],\ c^{(i)}[k],\ s^{(i)}[k] \right) \in \mathcal{I}_v[k] \times \mathcal{D}_v[k] \right\}_{i=1}^{N_v^t[k]}, \tag{2}$$

where '×' denotes the Cartesian product, $\mathcal{I}_v[k]$ is the set of unique vehicle IDs in v up to frame k, and $id^{(i)}[k] \in \{1, 2, \ldots, V_v[k]\}$ is the unique ID for each tracked vehicle, consistent across all frames. The number of active tracks in F_v^k is $N_v^t[k]$, while $V_v[k]$ is the total number of unique vehicles observed up to frame k. The remaining symbols in Eq. (2) have the same meanings as in Eq. (1).

After applying tracking to all frames F_v^k, $k \in \{1, \ldots, N_v\}$, where N_v is the total number of frames in v, we implement a class label refinement step for $c^{(i)}[k]$ in $\mathcal{T}v[k]$ to maintain classification consistency across frames for each tracked vehicle. Although our observations indicate such inconsistencies are very rare, we formalize this refinement through the following aggregation function:

$$f_c(\mathrm{id}) = \arg\max_{\mathrm{c} \in C} \left\{ \sum_{k=1}^{N_v} \sum_{i=1}^{N_v^t[k]} s^{(i)}[k] \cdot \mathbf{1}\left(c^{(i)}[k] = \mathrm{c}\right) \cdot \mathbf{1}\left(id^{(i)}[k] = \mathrm{id}\right) \right\}, \quad \mathrm{id} = 1, \ldots, V_v[N_v], \tag{3}$$

where $\mathbf{1}(\cdot)$ is the indicator function that is 1 if the condition is true and 0 otherwise. In this process, for each vehicle ID, the ultimate class label is assigned by selecting the class $c \in C$ that maximizes the sum of confidence scores across all frames where that ID appears, that is, we refine the class labels to $c^{(i)}[k] = f_c(id^{(i)}[k])$.

The MOT algorithm efficiently manages the assignment of new IDs and the removal of vehicles that no longer appear in the frame, ensuring that $N_v^t[k] \leq V_v[k]$ is maintained for all frames. While intermittent tracking failures due to prolonged occlusions or misdetections can result in multiple IDs for a single vehicle, such cases are rare and primarily occur with motorcycles navigating complex road infrastructure, as illustrated in Fig. 3. These instances have minimal impact on our primary goal of extracting precise trajectories.

We adopted the BoT-SORT algorithm (Aharon et al., 2022) for its enhanced Kalman filter state vector and efficient EMC, which improve tracking accuracy for fast-moving and smaller vehicles like motorcycles. As an enhancement over the original ByteTrack (Zhang et al., 2022b), BoT-SORT's optimizations make it effective in handling sporadic misdetections and suitable for dynamic environments. Given the minimal occlusions in our aerial footage, integrating a dedicated Re-ID module would yield limited improvement, making BoT-SORT preferable due to its balance of performance and computational efficiency.

To manage occasional missed detections of smaller objects, particularly motorcycles, caused by the detector's challenges in complex scenes, we employ a track buffer of $T_{\text{buff}} = 30$ frames. This buffer retains information about tracks that may temporarily disappear due to occlusion or missed detections, helping prevent premature ID deletion. However, in our Songdo experiment, its importance is limited since obstructions are rare and primarily confined to a few small bridges located away from main road arteries and intersections, typically at exit roads from underground garages (see Fig. 1). The BoT-SORT algorithm manages track initiation and termination using several thresholds: a new track is created if the detection confidence score exceeds $\tau_{\text{n}} = 0.6$, high-confidence

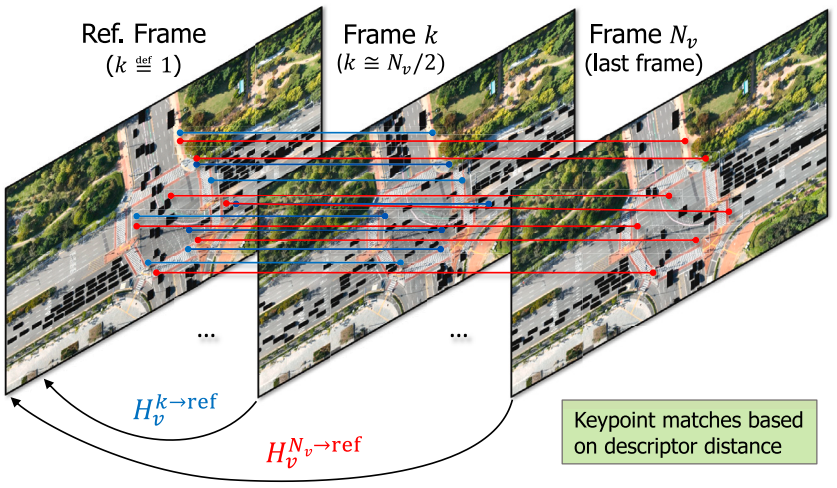


Fig. 5. Illustration of the image registration process. Keypoints from static background features are matched between frames and used to compute a homography matrix $H_v^{k\to\text{ref}}$ to align the frames. Exclusion masks prevent keypoint detection in dynamic vehicle regions, ensuring stable matching.

matches are made above $\tau_\text{h} = 0.5$, and low-confidence matches are allowed above $\tau_\text{l} = 0.1$. Associations between detections and tracks are ensured with a matching threshold of $\tau_\text{m} = 0.8$. Additionally, BoT-SORT's efficient EMC method based on sparse OF compensates for camera movements by adjusting tracking to mitigate global background motion errors. This allows us to perform track stabilization as a subsequent step to object tracking, rather than relying on traditional video stabilization earlier in the pipeline (see Fig. 2). The benefits of this approach are further discussed in Section 3.5.

Although BoT-SORT was originally validated on pedestrian tracking in street-level perspectives through the MOT Challenge dataset (Aharon et al., 2022), several factors support its reliability for our aerial vehicle tracking application. Unlike object detection, for which we have a representative sample of ground truth annotations, a comparable dataset for vehicle trajectories is unavailable in our context. Nevertheless, BoT-SORT inherently integrates appearance-based re-identification principles through its advanced Kalman filtering and efficient EMC, mitigating global camera movements and significantly reducing identity-switching risks. Moreover, our setup benefits from favorable tracking conditions, including high-resolution BEV imagery captured at 29.97 FPS and vehicle speeds typically below 80 km/h, which ensure sufficient frame-to-frame overlap. Complete occlusions are exceedingly rare, minimizing the need for external visual-based ReID modules. Additionally, qualitative validation through manual inspection showed consistent tracking across sampled trajectories, while the comparison with trajectories recorded by the instrumented probe vehicle provided further confidence in the tracking accuracy.

To further assess the suitability of BoT-SORT, we conducted a quantitative comparison with ByteTrack by evaluating vehicle trajectory length distributions across four representative drone videos from our dataset, see Section 4.1. Both algorithms produced virtually identical results, even in highly dense urban conditions and high-speed vehicle scenarios exceeding 150 km/h, as indicated by negligible KL divergence (0.0005) and matching trajectory counts. These findings align with previous studies showing that BoT-SORT achieves similar or higher accuracy than ByteTrack, despite ByteTrack's faster processing speed (Petersson and Kifle Solomon, 2024). Combined with qualitative assessments, this analysis supports our choice of BoT-SORT. While ByteTrack offers comparable performance, BoT-SORT's superior built-in EMC and better handling of intermittent detections aligned more closely with our objectives. FairMOT was also considered; however, its joint detection-ReID architecture requires extensive fine-tuning and has high computational costs, which are impractical due to the lack of annotated aerial tracking data. Thus, FairMOT was deemed unsuitable for large-scale processing.

3.5. Track stabilization

To mitigate the effect of drone movements on the extracted trajectories, we "stabilize" each track $\mathcal{T}_v[k]$ by aligning the coordinates of each frame F_v^k ($k = 2, \ldots, N_v$) with a reference frame F_v^ref, which is set as the first frame of the video v, i.e., $F_v^\text{ref} \triangleq F_v^1$. As illustrated in Fig. 5, we achieve this alignment through image registration by identifying corresponding keypoints between pairs of grayscale frames (F_v^ref and F_v^k), with optional CLAHE applied to improve contrast. The matched keypoints are used to compute a 3×3 homography matrix $H_v^{k\to\text{ref}}$ using the least squares method, enhanced with a variant of the RANSAC algorithm to reject outliers. Despite the marginally higher computational cost, we choose a projective transformation over an affine one to more effectively account for potential camera tilting caused by wind.

This approach ensures consistent alignment throughout the footage without accumulating errors over time, as each frame is independently aligned to the fixed reference frame rather than cumulatively re-using previous transformations. Consequently, for static scenes, the likelihood of error remains consistent for every frame, particularly considering the quasi-stationary nature of our

drone setups. Additionally, this method significantly reduces storage and computational overhead compared to full-frame video stabilization techniques.

Recent studies (Jin et al., 2021; Efe et al., 2021) suggest that classical image matching algorithms may perform comparably to the perceived SOTA represented by learning-based approaches like SuperPoint or SuperGlue. These findings highlight the importance of careful hyperparameter tuning for classical methods and point out that DL-based methods rely heavily on large training datasets and require fine-tuning for specific tasks. Given the absence of annotated training data for fine-tuning DL-based methods and the computational cost associated with GPU-based inference, we opted for classical methods. This choice also aligns with the practical need for computational efficiency, considering our dataset size of approximately 12TB. Therefore, classical methods offer a robust solution without compromising accuracy, making them suitable for our experimental conditions.

3.5.1. Implementation details

We extract up to K and K_{ref} keypoints from each frame F_v^k $(k = 2, \dots, N_v)$ and the reference frame F_v^{ref}, respectively, using a local feature detector. To ensure stable frame alignment, we focus keypoint detection on static background elements by leveraging the BBs from $\mathcal{D}_v[k]$ as "exclusion masks". These masks prevent keypoint detection within vehicle regions, which would otherwise introduce unstable matches due to vehicle movement. To account for potential inaccuracies in BBs, each box is enlarged by a factor $\epsilon > 0$, adjusting mask dimensions to $\tilde{w} = w(1+\epsilon)$ and $\tilde{h} = h(1+\epsilon)$.

Keypoint matches between each pair $(F_v^{\text{ref}}, F_v^k)$ are identified based on a similarity measure calculated from keypoint descriptors, typically using Euclidean distance or cosine similarity, depending on the descriptor type. These potential matches, denoted as $m_v^{k\to\text{ref}}$, are refined using Lowe's ratio test (Lowe, 2004), also known as SNN test, which filters out non-discriminative matches using a threshold $\theta_{\text{SNN}} \in (0, 1]$, resulting in a refined match set $\tilde{m}_v^{k\to\text{ref}}$. This refined set is then used by a robust estimator to compute the homography matrix $H_v^{k\to\text{ref}}$. The confidence level τ, the maximum number of iterations Γ, and the epipolar threshold η are tuning parameters for most RANSAC algorithms. To reduce computational load when determining homography matrices, we downscale the 4K input frames F_v^k $(k = 1, \dots, N_v)$ using a scale factor ρ, resulting in adjusted frame dimensions $\tilde{w}_I = \rho \times w_I$ and $\tilde{h}_I = \rho \times h_I$.

The derived homography matrix $H_v^{k\to\text{ref}}$ can transform any pixel coordinates from frame F_v^k to the reference frame. For a generic point $p_v^k = [x,\ y]^T$ in frame F_v^k, its corresponding location in the reference frame is:

$$p_v^{\text{ref}} = \begin{bmatrix} x_{\text{ref}} & y_{\text{ref}} \end{bmatrix}^T = \begin{bmatrix} x'/z & y'/z \end{bmatrix}^T, \tag{4}$$

where

$$\begin{bmatrix} x' & y' & z \end{bmatrix}^T = H_v^{k\to\text{ref}} \begin{bmatrix} x & y & 1 \end{bmatrix}^T. \tag{5}$$

Here, the point p_v^k is converted to homogeneous coordinates by appending 1, the homography is applied, and the result is normalized by dividing by z. This transformation can be concisely represented as $p_v^{\text{ref}} = H_v^{k\to\text{ref}}(p_v^k)$, directly mapping each point from the original frame to its position in the reference frame.

In our track stabilization approach, we apply these transformations to all BB coordinates in $\mathcal{T}_v[k]$, resulting in a new set of stabilized tracks, $\mathcal{T}_v^{\text{ref}}[k]$. We first convert the BBs from the YOLO format (x, y, w, h) into a four-point format representing each box's corners. Each corner is then transformed using $H_v^{k\to\text{ref}}$ to align with the reference frame. After transformation, the new positions are converted back into the YOLO format by recalculating the center coordinates and dimensions, ensuring consistency. This direct application of homography to track data bypasses the need to stabilize the entire video, saving storage space and eliminating artifacts that can obscure parts of the scene, thereby enabling vehicle detection across the entire recorded area. This is a key feature of our work, emphasizing the novelty of stabilizing tracks rather than the video itself.

Furthermore, our masking strategy introduces another key innovation. Unlike conventional methods that warp each video frame to a reference frame before detection and tracking, our approach leverages the detected vehicles to enhance image registration. By applying exclusion masks based on detected BBs, we avoid keypoint matches on moving objects, which can undermine stabilization (Zheng et al., 2024). This reordering improves accuracy and allows for less stringent hyperparameters without additional computational costs. Our stabilization process is available as the open-source library *stabilo* (Fonod, 2025b), which supports both video and track stabilization using user-provided masks, and can be integrated into diverse applications including autonomous surveillance, mapping, and environmental monitoring frameworks beyond vehicle tracking.

3.5.2. Method and parameter search campaign

To systematically validate and refine our selection of classical algorithms and hyperparameters, we utilized 29 unique aerial scenes S, each representing a randomly selected frame from our fleet of 10 drones monitoring 20 intersections. With multiple drones covering the same intersections, this compilation ensures a complete array of viewpoints, totaling $|S| = 29$ distinct scenes similar to those shown in Fig. 1 (right).

For each method and hyperparameter combination, we performed a comprehensive evaluation using all scenes. Each scene was subject to $N_t = 100$ trials involving both photometric and homographic distortions, amounting to 2900 evaluations per combination. Photometric distortions include brightness variations of up to ±25%, Gaussian blur with kernel sizes of up to 5 pixels, saturation adjustments within ±5%, and fog effects with an intensity of up to 10%. These distortions simulate various lighting conditions, weather scenarios, and camera focus variations to challenge the robustness of feature detection and matching algorithms. We then applied random homography transformations $H_s^{i\to i'}$ to each scene $s \in S$. These transformations, designed to mimic drone movements and environmental effects like wind gust or altitude changes, included rotations up to ±15 degrees, translations up to ±10% on both

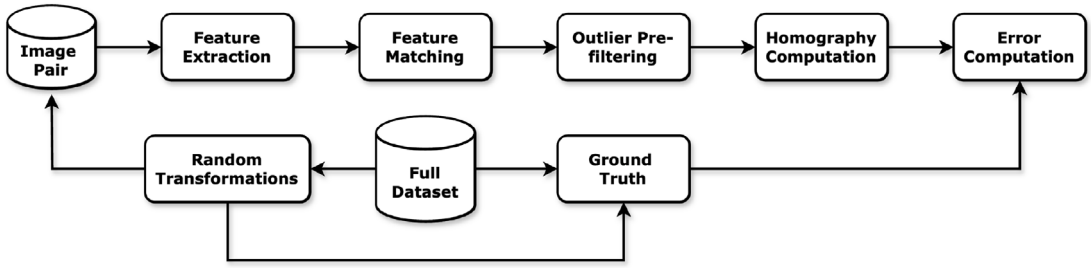


Fig. 6. Overview of the image registration campaign.

axes, scale changes up to ±5%, and perspective shifts with coefficients up to 5×10^{-5} to represent subtle changes in drone pitch or roll.

Using image pairs of original and distorted scenes, we applied the image registration techniques discussed earlier to estimate the homography matrix $\hat{H}_s^{i' \to i}$, using $H_s^{i \to i'}$ as the ground truth. Comparing the estimated transformations against the actual ones ensures that our algorithms can robustly handle and correct the dynamic transformations typically encountered by drones in real-world scenarios. Fig. 6 provides an overview of our experimental setup, highlighting the rigor of our testing environment tailored for drone video analysis under varying and challenging conditions.

We implemented several prominent local feature detectors including SIFT, RSIFT, BRISK, ORB, KAZE, and A-KAZE, along with various RANSAC implementations such as the original RANSAC algorithm, MAGSAC++, GC-RANSAC, DEGENSAC, etc. Both BF and FLANN matchers were evaluated, with the options to use exclusion masks and apply CLAHE. We conducted extensive grid searches to optimize parameters such as the SNN ratio threshold (θ_{SNN}), the maximal number of extracted keypoints (K and K_{ref}), the down-sample ratio (ρ), and the epipolar threshold (η), aiming to balance accuracy and computational efficiency. Following recommendations from recent studies (Jin et al., 2021), we fixed τ at 0.999999 due to its minimal sensitivity to changes and set the maximum number of iterations Γ at 5000 for all RANSAC algorithms to ensure sufficient convergence. Finally, we opted for unidirectional matching, avoiding bidirectional approaches to reduce computational demands.

Traditionally, the mean Average Accuracy (mAA) metric is used to evaluate the effectiveness of image registration. However, this requires knowledge of the camera intrinsics and computation of the essential matrix. Alternatively, homography estimation accuracy (HEA) is a widely used metric (Efe et al., 2021) for image registration evaluation, defined as:

$$\text{HEA} = \frac{1}{N_t \times |\mathcal{S}|} \sum_{s \in \mathcal{S}} \sum_{i=1}^{N_t} \begin{cases} 1, & \text{if } \left(\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left\| p - \hat{H}_s^{i' \to i} \left(H_s^{i \to i'}(p) \right) \right\| \right) \leq \varepsilon \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

This equation calculates the mean accuracy across all transformations for each scene, where each corner's displacement after a round-trip transformation through the true homography H and its estimation $\hat{H}$ is computed. The set $\mathcal{P}$ contains the 2D coordinates of the four corner points of the original scene, and $H(\cdot)$ or $\hat{H}(\cdot)$ represent the homography transformation of a 2D point as described in (4) and (5). If the average displacement is within a threshold ε, it contributes a '1' (accurate) to the mean; otherwise, it contributes a '0'.

Although HEA effectively measures global scene alignment, it may overlook local inaccuracies critical for tracking vehicles. To overcome this limitation, we introduce a new metric, mean IoU (MIoU), specifically designed to measure accuracy by comparing re-projected BBs, thus directly assessing object-level registration quality essential for reliable vehicle trajectory extraction. Formally, MIoU is defined by:

$$\text{MIoU} = \frac{1}{N_t \times |\mathcal{S}|} \sum_{s \in \mathcal{S}} \sum_{i=1}^{N_t} \left(\frac{1}{|\mathcal{B}_s|} \sum_{B \in \mathcal{B}_s} \frac{B \cap \hat{H}_s^{i' \to i} \left(H_s^{i \to i'}(B) \right)}{B \cup \hat{H}_s^{i' \to i} \left(H_s^{i \to i'}(B) \right)} \right). \tag{7}$$

This equation measures the mean IoU for BBs across all scenes and transformations, where $\mathcal{B}_s$ denotes the set of BBs in scene $s \in \mathcal{S}$. It computes the IoU between each original BB B and its transformed version after applying the true and estimated homographies. The advantage of this metric is that it does not require a pixel threshold and yields values ranging from 0 to 1, making it suitable for our application and valuable addition to HEA.

In practical terms, HEA provides a measure of the overall consistency of frame alignment, which is critical for maintaining accurate spatial relationships between frames during the trajectory extraction process. Meanwhile, MIoU directly reflects the precision of object alignment by evaluating the overlap of re-projected vehicle bounding boxes, thereby serving as an indicator of the accuracy of detected vehicle positions after transformation. Together, these metrics ensure that both global frame alignment and local object positioning are accurately preserved, which is essential for reliable kinematic analysis and trajectory-based traffic studies.

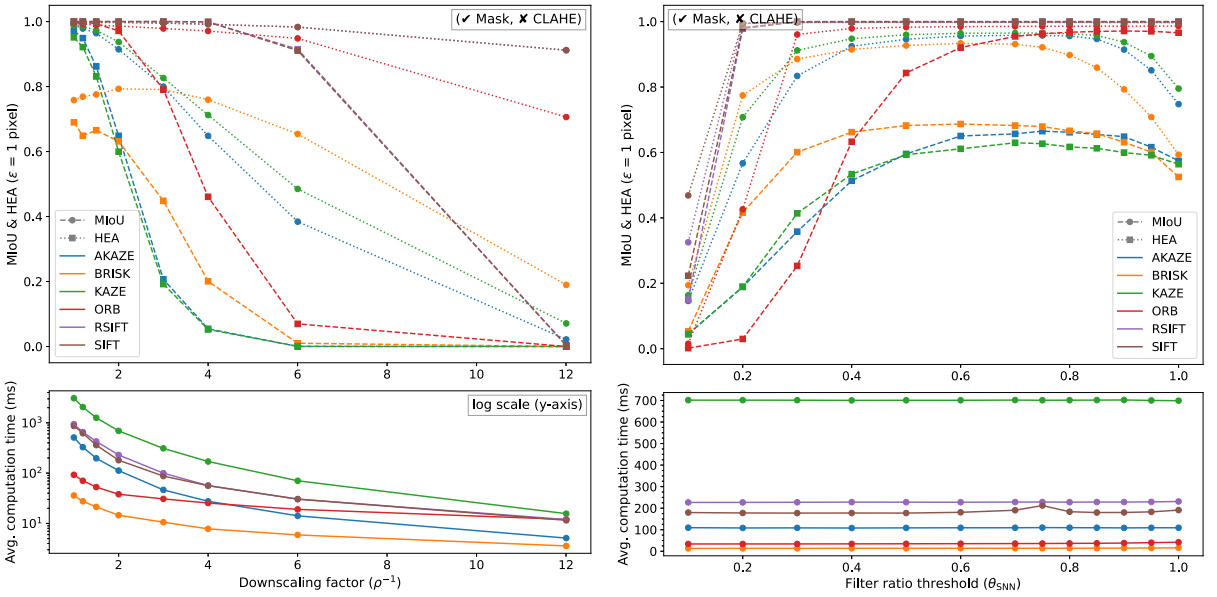


Fig. 7. Effect of downscaling ratio and SNN filter ratio threshold on the image registration performance.[1]

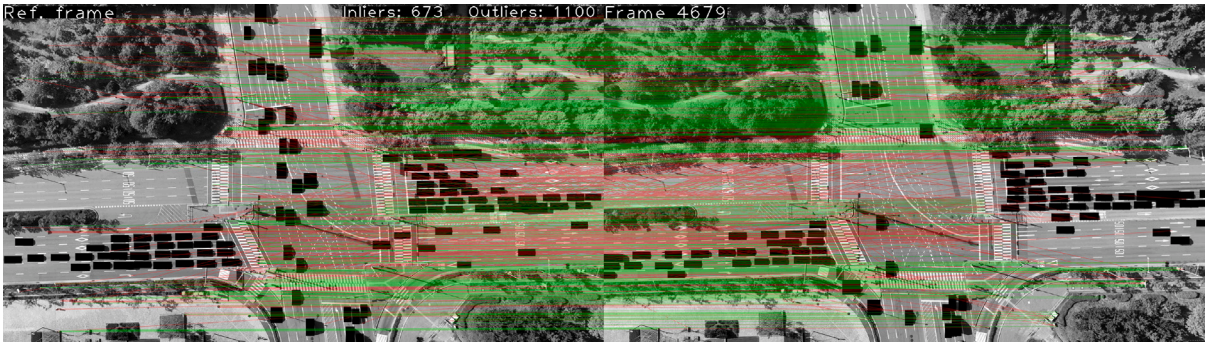


Fig. 8. Image matching between the reference frame and the last frame of a video. Green lines indicate inliers used for homography computation, while red lines represent outliers rejected by the MAGSAC++ algorithm. Exclusion masks effectively prevent keypoint detection in vehicle regions.

While a detailed presentation of our campaign results is beyond the scope of this paper, we will make *stabilo-optimize* (Fonod, 2025c), an open-source optimization tool for stabilo, publicly available to enable replication and adaptation of our methods and hyperparameter tuning for similar image registration tasks. Fig. 7 offers insights into the optimization process, particularly highlighting the impact of downscaling and SNN threshold on image registration performance.

Ultimately, our findings suggest the following optimal set of methods and parameters: an ORB detector with $K = 2000$ and $K_{\text{ref}} = 4000$, a BF feature matcher with $\theta_{\text{SNN}} = 0.9$, and MAGSAC++ with $\tau = 0.999999$, $\Gamma = 5000$, and $\eta = 2$, coupled with downscaling by a factor of $\rho = 0.5$ and the use of BBs with $\epsilon = 0.15$ as exclusion masks. The use of CLAHE did not provide any significant benefits. Except for θ_{SNN} or ρ, these parameters were fixed for the considered trials. As shown in Fig. 7, the selected combination of methods and parameters, particularly ORB with a downscaling factor of $\rho = 0.5$, achieves high MIoU and HEA values. This performance is on par with leading results reported in image matching studies (Efe et al., 2021; Jin et al., 2021) and effectively balances superior accuracy with the computational efficiency required to process our extensive 12TB of video data.

Fig. 8 illustrates image matching between the reference (first) frame and the last frame ($N_\upsilon = 4679$) of an example video υ, spanning approximately 2 min and 36 s. Among the 1773 keypoints that passed the SNN threshold, 673 inliers were retained for computing $H_\upsilon^{N_\upsilon \to \text{ref}}$, while MAGSAC++ rejected the remaining 1100 as outliers. Notably, all matched keypoints lie outside the exclusion masks covering detected vehicles. Unlike the discarded matches, the inliers exhibit consistent directionality, indicating a robust subset for accurate homography estimation.

[1] All computational timings were obtained on a system equipped with an Intel Core i9-13900KF processor, NVIDIA GeForce RTX 3090 graphics card, and 64 GB of RAM, operating on Ubuntu 22.04.3 LTS.

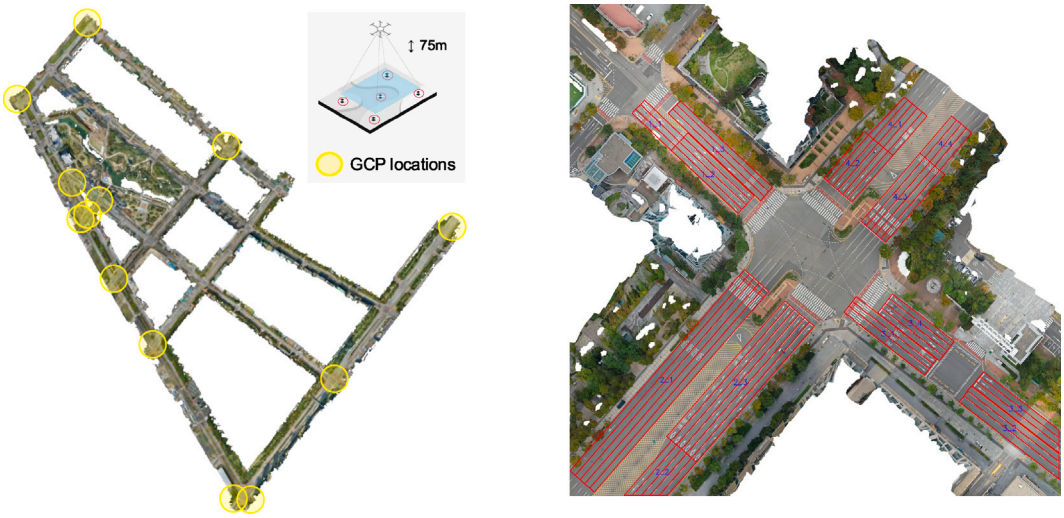


Fig. 9. Left: The large orthophoto encompassing all 20 monitored intersections and showing the locations of the GCPs used during its creation. Right: Orthophoto cut-out example of an intersection, with overlaid road section and lane segmentations.

3.6. Georeferencing

3.6.1. Orthophoto creation and road segmentation

In our study, a detailed orthophoto encompassing all 20 monitored intersections within the Songdo experimental area was produced, as shown in Fig. 9 (left). This was achieved using a dedicated DJI Mavic 2 Pro drone flying at an altitude of 75 m, capturing overlapping high-resolution (6000×4000 pixels) and distortion-free imagery that was then meticulously stitched using sophisticated photogrammetry methods facilitated by Dromii's in-house Drone as a Service (DaaS) platform.[2] The drone captured images over the course of several days during the experiment and on additional dates to further enhance the final orthophoto.

The geospatial accuracy of the orthophoto was significantly enhanced using 13 GCPs, each marked with a 1×1 m QR code and strategically placed across the experimental area, see Fig. 9 (left). These GCPs were calibrated using an EMLID RS2+ multi-band RTK-GNSS receiver, featuring multi-feed antennas for multipath rejection and a fast RTK convergence. According to the manufacturer, this rover can reliably ensure geolocation accuracy up to 8 km from the base station. The final orthophoto boasts a resolution of $209{,}181 \times 181{,}692$ pixels and a file size exceeding 30 GB, providing an unprecedented level of detail.

For focused traffic analysis, $15{,}000 \times 15{,}000$ pixel cut-outs centered on each intersection were first extracted from the large orthophoto, see Fig. 9 (right) of an illustration of a cut-out. These cut-outs sufficiently cover areas our drones monitor and are instrumental for in-depth traffic analysis and visualizations. DL-based techniques were integrated into the photogrammetry process to systematically exclude vehicles from the imagery, ensuring that the orthophoto is devoid of transient obstructions such as cars. Tall objects such as buildings were also removed to prevent parallax errors and ensure reliable planar homography for georeferencing. These exclusions do not affect the accuracy of georeferencing, as sufficient areas remain for extracting relevant feature points. Crucially, the primary objective for accurate trajectory georeferencing is to ensure feature correspondences at the road level.

Each of the 20 cut-outs underwent manual segmentation to delineate road sections and lanes, as illustrated in Fig. 9 (right). Sections refer to areas that group lanes moving in the same direction within defined parts of the intersection. Each section is labeled as "N_G", where N designates the specific intersection node and G represents the grouping of lanes by direction and number. Lanes within each section are numbered sequentially from the innermost lane outward. This manual segmentation required approximately 50 working hours, involving similar iterative validation procedures as described previously for vehicle bounding box annotations (cf. Section 3.2). This enabled linking each trajectory point directly to a specific road section and lanes, enhancing the dataset's utility and providing invaluable metadata for intersection-level traffic analysis.

3.6.2. Trajectory georeferencing

The objective of georeferencing is to convert vehicle trajectories in $\mathcal{T}_u^{\text{ref}}$ into real-world geographic coordinates, such as longitude and latitude. This transformation, illustrated in Fig. 10, is achieved through a three-step process, where a key novelty of our approach is the introduction of a "master frame" as an intermediary CS. Using a unique master frame, selected based on optimal image quality and consistent reference features, for each intersection, we ensure that trajectories extracted from different drone viewpoints (with varying orientations, hovering locations, or altitudes) are consistently transformed into a unified coordinate system.

[2] http://new.dromii.com/

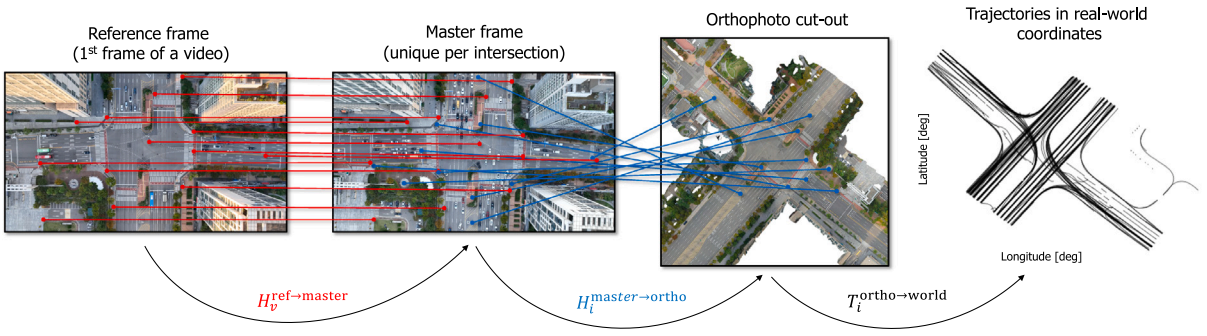


Fig. 10. Illustration of the steps involved in the georeferencing process.

Step 1: For each video v, we calculate a homography matrix $H_v^{\text{ref}\to\text{master}}$, which maps the stabilized tracks from the reference frame F_v^{ref} to a unique master frame F_i^{master} designated for each intersection. The master frame serves as a unified basis to ensure consistent alignment across all videos capturing the same location.

Step 2: A homography matrix $H_i^{\text{master}\to\text{ortho}}$ is then computed to map coordinates from the master frame to the orthophoto coordinates F_i^{ortho}. Here, F_i^{ortho} is the orthophoto cut-out corresponding to intersection i. The complete transformation is expressed as:

$$H_v^{\text{ref}\to\text{ortho}} = H_i^{\text{master}\to\text{ortho}} \cdot H_v^{\text{ref}\to\text{master}}. \tag{8}$$

This composite homography $H_v^{\text{ref}\to\text{ortho}}$ allows us to map any point p_v^{ref} in the reference frame F_v^{ref} directly to its corresponding point p_i^{ortho} in the orthophoto cut-out F_i^{ortho}, i.e., $p_i^{\text{ortho}} = H_v^{\text{ref}\to\text{ortho}}(p_v^{\text{ref}})$. Applying this transformation to all stabilized vehicle coordinates in $\mathcal{T}_v^{\text{ref}}[k]$, we obtain the vehicle trajectories in the orthophoto CS.

Step 3: The transformation from orthophoto coordinates to real-world geographic coordinates is performed using a 2×3 affine transformation matrix $T_i^{\text{ortho}\to\text{world}}$ derived from the ith orthophoto metadata. This matrix enables the accurate conversion of pixel coordinates into geographic coordinates as follows:

$$\begin{aligned} \phi &= a_i \cdot x_{\text{ortho}} + b_i \cdot y_{\text{ortho}} + t_{x_i}, \\ \lambda &= c_i \cdot x_{\text{ortho}} + d_i \cdot y_{\text{ortho}} + t_{y_i}, \end{aligned} \tag{9}$$

where x_{ortho} and y_{ortho} are pixel coordinates in the orthophoto CS, and a_i, b_i, c_i, d_i, t_{x_i}, and t_{y_i} are the parameters of $T_i^{\text{ortho}\to\text{world}}$. These parameters account for scale, rotation, and translation adjustments required to map pixel coordinates to real-world coordinates ϕ (latitude) and λ (longitude). By systematically applying these steps, we ensure precise conversion of all stabilized vehicle center coordinates $(x^{(i)}[k], y^{(i)}[k])$ in $\mathcal{T}_v^{\text{ref}}[k]$ into both global geographic CS (latitude and longitude, EPSG:4326, WGS84) and local Cartesian CS (X and Y, EPSG:5186).

This three-step georeferencing approach addresses the challenges of directly matching drone imagery with orthophotos, which differ significantly due to orthorectification, resolution, and excluded areas. Direct matching for each video would be more time-consuming due to higher orthophoto resolution and would likely introduce varying errors, as the homography between each reference frame and the orthophoto needs to be computed independently. By introducing the master frame as an intermediary CS, our method consolidates this process into a simple homography computation per video, converting potential varying errors into a consistent, systematic error for all trajectories at a given intersection. This approach simplifies the transformation process by relying on $H_v^{\text{ref}\to\text{master}}$ to match visually similar images, thereby improving reliability. To select the optimal master frame, we sorted reference frames based on their proximity to the average hovering location and chose the one with the least vehicle coverage among the top ten. This multi-stage approach contributes to the accuracy of trajectory mapping and enhances consistency across datasets obtained from multiple viewpoints, representing an important aspect of our methodology.

3.6.3. Optimization of orthophoto cut-out resolution for accurate georeferencing

While the transformation $T_i^{\text{ortho}\to\text{world}}$ is directly obtained from the orthophoto creation process, the homographies involved in georeferencing must be computed using image registration techniques similar to those used in track stabilization. Since these homographies are computed only once per video ($H_v^{\text{ref}\to\text{master}}$) and once per intersection ($H_i^{\text{master}\to\text{ortho}}$), computational efficiency is not a limiting factor. This allows for the use of RSIFT, which employs a square root (Hellinger) kernel to measure the similarity of SIFT descriptors instead of the standard Euclidean distance. In our implementation, we set a small positive parameter of 10^{-8} for RSIFT normalization to prevent division by zero. Optimization experiments confirmed that this approach improves performance across all stages of image registration (Arandjelović and Zisserman, 2012). We configured the implementation with a maximum of $K = 250{,}000$ feature points, a BF matcher with $\theta_{\text{SNN}} = 0.55$, and MAGSAC++ with $\tau = 0.999999$, $\Gamma = 10{,}000$, and $\eta = 3$. Furthermore, we did not apply CLAHE or exclusion masks, as the vehicles in the reference and master frames differ, and the orthophoto does not contain vehicles.

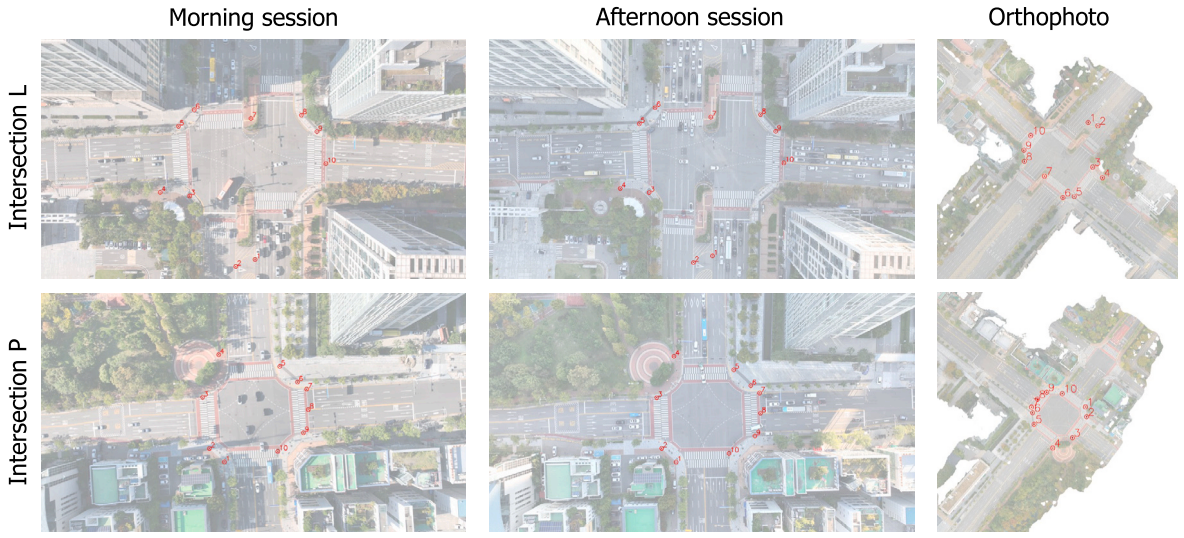


Fig. 11. Locations of the GCPs for the selected intersections L and P. Each image has been slightly dimmed to enhance the visibility of the overlaid markers.

Table 4
Image matching performance between selected frames and orthophoto cut-outs at various resolutions.

Cut-out resolution (px)	Mean computation time (s)	Reprojection error (px)	Mean # of inliers	Minimum # of inliers
2′000	2.584	3.225 ± 2.748	116.75	39
3′000	4.060	1.945 ± 1.357	189.25	62
4′000	8.632	1.671 ± 1.315	258.75	110
5′000	15.004	1.479 ± 1.014	235.50	96
6′000	23.337	1.424 ± 1.064	236.25	107
7′000	31.212	1.317 ± 0.888	234.50	95
8′000	41.978	1.323 ± 0.970	198.25	82
9′000	51.306	1.304 ± 0.928	208.00	104
10′000	53.432	1.421 ± 0.890	151.75	57
11′000	55.936	1.454 ± 0.942	145.75	71
12′000	58.552	1.378 ± 0.936	130.25	53
13′000	61.486	1.312 ± 0.886	115.25	38
14′000	64.622	1.419 ± 0.932	121.00	50
15′000	67.898	1.409 ± 0.908	96.50	37

To validate the reliability of our georeferencing process, we conducted a comprehensive evaluation to determine the optimal resolution for orthophoto cut-outs. While the 4K reference and master frames were used at their full resolution without downscaling, we performed an ablation study to assess the impact of different resolutions for the orthophoto cut-outs. Specifically, we selected two intersections, L and P. For each intersection, we chose two reference frames, one from a morning and one from an afternoon flight session, to account for varying lighting conditions. We meticulously annotated these frames and the full-sized cut-outs with 10 corresponding GCPs for each frame and cut-out pair, aiming for pixel-level accuracy (see Fig. 11 for illustration). This resulted in a total of 40 GCP correspondences for validation.

For each combination of image frame and orthophoto cut-out, we computed the homography matrix using the specified image registration techniques and parameters. This process was repeated for various orthophoto cut-out resolutions. Mean and standard deviation of pixel reprojection errors were calculated by transforming points from the orthophoto cut-outs back to the image frame and comparing them with the original GCPs. Table 4 summarizes these errors for each resolution, along with the mean computation time and the mean and minimum number of inliers after MAGSAC++ based outlier rejection. The results may include some human error due to manual GCPs assignment. All computations were performed on the same infrastructure reported in Section 3.5.2 to ensure consistency in timing measurements.

From the results, we observe that increasing the orthophoto cut-out resolution improves matching accuracy, as indicated by a decrease in reprojection error. Although the lowest error is achieved at 9000 pixels (1.304 px) and comparable results are seen at 7000 pixels (1.317 px), the 8000-pixel resolution offers a robust trade-off between accuracy and computational efficiency. Importantly, based on our observations, the minimum number of inliers must remain above approximately 30 to ensure reliable homography estimation. Additionally, we note that the number of inliers decreases at higher resolutions, likely due to the increased image detail causing more mismatches and the computational limits of feature matching algorithms at higher resolutions.

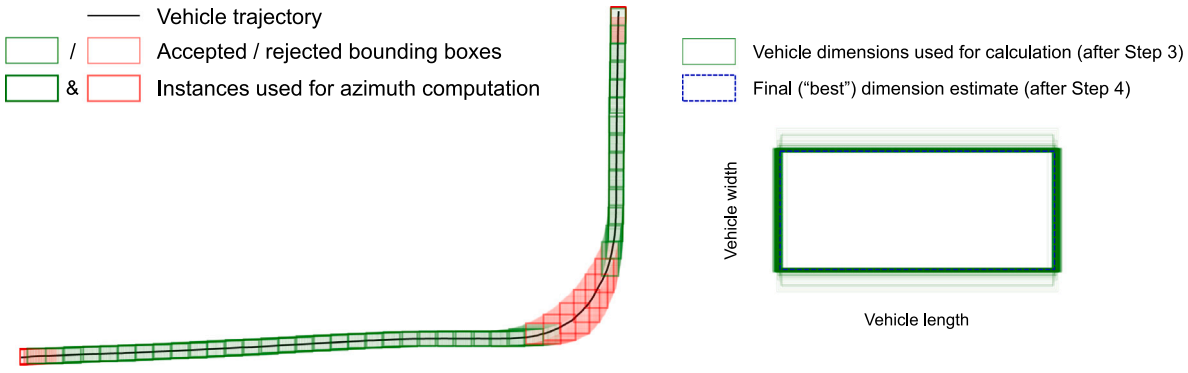


Fig. 12. Illustration of the vehicle dimension estimation process for moving vehicles.

Given these considerations, we selected an 8000×8000 pixel resolution for subsequent processing. This resolution not only facilitates the manual segmentation of road sections and lane delineations but also results in manageable orthophoto cut-out file sizes, enhancing dataset usability. With a mean reprojection error well below two pixels, our orthophoto registration method demonstrates high accuracy, validating the effectiveness of our georeferencing approach. Furthermore, the process of mapping from reference frames to master frames, an integral part of our method, is significantly faster and yields a notably higher average number of inliers, ranging from several hundreds to thousands, depending on the intersection.

3.7. Traffic data

3.7.1. Robust vehicle dimension estimation

Accurate estimation of vehicle dimensions from drone-based video imagery is vital for various traffic analysis applications, yet it remains an underrepresented challenge in scientific literature. To address this gap, we present a novel approach that leverages both un-stabilized and stabilized BBs from sequences $\mathcal{T}_v[k]$ and $\mathcal{T}_v^{\text{ref}}[k]$, respectively. Our unique methodology offers a significant advancement in the field, providing a level of accuracy and robustness previously unattained in similar studies.

Our approach employs a systematic process designed to enhance the robustness and accuracy of vehicle dimension estimates. The methodology focuses on fully visible BBs and vehicle motions parallel to the X or Y axes of the reference frame F_v^{ref}. The process encompasses visibility filtering, initial dimensions computation, azimuth-based filtering, final dimension computation, and real-world conversion. Additionally, we incorporate a solution for stationary vehicles, which is applicable only when these vehicles are aligned with the F_v^{ref} axis. Detailed implementation steps and experiment-specific parameters are provided in Appendix A.

Fig. 12 illustrates the BB filtering process, using actual BBs obtained from a sample trajectory. The figure also shows the final dimension estimate, computed as the first quartile of all "accepted" BBs. The right subfigure demonstrates the consistency of detected BBs along quasi-straight trajectories, with only minor inflations due to the permissible azimuth angle tolerance ($\bar{\theta} = 15°$) and deviations in road alignment relative to the reference drone's orientation.

It is important to acknowledge certain limitations of our approach. The dimensions derived from BBs in pixel coordinates may slightly overestimate the actual real-world vehicle sizes, often due to the BB encapsulating protruding parts, such as rear-view mirrors, or strong shadows. The method also faces challenges when vehicles travel exclusively on non-parallel roads or remain parked or stationary out of alignment with the image axes. In these cases, as well as when persistent partial occlusions distort the BBs, no dimension estimates are calculated.

Despite these challenges, our approach provides a robust solution for vehicle dimension estimation from drone footage. By addressing the complexities of real-world scenarios and incorporating solutions for both moving and stationary vehicles, this method contributes to advancing the field of traffic analysis in drone-based applications.

3.7.2. Speed and acceleration computation

To maintain the integrity of the extracted trajectories, we refrain from applying smoothing or interpolation techniques on the raw data, thereby preserving its original measurement characteristics. This approach allows users to select the most suitable post-processing methods for their specific needs. However, to facilitate efficient querying of trajectories based on predefined kinematics thresholds, we provide speed and acceleration estimates, which are computed after applying linear interpolation to fill in any potentially missing trajectory points, along with a simple smoothing technique to ensure reliable computation of the speed and acceleration profiles.

We apply a carefully tuned Gaussian filter to smooth the speed data, reducing noise while retaining critical dynamic patterns. The speed tuning process was guided by AV data (see Section 4.2) to ensure accurate real-world vehicle motion representation. The details of the vehicle kinematics computation and tuning are provided in Appendix B. The primary purpose of the smoothed speed and acceleration estimates is to facilitate efficient querying of trajectories based on kinematic thresholds, such as filtering out stationary vehicles. These estimates are not intended to represent definitive kinematic profiles for detailed traffic analysis.

Table 5
Descriptions of columns in the Songdo Traffic dataset, detailing their format, data type, and interpretation. Floating-point values are reported with specified decimal places (d.p.).

Dataset column name	Format/Units	Data type	Explanation
`Vehicle_ID`	1, 2, …	Integer	Unique vehicle identifier within each CSV file
`Local_Time`	`hh:mm:ss.sss`	String	Local Korean time (GMT+9) in ISO 8601 format
`Drone_ID`	1, 2, …, 10	Integer	Unique identifier for the drone capturing the data
`Ortho_X, Ortho_Y`	px (1 d.p.)	Float	Vehicle center coordinates in the orthophoto cut-out image
`Local_X, Local_Y`	m (2 d.p.)	Float	KGD2002/Central Belt 2010 planar coordinates (EPSG:5186)
`Latitude, Longitude`	° DD (7 d.p.)	Float	WGS84 geographic coordinates in decimal degrees (EPSG:4326)
`Vehicle_Length`[a], `Vehicle_Width`[a]	m (2 d.p.)	Float	Estimated physical dimensions of the vehicle
`Vehicle_Class`	Categorical (0–3)	Integer	Vehicle type: 0 (car/van), 1 (bus), 2 (truck), 3 (motorcycle)
`Vehicle_Speed`[a]	km/h (1 d.p.)	Float	Estimated speed computed from trajectory data using (B.1)
`Vehicle_Acceleration`[a]	m/s^2 (2 d.p.)	Float	Estimated acceleration derived from smoothed speed values
`Road_Section`[a]	`N_G`	String	Road section identifier (N = node, G = lane group)
`Lane_Number`[a]	1, 2, …	Integer	Lane position (1 = leftmost lane in the direction of travel)
`Visibility`	0/1	Boolean	1 = fully visible, 0 = partially visible in the camera frame

[a] These columns may be empty under certain conditions (see Section 3.6.1, Section 3.7.1, Section 3.7.2).

Our approach allows for future exploration of more advanced methods, such as those employed in Venthuruthiyil and Chunchu (2020). The availability of raw trajectory data in our dataset allows users to implement custom filtering techniques, such as the Kalman filter (Kalman, 1960), or the Rauch-Tung-Striebel (RTS) smoother (Rauch et al., 1965), tailored to their specific needs and applications.

3.7.3. Construction of the songdo traffic dataset

The Songdo Traffic dataset (Fonod et al., 2025a) is compiled from data collected by 10 drones monitoring 20 intersections over four days, encompassing 688,840 unique vehicle trajectories. These trajectories are preserved at the original camera capture rate of 29.97 FPS, ensuring high temporal resolution without downsampling. The extraction and processing of trajectories were conducted first on a per-cut-video basis. As detailed in Section 3.1, the video segmentation methodology systematically associates each video with its corresponding day, flight session, intersection, drone, and hovering sequence, resulting in over 3000 individual comma-separated values (CSV) files.

To streamline data management, trajectories were aggregated by intersection and flight session, with filenames encoding key metadata such as the collection date, intersection label, and session ID. For example, `2022-10-04_S_AM1.csv` contains all extracted trajectories from the first morning session of the first day at the intersection 'S'. A detailed description of the dataset columns is provided in Table 5. The aggregation process consolidated the dataset into 800 CSV files, representing the complete Songdo Traffic dataset. These files were then packaged into 80 ZIP archives, one per intersection and experiment day, resulting in a total compressed trajectory dataset size of 12.21 GB. Each ZIP archive contains all 10 flight sessions for a given intersection and day (e.g., `2022-10-04_S.zip`).

To minimize noise from detection anomalies, trajectories with 15 or fewer points are excluded. For uninterrupted tracking, this filtering corresponds to approximately 0.5 s of observation, ensuring that only meaningful motion patterns are retained. Additionally, a balance between file size and retained precision has been carefully considered: local and geographical coordinates are rounded to retain centimeter-level accuracy, while orthophoto coordinates are rounded to preserve sub-pixel granularity. For a detailed discussion of known dataset artifacts and limitations, see Appendix D.

Beyond trajectory data, Songdo Traffic also includes the 20 high-resolution orthophoto cut-outs (8000 × 8000 pixels), complemented by corresponding road section and lane segmentations. These cut-outs provide essential spatial context, enabling seamless integration of vehicle pixel coordinates with roadway geometry for enhanced visualization. Additionally, the dataset features 29 video samples, each capturing the first 60 s of drone activity over its designated intersections during the final session (PM5) of the last day (2022-10-07) of the experiment. These supplementary assets enhance the dataset's utility for detailed traffic analysis and visualization.

4. Results

4.1. Traffic data extraction

We present key characteristics of the georeferenced trajectories extracted using our complete methodology, focusing on the final day of the experiment (Friday) and the final afternoon session (PM5), which commenced at 17:40, representing one of the most congested periods. Specifically, we analyze intersection L, depicted in Fig. 13 (left), where the horizontal road has a speed limit of 30 km/h and the vertical road has a speed limit of 50 km/h. This intersection was monitored by both drones D6 and D7, however, we limit our analysis to D7, which maintained an altitude of 150 m.

During this period, D7 recorded a total of four video segments ($v \in \{\mathrm{L1}, \mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}$), each corresponding to an individual hovering, with a combined duration of approximately 6 min and 55 s. From these videos, we extracted 1,194 unique vehicle trajectories of varying lengths. The effectiveness of our detection and tracking approach is demonstrated in Fig. 13 (left), where

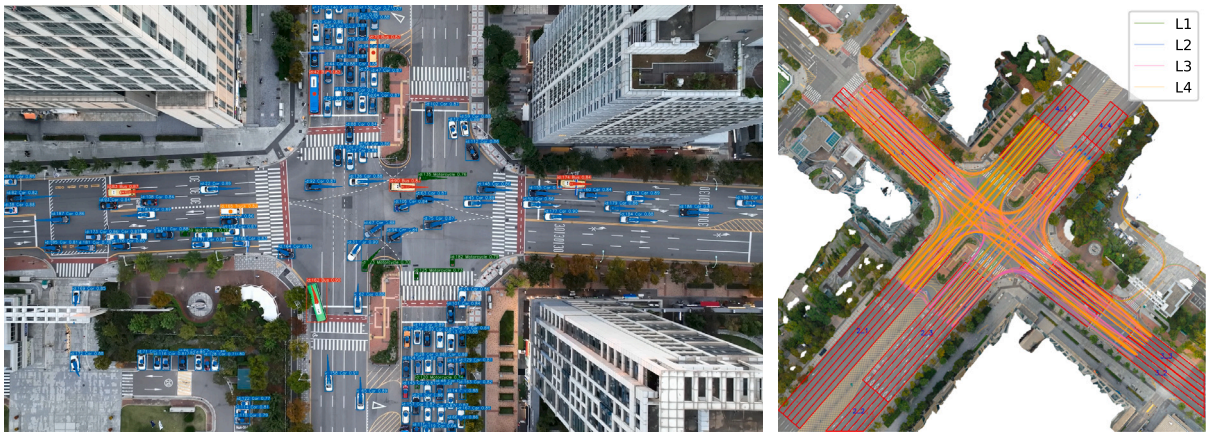


Fig. 13. Left: Illustration of vehicle detection and tracking. Right: Nearly 1,200 vehicle trajectories extracted from four separate drone hovering sessions, overlaid on the segmented orthophoto cut-out.

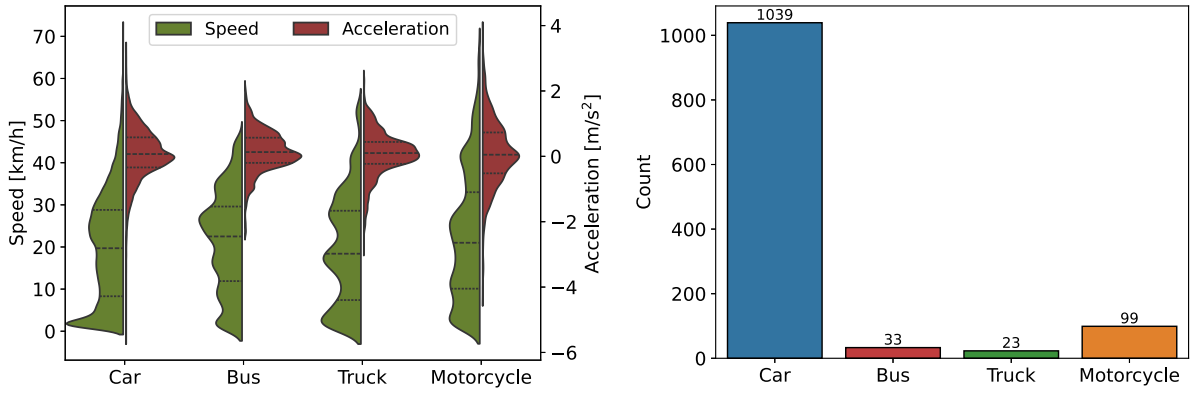


Fig. 14. Left: Speed and acceleration distributions for each vehicle class. Right: Distribution of vehicle classes.

color-coded BBs indicate accurate vehicle identification and classification: blue for cars, red for buses, orange for trucks, and green for motorcycles. The trailing tails represent the last second of vehicle movement, providing insights into speed and stopping behavior.

The aggregated trajectories in the orthophoto cut-out CS, shown in Fig. 13 (right), illustrate the ability of our framework to transform multiple separate video feeds into consistent georeferenced trajectories. This visualization highlights the robustness of our methodology for analyzing dynamic traffic flows in congested urban environments and emphasizes the contributions of the Songdo Traffic dataset in providing detailed insights into urban mobility.

Unlike previous UAV-based extraction methodologies, our approach yields realistic acceleration and speed distributions, as shown in Fig. 14 (left). Speed values are filtered to exclude those below 1 km/h for better plot legibility. The corresponding vehicle class distribution is shown in Fig. 14 (right). Notably, the speed distributions resemble realistic patterns for an intersection with 30 km/h and 50 km/h road limits, however, the significant congestion during this period results in limited probability mass near the 50 km/h mark. Additionally, the acceleration distributions align closely with those reported in fixed camera setups (Venthuruthiyil and Chunchu, 2020), reflecting the effectiveness of our track stabilization. In contrast, previous UAV-based trajectory extraction methods often exhibit large and unrealistic acceleration variances with heavy tails, often reaching up to ± 300 m/s^2, mainly due to camera shakiness and inadequate video stabilization (Venthuruthiyil and Chunchu, 2020).

The estimated vehicle dimensions are depicted in Fig. 15, which shows the length and width distributions for different vehicle classes. Notably, the median length and width for the car category are approximately 4.9 m and 2.1 m, respectively, which align closely with mean vehicle dimensions reported in previous studies. For example, a study on passenger car external dimensions found that the mean length of sedans is 5.1 m and the mean width is 1.89 m, while SUVs have an mean length of 4.95 m and an mean width of 1.94 m (Zhang et al., 2022c). The significant differences in length and width between buses and motorcycles also demonstrate the effectiveness of our dimension estimation approach.

To further validate our dimension estimation approach, we leveraged the AV experiment, which is detailed in Section 4.2. The AV used in this experiment is a Hyundai IONIQ 5 compact crossover SUV, with known dimensions of 4.635 m in length and 1.890 m in width (2.116 m with side-view mirrors). Its distinctive Stanford logo made it easily identifiable in our drone footage. Our estimates, based on 15 video observations (see Fig. 18), yielded a length distribution of 4.931 ± 0.225 m and a width distribution of 2.15 ± 0.072

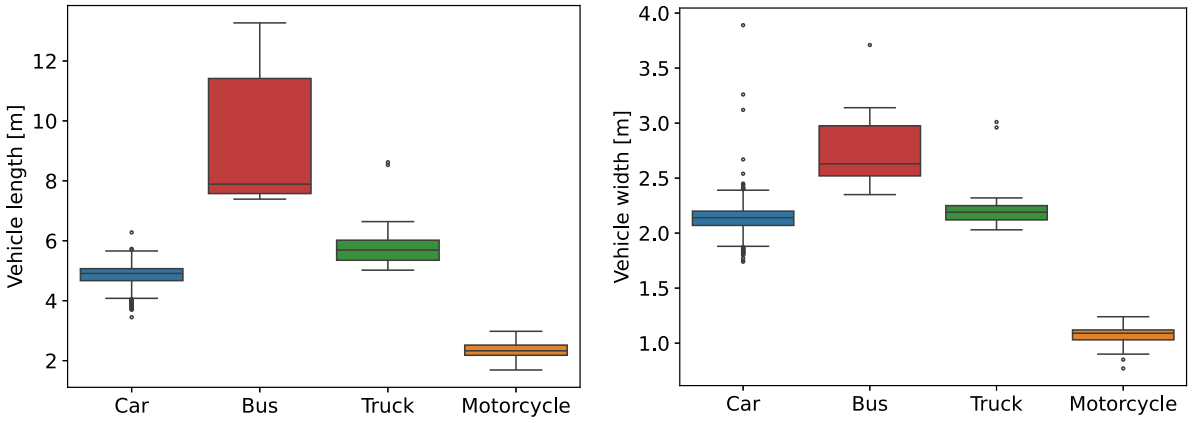


Fig. 15. Estimated real-world vehicle length and width distributions.

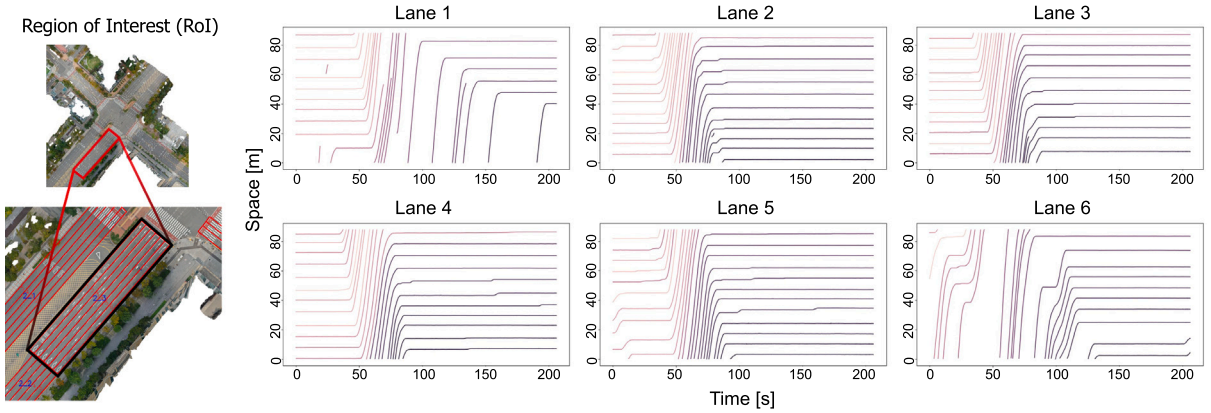


Fig. 16. spatio-temporal diagram example for intersection L, road segment 2_3 and lanes 1–6 within the depicted RoI.

m. While the width estimate is relatively accurate, length estimates were often inflated due to strong shadows present in the majority of videos. This is confirmed by the lower quartile of the length distribution (4.685 m), which is closer to the ground truth value (4.635 m).

We also conducted a similar analysis using a public transportation bus, specifically a Hyundai Super AeroCity model year 2018, which was easily identifiable in the drone footage due to its distinct color and shape. The bus, depicted in Fig. 13 (blue bus, top-center), has known dimensions of 10.955×2.490 m (excluding wing mirrors). From 12 videos, 6 from morning and 6 from afternoon sessions, captured at 2 intersections, we extracted 20 distinct instances of this bus and estimated their dimensions. Our results showed a length distribution of 11.540 ± 0.227 m and a width distribution of 3.063 ± 0.179 m, indicating that the estimates were slightly inflated due to the bus's protruding mirrors, shadows, and height-related perspective distortions. The wing mirrors, each 21 cm wide, likely contribute to the width discrepancy, bringing the total width to approximately 2.910 m, closer to our estimated width distribution.

To further assess our approach, we implemented a naive baseline that computes non-stationary vehicle dimensions by directly averaging the longer and shorter sides of all BBs, without any azimuth-based filtering. For the AV (Hyundai IONIQ 5) cases, where trajectories are mostly parallel to the image axes (see Fig. 18), the naive method estimated 5.005 ± 0.246 m (length) and 2.341 ± 0.223 m (width). This parallel alignment inherently favors the naive approach, yet the higher width variability indicates reduced reliability. A similar pattern emerged for the Hyundai Super AeroCity bus, where trajectories were also predominantly parallel, with the naive method estimating 11.585 ± 0.218 m (length) and 3.645 ± 0.416 m (width). The naive width estimate is significantly inflated compared to our results, reinforcing that width estimation is particularly sensitive without proper filtering. These findings suggest that while the naive approach may perform reasonably well for simple, axis-aligned trajectories, our method becomes crucial when dealing with more realistic scenarios involving frequent non-parallel vehicle motions.

The spatio-temporal diagram in Fig. 16 visualizes traffic flow dynamics at intersection L, highlighting the temporal progression of vehicle movements and queue formation and dissipation across six lanes. To maintain legibility, the diagram shows trajectories from a single video. Notable patterns include distinct queue buildup and clearance in the left-turn lane (Lane 1), consistent flow in straight-ahead lanes (Lanes 2–5), and dynamic stop-and-go behavior in the right-turn lane (Lane 6). Shorter trajectories, particularly in Lanes

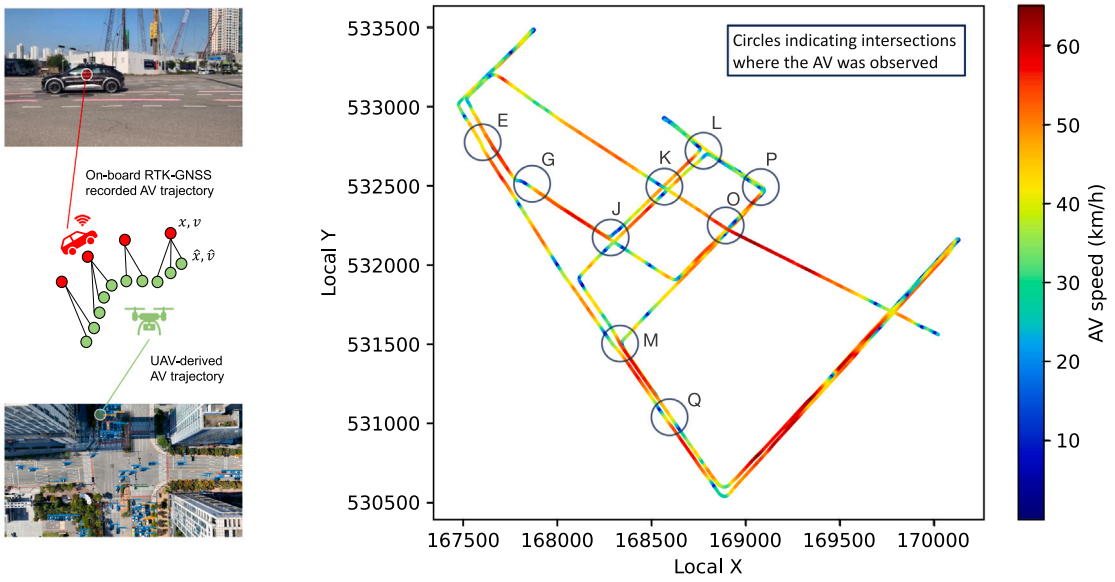


Fig. 17. Left: Illustration of the two AV trajectory measuring principles. Right: On-board RTK-GNSS recorded AV trajectory converted to local CS, with marked intersections where our drones recorded the AV passing.

1 and 2, correspond to motorcycles changing lanes or exiting the road section, causing intermittent traffic patterns. The inclusion of road section and lane-level information in our dataset enables users to create similar spatio-temporal analyses, facilitating granular studies of traffic behavior that are often limited by other datasets. A video demonstration of our extraction pipeline (Fonod, 2025a), *Geo-trax*, is available at https://youtu.be/gOGivL9FFLk.

4.2. Comparison with RTK-GNSS data from the probe AV

To assess the reliability of our extraction framework, we compared the drone-derived trajectories and speed profiles with those recorded by the RTK-GNSS sensor on the AV. Conducted by SCIGC, the AV experiment coincided with our monitoring campaign, providing a unique validation opportunity. This comparison allowed us to evaluate the accuracy of our framework against an independent measurement source (see Fig. 17, left).

The AV experiment commenced at 8:53:38 AM on the final day of our monitoring campaign and lasted approximately 1 h and 10 min. Fig. 17 (right) shows the AV's full trajectory and speed profile. Given the dynamic nature of our drone operations, which alternated over multiple intersections during each flight session, our analysis here focuses exclusively on static hoverings above intersections. Consequently, we observed the AV at nine intersections (see Fig. 17), recording a total of fifteen distinct AV passes, as illustrated in Fig. 18.

Unfortunately, the timestamps of the AV trajectory from the RTK-GNSS sensor were not synchronized with our UAVs' internal clocks. Additionally, the RTK-GNSS data was asynchronously sampled and undersampled, averaging 10 Hz, while our trajectories were extracted at a fixed 29.97 FPS rate. Due to these differences, we use separate time indices: k_1 for the RTK-GNSS data and k_2 for our UAV-derived trajectory. This discrepancy makes direct Euclidean distance comparisons infeasible, necessitating a more suitable error measurement method (Su et al., 2020).

In our methodology, we calculate the positional deviations between the two trajectories in the local CS by measuring the distance $d_P[k_1] \geq 0$. This distance represents the perpendicular separation from the RTK-GNSS trajectory point $P[k_1] = (x_L[k_1], y_L[k_1])$ to the line defined by the two closest trajectory points from our UAV-derived data. For a visual illustration of this computation, see Fig. 19.

The absolute positional deviation $d_P[k_1]$ at discrete time k_1 is computed as follows:

$$d_P[k_1] = \left| \frac{(\hat{P}^{\star 2} - \hat{P}^{\star 1}) \times (\hat{P}^{\star 1} - P[k_1])}{\|\hat{P}^{\star 2} - \hat{P}^{\star 1}\|} \right|, \tag{10}$$

where '×' denotes the cross product. Here, $\hat{P}^{\star 1}$ is the point on the UAV-derived trajectory that is closest to $P[k_1]$, and $\hat{P}^{\star 2}$ is the adjacent point to with the second shortest Euclidean distance to $P[k_1]$.

In a similar manner, we compute the speed difference $\Delta v[k_1]$ at k_1 as follows:

$$\Delta v[k_1] = v[k_1] - \left(w^{\star 1} \tilde{v}^{\star 1} + w^{\star 2} \tilde{v}^{\star 2} \right), \tag{11}$$

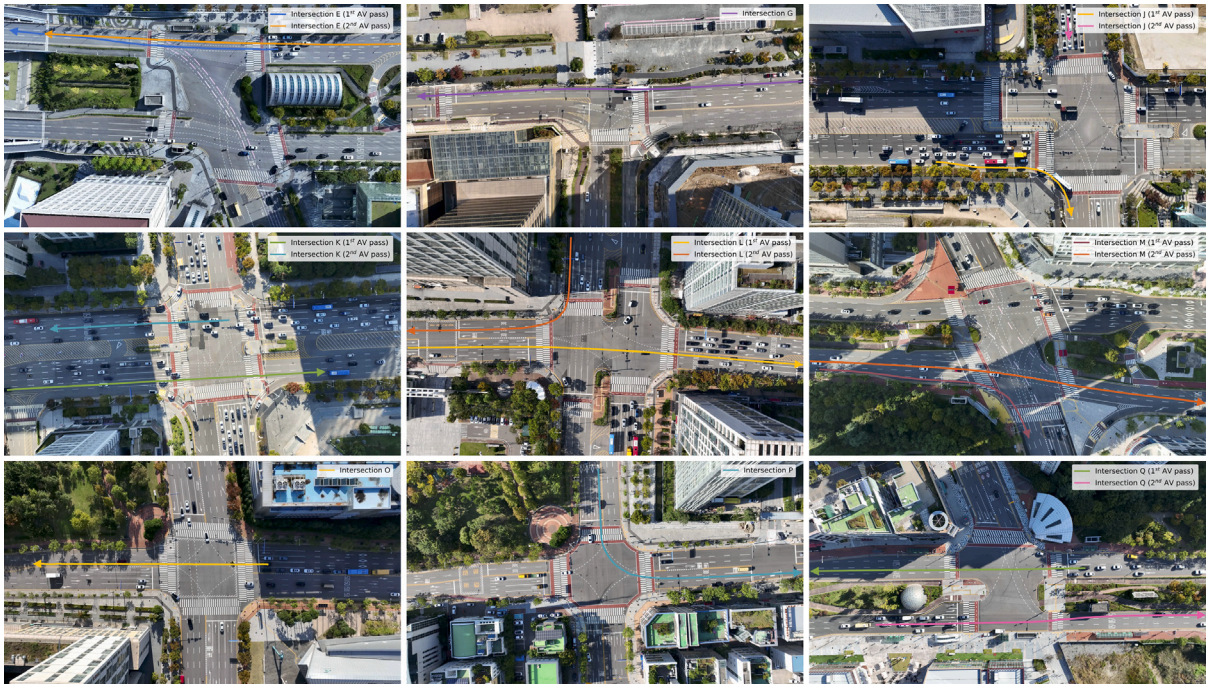


Fig. 18. Trajectories of the AV observed by our drones.

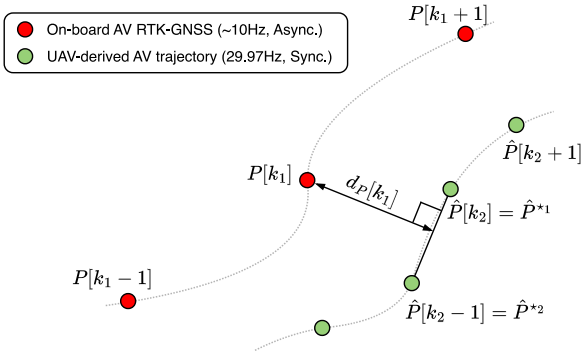


Fig. 19. Illustration of positional deviation determination between RTK-GNSS data point and UAV-derived trajectory.

where $\tilde{v}^{\star i}$ is the smoothed speed estimate associated with $\hat{P}^{\star i}$. The weights $0 \leq w^{\star i} \leq 1$ are given by: $w^{\star 1} = d^{\star 2}/(d^{\star 1} + d^{\star 2})$ and $w^{\star 2} = d^{\star 1}/(d^{\star 1} + d^{\star 2})$, where $d^{\star 1} = \|\hat{P}^{\star 1} - P[k_1]\|$ and $d^{\star 2} = \|\hat{P}^{\star 2} - P[k_1]\|$ are the Euclidean distances between the on-board measurement $P[k_1]$ and the UAV-derived points $\hat{P}^{\star 1}$ and $\hat{P}^{\star 2}$, respectively.

We applied the above methodology to compute d_P and Δv for each of the captured AV passes (illustrated in Fig. 18). To ensure accurate calculations, we segmented the AV videos to exclude any initial or final stationary periods, such that the start and end of each video segment capture the AV in motion. Additionally, we filtered out speed difference values from $\Delta v[k_1]$ that corresponded to $v[k_1] \leq 1$ to prevent bias in our results. Finally, we aggregated the computed differences for each intersection to facilitate a clear and concise presentation of our findings.

The comparison results are summarized in Table 6. Overall, the UAV-derived trajectories show strong agreement with RTK-GNSS measurements, with mean positional deviations remaining below 1 meter at most intersections, except for L and P. Similarly, speed differences between the two methods are minimal, averaging within ±1 km/h, except at intersection M, where a larger discrepancy likely reflects the high speed variability observed during both AV passes. The observed deviations represent differences rather than errors, as neither the UAV-derived data nor the RTK-GNSS measurements can be considered absolute ground truth. These differences may stem from variations in measurement methodologies, such as the location of the on-board RTK-GNSS sensor relative to the vehicle center derived from the BBs in our UAV footage. Additionally, potential inaccuracies in the RTK-GNSS data could be caused by signal reflections or obstructions from surrounding structures. A more detailed analysis of intersections L and P, including possible sources of RTK-GNSS inaccuracies, is provided later in this section. There could also be cumulative effects of small inaccuracies in

Table 6
Comparison of positional and speed differences (mean ± standard deviation) between drone-extracted trajectories and on-board RTK-GNSS measurements of the AV, along with aggregated trajectory statistics (length and duration).

Intersection label	Positional deviation (m)	Speed difference (km/h)	Trajectory length (m)	Trajectory duration (s)
E	0.459 ± 0.334	-0.555 ± 1.551	336.13	26.63
G	0.241 ± 0.073	-0.606 ± 0.646	190.41	15.70
J	0.494 ± 0.252	0.867 ± 1.314	99.78	20.20
K	0.326 ± 0.220	-0.127 ± 0.538	266.16	21.81
L	1.879 ± 0.330	-0.103 ± 0.928	331.22	35.27
M	0.558 ± 0.616	-1.083 ± 2.872	334.37	49.88
O	0.647 ± 0.138	-0.816 ± 1.182	123.47	14.00
P	1.134 ± 1.200	-0.545 ± 2.343	151.97	38.71
Q	0.382 ± 0.259	-0.198 ± 1.578	322.18	33.20

BB detection, object tracking, track stabilization, and georeferencing. Despite these potential sources of error, the high consistency between the two measurement methods strengthens confidence in the reliability of the Songdo Traffic dataset's extracted trajectories and speed estimates.

To investigate the elevated positional discrepancies at intersections L and P, we generated locally rectified experimental orthophotos using enhanced georeferencing techniques. These orthophotos were intended to examine and potentially rule out any inaccuracies in the original orthophoto. A DJI MAVIC 3 drone was employed to capture high-resolution, overlapping images from an altitude of 75 m. To ensure higher geospatial accuracy, four GCPs were strategically placed at each corner of intersections L and P, each marked with a 3×3 m QR code. These GCPs were precisely positioned using dual EMLID RS2+ RTK-GNSS receivers operating in FIX mode, providing centimeter-level accuracy, in contrast to the less precise SINGLE mode. The experimental orthophotos were then generated using Agisoft Metashape Pro software,[3] leveraging the improved accuracy and increased number of GCPs.

The newly generated orthophotos, intended to enhance georeferencing accuracy, resulted in only marginal reductions in positional discrepancies. The recalculated positional deviations were 1.176 ± 0.327 m for intersection L and 0.816 ± 0.539 m for intersection P, with corresponding speed differences of -0.359 ± 0.966 km/h and -0.475 ± 2.192 km/h, respectively. Considering the superior track stabilization performance (see Fig. 7) and orthophoto matching accuracy (see Table 4), these elevated discrepancies are unlikely to originate from errors in the original orthophoto. Instead, a closer examination of the urban environments surrounding intersections L and P suggests that signal multipath effects and occlusions are more plausible contributors. The presence of tall buildings around these intersections (see Fig. 18) is typical of densely built urban areas like Songdo and is well-documented to cause the "urban canyon effect", where GNSS accuracy is degraded due to signal reflections and obstructions from surrounding structures (Groves, 2011; Kaplan and Hegarty, 2017).

5. Conclusion

In this study, we introduced a comprehensive framework for extracting georeferenced vehicle trajectories from high-altitude drone imagery, addressing key challenges in urban traffic monitoring. Our approach integrates several novel contributions, including a tailored object detector optimized for high-altitude BEV perspectives, a unique track stabilization method utilizing detected vehicle bounding boxes as exclusion masks during image registration, and a georeferencing strategy based on orthophoto and master frame that enhances consistent alignment across multiple drone viewpoints. Additionally, we proposed a robust vehicle dimension estimation method leveraging bounding box dimensions and azimuth information derived from vehicle dynamics. A novel traffic-oriented evaluation metric was introduced to complement existing metrics, directly assessing the accuracy of vehicle alignment after transformation. These contributions collectively advance the accuracy, efficiency, and scalability of drone-based traffic monitoring systems. Our methodology demonstrated robust detection performance despite challenges posed by high-altitude perspectives. The innovative use of track stabilization after object tracking effectively circumvented limitations inherent in traditional video stabilization methods, enabling reliable extraction of vehicle trajectories and kinematics. The incorporation of GCPs within the orthophoto further enhanced georeferencing accuracy, critical for detailed traffic dynamics analysis.

The application of our methodology in a multi-drone experiment over 20 intersections in the Songdo International Business District resulted in two high-quality datasets: the Songdo Traffic dataset, comprising nearly 690,000 unique vehicle trajectories, and the Songdo Vision dataset, featuring over 5000 human-annotated frames for object detection. Notably, the Songdo Vision dataset fills a critical gap in existing datasets by providing detailed annotations for smaller vehicles, such as motorcycles, which are often challenging to detect from elevated BEV perspectives. Publicly releasing these datasets along with the extraction pipeline's complete source code sets new standards for data quality, reproducibility, and scalability, promoting collaboration and accelerating research innovation. Comparisons between drone-derived trajectories and high-precision sensor data from an instrumented probe vehicle further validate the consistency and accuracy of our extraction methods. Although the dataset may exhibit minor systematic positional biases, these discrepancies do not significantly hinder its applicability for a broad range of traffic analyses, including traffic

[3] https://www.agisoft.com/

flow analysis, congestion detection, speed profiling, and lane-level behavior studies. The provided road and lane segmentations, derived directly from visual data, further enhance the usability of the data set.

While our study makes significant contributions, we acknowledge certain limitations, including the exclusion of smaller, non-motorized road users and the manual effort required for video and road segmentations. Additionally, the vehicle classification granularity was constrained by practical annotation constraints, grouping certain vehicle types together. Future work will focus on refining classification schemes by differentiating trucks into subclasses and including smaller, non-motorized road users, automating segmentation tasks, and improving trajectory accuracy through rotated bounding boxes. Enhancements in video stabilization, such as deep learning-based feature matching, could further refine alignment accuracy. Additionally, exploring transformer-based detection models and integrating motion forecasting modules may enhance tracking robustness in complex urban settings. By combining drone-based monitoring with advanced computer vision, our approach provides a scalable and cost-effective solution for urban traffic analysis, contributing to data-driven transportation planning and intelligent mobility systems.

CRediT authorship contribution statement

Robert Fonod: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Haechan Cho:** Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Hwasoo Yeo:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Nikolas Geroliminis:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Acknowledgments

This work was supported by research grants from the Swiss National Science Foundation under NCCR Automation (No. 51NF40_180545, N.G. and R.F.), Innosuisse (No. 101.645 IP-ENG, N.G. and R.F.), the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022R1A2C1012380, H.Y. and H.C.), and the Open Research Data (ORD) Program of the ETH Board (R.F.). The authors express their gratitude to Sohyeong Kim, Yura Tak, and Weijiang Xiong from EPFL for their invaluable assistance in the data-wrangling process, Artem Vasilev for his dedicated efforts in data annotation, Jasso Espadaler Clapés for his insightful contributions to vehicle dimension estimation, and Muhammad Ahmed for his support in creating the experimental orthophoto for validation purposes and his contributions to road segmentation. We also thank DroMii company for producing the primary orthophoto and Stanford Center at the Incheon Global Campus (SCIGC) for providing the on-board recorded trajectories of their autonomous vehicle. We also thank the research teams of Prof. Simon Oh (Korea University) and Prof. Minju Park (Hannam University) for their assistance during the data collection campaign, including the provision of drone equipment and student support.

Appendix A. Vehicle dimension estimation steps and implementation details

For ease of notation, the sequentially ordered BBs for a vehicle with identifier "id" in video v are denoted as ($x^{\mathrm{id}}[k]$, $y^{\mathrm{id}}[k]$, $w^{\mathrm{id}}[k]$, $h^{\mathrm{id}}[k]$), where k ranges from 1 to N_v^{id}. These values, un-normalized by the image dimensions w_I and h_I, are used to calculate the vehicle's length (L^{id}) and width (W^{id}) in meters through a structured five-step process:

Step 1 - Visibility filtering: This initial step filters out BBs that are not fully visible within the frame, focusing on fully observable vehicles. We define a visibility set $\mathcal{V}^{\mathrm{id}}$ containing indices k of BBs satisfying:

$$x^{\mathrm{id}}[k] - \frac{w^{\mathrm{id}}[k]}{2} > \varepsilon, \quad x^{\mathrm{id}}[k] + \frac{w^{\mathrm{id}}[k]}{2} < w_I - (\varepsilon + 1), \quad y^{\mathrm{id}}[k] - \frac{h^{\mathrm{id}}[k]}{2} > \varepsilon, \quad y^{\mathrm{id}}[k] + \frac{h^{\mathrm{id}}[k]}{2} < h_I - (\varepsilon + 1), \tag{A.1}$$

where $\varepsilon > 0$ is a margin to keep BBs well within the frame's boundaries. The detection system ensures that BBs dimensions do not exceed the frame limits, with pixel coordinates ranging from 0 to $w_I - 1$ horizontally and $h_I - 1$ vertically. The margin ε helps filter out transient BBs generated as vehicles enter or exit the frame, as the detector can identify partially visible vehicles, leading to variable BB sizes due to vehicles' gradual appearance or disappearance at the frame boundaries.

Step 2 - Initial dimensions computation: Using the remaining BBs after Step 1, we curate the set of instantaneous vehicle lengths $\mathcal{L}^{\mathrm{id}}$ and widths $\mathcal{W}^{\mathrm{id}}$ according to the following simple rule:

$$\begin{aligned} \mathcal{L}^{\mathrm{id}} &\triangleq \left\{ L^{\mathrm{id}}_{\inf \mathcal{V}^{\mathrm{id}}}, \dots, L^{\mathrm{id}}_{\sup \mathcal{V}^{\mathrm{id}}} \right\} = \left\{ \max\left(w^{\mathrm{id}}[k],\ h^{\mathrm{id}}[k] \right) \mid k \in \mathcal{V}^{\mathrm{id}} \right\}, \\ \mathcal{W}^{\mathrm{id}} &\triangleq \left\{ W^{\mathrm{id}}_{\inf \mathcal{V}^{\mathrm{id}}}, \dots, W^{\mathrm{id}}_{\sup \mathcal{V}^{\mathrm{id}}} \right\} = \left\{ \min\left(w^{\mathrm{id}}[k],\ h^{\mathrm{id}}[k] \right) \mid k \in \mathcal{V}^{\mathrm{id}} \right\}. \end{aligned} \tag{A.2}$$

Step 3 - Azimuth-based filtering: We refine the sets $\mathcal{L}^{\mathrm{id}}$ and $\mathcal{W}^{\mathrm{id}}$ using the vehicle's azimuth estimate $\theta_k^{\mathrm{id}} \in [0,\ 2\pi)$, calculated asynchronously from trajectory points in the CS of the reference frame F_v^{ref}, as follows:

$$\theta_k^{\mathrm{id}} = \operatorname{atan2}\left(y^{\mathrm{id}}[\delta_{k-1}^{\mathrm{id}}] - y^{\mathrm{id}}[\delta_k^{\mathrm{id}}],\ x^{\mathrm{id}}[\delta_k^{\mathrm{id}}] - x^{\mathrm{id}}[\delta_{k-1}^{\mathrm{id}}] \right), \quad k = 1, 2, \dots, \tag{A.3}$$

where $\delta_k^{\mathrm{id}} > 0$ is a discrete-time window growing recursively:

$$\delta_k^{\mathrm{id}} = \delta_{k-1}^{\mathrm{id}} + \inf_{\delta \in \mathbb{N}} \left\{ \delta \,\middle|\, \sqrt{\left(\Delta_x^{\mathrm{id}}(\delta_{k-1}^{\mathrm{id}}, \delta) \right)^2 + \left(\Delta_y^{\mathrm{id}}(\delta_{k-1}^{\mathrm{id}}, \delta) \right)^2} \geq r_{\mathrm{px}},\ \delta_{k-1}^{\mathrm{id}} + \delta \leq \sup(\mathcal{V}^{\mathrm{id}}) \right\}. \tag{A.4}$$

Here, $\Delta_x^{\mathrm{id}}(\delta_{k-1}^{\mathrm{id}}, \delta) = x^{\mathrm{id}}[\delta_{k-1}^{\mathrm{id}} + \delta] - x^{\mathrm{id}}[\delta_{k-1}^{\mathrm{id}}]$ and $\Delta_y^{\mathrm{id}}(\delta_{k-1}^{\mathrm{id}}, \delta) = y^{\mathrm{id}}[\delta_{k-1}^{\mathrm{id}} + \delta] - y^{\mathrm{id}}[\delta_{k-1}^{\mathrm{id}}]$ represent the distances traveled in the x and y directions, respectively, for a step δ. The initial index $\delta_0^{\mathrm{id}} = \inf(\mathcal{V}^{\mathrm{id}})$ corresponds to the time index of the 1st visible BB, and $r_{\mathrm{px}} > 0$ denotes the radius in pixels regulates the frequency of azimuth computations.

To provide a physical interpretation of r_{px}, we define $r_{\mathrm{px}} = r_{\mathrm{m}}/\mathrm{GSD}$, where GSD is the ground distance per pixel (meters/pixel) and $r_{\mathrm{m}} > 0$ is the minimum distance (meters) required for azimuth computation. This ensures azimuth calculations rely on significant vehicle movements, avoiding inaccuracies during stationary periods. GSD can be calculated using camera specifications and flight altitude or derived from the orthophoto.

If at least one azimuth is available θ_k^{id} $(k \geq 1)$, we further refine $\mathcal{L}^{\mathrm{id}}$ and $\mathcal{W}^{\mathrm{id}}$ using all available θ_k^{id} as follows:

$$\begin{aligned} \mathcal{L}^{\mathrm{id}} &= \left\{ L_i^{\mathrm{id}} \mid L_i^{\mathrm{id}} \in \mathcal{L}^{\mathrm{id}},\ i \in [\delta_{k-1}^{\mathrm{id}}, \delta_k^{\mathrm{id}}) \text{ if } |\theta_k - \phi_j| \leq \bar{\theta},\ \text{for any } \phi_j \in \boldsymbol{\Phi} \right\}, \\ \mathcal{W}^{\mathrm{id}} &= \left\{ W_i^{\mathrm{id}} \mid W_i^{\mathrm{id}} \in \mathcal{W}^{\mathrm{id}},\ i \in [\delta_{k-1}^{\mathrm{id}}, \delta_k^{\mathrm{id}}) \text{ if } |\theta_k - \phi_j| \leq \bar{\theta},\ \text{for any } \phi_j \in \boldsymbol{\Phi} \right\}, \end{aligned} \tag{A.5}$$

where $\boldsymbol{\Phi} = \{0,\ \pi/2,\ \pi,\ 3\pi/2,\ 2\pi\}$ is a set of cardial directions and $\bar{\theta} > 0$ is the maximum azimuth deviation allowed. The parameter $\bar{\theta}$ accounts for vehicle orientations that are non-parallel to the image axes, with a smaller $\bar{\theta}$ indicating more stringent filtering, thus ensuring only representative vehicle dimensions are considered.

When no azimuth data is available (e.g., stationary or parked vehicles), we filter BBs exhibiting square-like proportions, as these often reflect non-parallel parking or partial occlusions that compromise accuracy. This is achieved by a ratio-based criterion:

$$\mathcal{L}^{\mathrm{id}} = \left\{ L_i^{\mathrm{id}} \mid L_i^{\mathrm{id}} \in \mathcal{L}^{\mathrm{id}},\ \frac{L_i^{\mathrm{id}}}{W_i^{\mathrm{id}}} \geq \kappa_c \right\}, \quad \mathcal{W}^{\mathrm{id}} = \left\{ W_i^{\mathrm{id}} \mid W_i^{\mathrm{id}} \in \mathcal{W}^{\mathrm{id}},\ \frac{L_i^{\mathrm{id}}}{W_i^{\mathrm{id}}} \geq \kappa_c \right\}. \tag{A.6}$$

Here, $\kappa_c > 1$ is a class-specific threshold for the degree of non-squareness, with ratios $L_i^{\mathrm{id}}/W_i^{\mathrm{id}}$ approaching 1 indicating square-like dimensions, which are atypical in the Songdo footage and are therefore flagged as unreliable.

Step 4 - Final dimension computation: After implementing Steps 1 to 3, if the sets $\mathcal{L}^{\mathrm{id}}$ and $\mathcal{W}^{\mathrm{id}}$ are not empty, the unique vehicle dimensions in pixels are determined as follows:

$$L_{\mathrm{px}}^{\mathrm{id}} = \mathrm{Q}_{\mathrm{lower}}\left(\mathcal{L}^{\mathrm{id}}\right), \quad W_{\mathrm{px}}^{\mathrm{id}} = \mathrm{Q}_{\mathrm{lower}}\left(\mathcal{W}^{\mathrm{id}}\right). \tag{A.7}$$

Here, $\mathrm{Q}_{\mathrm{lower}}(\cdot)$ computes the first quartile, indicating the value below which 25% of observations fall. This measure is preferred due to the leniency of $\bar{\theta}$, resulting in dimensions that meet or exceed the ground truth, while mean or median methods often yield overly conservative estimates. Minimum values are avoided to prevent errors from partial occlusions.

Step 5 - Real-world conversion: In this final step, the vehicle dimensions $L_{\mathrm{px}}^{\mathrm{id}}$ and $W_{\mathrm{px}}^{\mathrm{id}}$ are converted from pixels to meters by placing a hypothetical BB at the center of the video frame. Centering the BB leverages the higher precision of the orthophoto, enhanced by strategically placed GCPs at intersection edges. The conversion uses a three-point BB decomposition defined as:

$$p_{\mathrm{px1}}^{\mathrm{id}} = \tfrac{1}{2}(w_I,\ h_I), \quad p_{\mathrm{px2}}^{\mathrm{id}} = \tfrac{1}{2}(w_I,\ h_I + W_{\mathrm{px}}^{\mathrm{id}}), \quad p_{\mathrm{px3}}^{\mathrm{id}} = \tfrac{1}{2}(w_I + L_{\mathrm{px}}^{\mathrm{id}},\ h_I). \tag{A.8}$$

These points are transformed to local Cartesian coordinates using $H_v^{\mathrm{ref}\to\mathrm{ortho}}$ and $T_i^{\mathrm{ortho}\to\mathrm{world}}$ (see Section 3.6) yielding real-world coordinates p_1^{id}, p_2^{id}, and p_3^{id}. Finally, the real-world vehicle dimensions L^{id} and W^{id} are determined using appropriate geometric relations:

$$L^{\mathrm{id}} = 2\|p_3^{\mathrm{id}} - p_1^{\mathrm{id}}\|, \quad W^{\mathrm{id}} = 2\|p_2^{\mathrm{id}} - p_1^{\mathrm{id}}\|, \tag{A.9}$$

where $\|\cdot\|$ is the Euclidean norm of a vector, effectively measuring the physical distances between the points.

All steps outlined above should employ the unstabilized BBs from $\mathcal{T}_v$, with the sole exception of the azimuth angle calculation, θ_k^{id}, which should be based on the stabilized vehicle trajectories contained in $\mathcal{T}_v^{\mathrm{ref}}$.

Implementation details

In our experiment, we set $\varepsilon = 4$, $\bar{\theta} = 15°$, $r_{\mathrm{m}} = 1.25$, and derived GSD = 0.02725 based on camera specifications and an average altitude of 145 m, resulting in $r_{\mathrm{px}} \approx 45.87$. The calibration of r_{m} is essential for effective azimuth-based filtering and is primarily influenced by the quality of the track stabilization process. r_{m} must be sufficiently large to avoid triggering new azimuth computations from drone movement artifacts while remaining small enough to capture azimuth updates during brief intervals of vehicle alignment with the image axes, particularly during road turns. This calibration is essential for vehicles on roads misaligned with the image axes due to infrastructure layout or drone orientation. Accurate dimension estimation can be achieved if, during brief moments of alignment, an azimuth satisfying $\bar{\theta}$ is computed.

The class-dependent threshold parameters κ_c, used in (A.6) to filter dimensions of stationary vehicles, were derived by applying Q_{lower} to the empirical distributions of class-filtered L/W ratios from moving vehicles. Extensive trajectory data analysis yielded specific thresholds for each vehicle category: $\kappa_{\mathrm{car}} = 1.83$, $\kappa_{\mathrm{bus}} = 2.85$, $\kappa_{\mathrm{truck}} = 1.7$, and $\kappa_{\mathrm{motorcycle}} = 1.8$. To obtain reliable estimates of L and W, we enforced stricter azimuth angle constraints ($\bar{\theta} = 5°$) and set κ_c to ∞ to retain BBs only from vehicles perfectly aligned with the reference frame F_v^{ref}.

Appendix B. Vehicle kinematics computation and tuning

The raw speed of a vehicle with identifier id at time k is computed as the Euclidean distance between two consecutive interpolated trajectory points $(x_L^{\mathrm{id}}[k],\, y_L^{\mathrm{id}}[k])$ and $(x_L^{\mathrm{id}}[k-1],\, y_L^{\mathrm{id}}[k-1])$ in the local coordinate system, divided by the time step $\Delta t = t_k - t_{k-1}$, which is equal to FPS^{-1} for the processed video v.

To smooth the speed data $v^{\mathrm{id}}[k]$, we apply a Gaussian filter through a discrete convolution with a 1D Gaussian kernel:

$$\tilde{v}^{\mathrm{id}}[k] = \sum_{i=-M}^{M} v^{\mathrm{id}}[r(k+i)] \cdot g_\sigma[i], \qquad k \in \{2, \dots, N_v^{\mathrm{id}}\}, \tag{B.1}$$

where $g_\sigma[i] = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{i^2}{2\sigma^2}\right)$, $\sigma > 0$ is the standard deviation, and M is set as 3σ to encompass the significant range of the Gaussian function $g_\sigma[i]$. The reflect function $r(j)$ handles boundary conditions by mirroring the data at the edges:

$$r(j) = j + 2(N_v^{\mathrm{id}} - j) \cdot \mathbf{1}(j > N_v^{\mathrm{id}}) - 2(j-1) \cdot \mathbf{1}(j < 1), \tag{B.2}$$

where the indicator function $\mathbf{1}(\cdot)$ effectively reflects indices when j lies outside the range $[1,\, N_v^{\mathrm{id}}]$.

The acceleration profile $a^{\mathrm{id}}[k]$ is computed by taking the backward difference of consecutive smoothed speed values: $a^{\mathrm{id}}[k] = (\tilde{v}^{\mathrm{id}}[k] - \tilde{v}^{\mathrm{id}}[k-1])/\Delta t$, for $k \in 3, \dots, N_v^{\mathrm{id}}$. Only indices that satisfy $k \in \mathcal{V}^{\mathrm{id}}$ retain $\tilde{v}^{\mathrm{id}}[k]$ and $a^{\mathrm{id}}[k]$ in the dataset, where $\mathcal{V}^{\mathrm{id}}$ is the visibility set defined in (A.1). This ensures that kinematic computations are unaffected by changes in BBs as vehicles enter or exit the camera's FoV.

The standard deviation σ of the Gaussian kernel is the only tuning parameter for smoothing speed data. To determine its optimal value, we used AV data to assess the relationship between different σ values and the speed estimation errors $\Delta v[k_1]$ from (11). Specifically, we computed $\Delta v[k_1]$, applied the absolute value, and averaged these absolute errors. This analysis showed that the mean optimal σ across all intersections is 14.5, while the weighted average, accounting for the lengths of observed AV trajectories, is 14.49. Additionally, the optimal σ that minimizes the error variance was found to be 14.17. Therefore, we set σ to 14 frames, equivalent to 0.467 s at 29.97 FPS, to balance smoothing and detail retention. Notably, the Savitzky–Golay filter (Savitzky and Golay, 1964b) produced comparable results to our Gaussian filter approach.

Appendix C. Post-processing validation, corrections, and potential systematic biases

In addition to the manual data wrangling steps detailed in Section 3.1, automated post-processing checks were conducted to ensure data consistency and correct any remaining video or flight-log anomalies in the Songdo Traffic dataset. These analyses examined flight logs and extracted trajectories to assess drone hovering stability (location and altitude), abrupt camera parameter variations (focal length, ISO, shutter speed, f-number, and color temperature), timestamp continuity, and outlier kinematic values.

Approximately 2% of segmented videos exhibited anomalies such as significant drone movements or zoom changes undetected during data wrangling, timestamp irregularities, or data recording gaps. To correct these, affected videos and flight logs were re-cut by trimming the beginning or end, or in some cases, splitting the video into two segments to remove anomalies that occurred mid-sequence. The latter approach, applied to approximately 1% of segmented videos, inevitably resulted in trajectory fragmentation and vehicle ID resets. These corrections ensured that the final dataset is free from major recording inconsistencies, preserving the integrity of extracted trajectories and associated metadata.

As discussed in Section 3.6.2, the use of the master frame as an intermediary coordinate system implies that any potential inaccuracies in georeferencing would manifest as systematic biases consistently affecting all trajectories at a given intersection. Such systematic biases can potentially be corrected post hoc, for instance, through map-matching techniques or recalibration of the orthophoto mapping parameters. The retention of vehicle trajectories in orthophoto cut-out coordinates, along with the release of these cut-outs, facilitates such corrections if required. Additionally, our object detector exhibited high localization accuracy, with BB center deviations averaging 2.21 pixels from human annotations. This reinforces the reliability of the initial detection step, which serves as the foundation for trajectory extraction, independent of any downstream georeferencing or stabilization effects.

Appendix D. Known dataset artifacts and limitations

In line with best practices for dataset transparency and reproducibility, this appendix summarizes known artifacts, limitations, and potential sources of error identified in the Songdo Traffic dataset (Fonod et al., 2025a).

(1) Trajectory fragmentation: As discussed in Section 3.2 and Section 3.4, trajectories may be fragmented for certain motorcycle instances over complex underrepresented road infrastructures (e.g., pedestrian crossings, bicycle lanes, traffic signals), see Fig. 3 (right). Similarly, some truck variants, such as exceptionally large or specialized trucks that are extremely underrepresented in both the training data and the trajectory dataset, may yield split trajectories or abnormal kinematic profiles, owing to the object detector's challenges in consistently providing accurate BB detections. Users are advised to exercise caution when interpreting trajectory data associated with these specific vehicle instances. Additional fragmentations resulted from necessary mid-hovering video splits during data wrangling (cf. Section 3.1) and post-processing corrections (cf. Appendix C) when drones experienced technical issues, naturally creating divided trajectories.

(2) Vehicle ID ambiguities: The largest `Vehicle_ID` in a given CSV file does not necessarily represent the actual number of unique vehicles observed. This is due to (a) the aforementioned trajectory fragmentation, where a single vehicle's trajectory is split

into multiple segments, each potentially being assigned a different `Vehicle_ID`, (b) the removal of trajectories with 15 or fewer points after the tracking algorithm assigned unique vehicle IDs (cf. Section 3.7.3), and (c) instances of the same vehicle may be observed simultaneously by multiple drones, leading to distinct `Vehicle_IDs`. While flight plans were carefully designed to avoid overlapping observations of the same intersection by multiple drones, minor synchronization deviations may have led to occasional overlaps. Nevertheless, the recorded flight logs indicate that all drone flights commenced as planned, with 31 flights (out of 400) concluding slightly after their planned time window, with an average delay of 19.04 ±5.8 s.

(3)) Kinematic estimation limitations: The speed and acceleration variables in the dataset are derived from raw trajectory data and are subject to inaccuracies due to imperfect and dynamically varying BB detections, particularly from shadows and the expansion or contraction of BBs as vehicles enter or exit the video frame. While the latter effect is indicated by the visibility flag, shadow-induced variations are not explicitly annotated. Additionally, suboptimal track stabilization and applied interpolation and smoothing (cf. Section 3.7.2) may further affect accuracy. Consequently, these kinematic estimates are intended primarily for preliminary filtering rather than precise dynamic analyses.

(4) Vehicle dimension estimation reliability: As discussed in Section 3.7.1, vehicle dimension estimates may be unreliable, especially when a vehicle is stationary or not moving parallel to one of the image axes. Additionally, these estimates may systematically overstate actual dimensions due to BBs capturing protruding parts or shadows.

(5) Road section and lane assignment inaccuracies: Due to perspective effects, vehicles with significant heights, such as trucks or buses, may have their centers fall outside the assigned section or lane in the orthophoto, leading to potential mislabeling. This limitation should be considered when analyzing lane-level traffic flows for taller vehicles.

(6) Occasional pedestrian pair misclassifications: In rare cases, two pedestrians walking side by side across a crosswalk may be momentarily misclassified as a motorcycle. Such occurrences are extremely brief and, if present, are likely filtered out by the short trajectory filter discussed earlier (cf. Section 3.7.3).

Data availability

The SongdoTraffic and SongdoVision datasets are publicly available for download from Fonod et al. (2025a) and Fonod et al. (2025b), respectively, under a CC-BY-4.0 license. These datasets will also be listed on https://open-traffic.epfl.ch. The complete source code for the trajectory extraction framework, including the stabilization library and the associated optimization tool, is archived in Fonod (2025a), Fonod (2025b), and Fonod (2025c), respectively, and is maintained on GitHub under the MIT license.

References

Abdel-Aty, M., Wang, Z., Zheng, O., Abdelraouf, A., 2023. Advances and applications of computer vision techniques in vehicle trajectory generation and surrogate traffic safety indicators. Accid. Anal. Prev. 191, 107191. http://dx.doi.org/10.1016/j.aap.2023.107191.

Adaimi, G., Kreiss, S., Alahi, A., 2023. Traffic perception from aerial images using butterfly fields. Transp. Res. Part C: Emerg. Technol. 153, 104181. http://dx.doi.org/10.1016/j.trc.2023.104181.

Aharon, N., Orfaig, R., Bobrovsky, B.-Z., 2022. Bot-SORT: Robust associations multi-pedestrian tracking. arXiv arXiv:2206.14651v2.

Akyon, F.C., Altinuc, S.O., Temizel, A., 2022. Slicing aided hyper inference and fine-tuning for small object detection. In: 2022 IEEE International Conference on Image Processing. ICIP, IEEE, pp. 966–970. http://dx.doi.org/10.1109/ICIP46576.2022.9897990.

Alcantarilla, P.F., Bartoli, A., Davison, A.J., 2012. KAZE features. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12. Springer, pp. 214–227. http://dx.doi.org/10.1007/978-3-642-33783-3_16.

Alcantarilla, P.F., Solutions, T., 2013. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: British Machine Vision Conference, BMVC. BMVA Press, pp. 1281–1298. http://dx.doi.org/10.5244/C.27.13.

Arandjelović, R., Zisserman, A., 2012. Three things everyone should know to improve object retrieval. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2911–2918. http://dx.doi.org/10.1109/CVPR.2012.6248018.

Barber, D.B., Redding, J.D., McLain, T.W., Beard, R.W., Taylor, C.N., 2006. Vision-based target geo-location using a fixed-wing miniature air vehicle. J. Intell. Robot. Syst. 47, 361–382. http://dx.doi.org/10.1007/s10846-006-9088-7.

Barmpounakis, 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. http://dx.doi.org/10.1016/j.trc.2019.11.023.

Barmpounakis, E., Montesinos-Ferrer, M., Gonzales, E.J., Geroliminis, N., 2021. Empirical investigation of the emission-macroscopic fundamental diagram. Transp. Res. Part D: Transp. Environ. 101, 103090. http://dx.doi.org/10.1016/j.trd.2021.103090.

Barmpounakis, E.N., Vlahogianni, E.I., Golias, J.C., Babinec, A., 2019. How accurate are small drones for measuring microscopic traffic parameters? Transp. Lett. 11 (6), 332–340. http://dx.doi.org/10.1080/19427867.2017.1354433.

Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B., 2016. Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing. ICIP, IEEE, pp. 3464–3468. http://dx.doi.org/10.1109/ICIP.2016.7533003.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M., 2020. YOLOv4: Optimal speed and accuracy of object detection. http://dx.doi.org/10.48550/arXiv.2004.10934, arXiv preprint arXiv:2004.10934.

Bouguettaya, A., Zarzour, H., Kechida, A., Taberkit, A.M., 2021. Vehicle detection from UAV imagery with deep learning: A review. IEEE Trans. Neural Netw. Learn. Syst. 33 (11), 6047–6067. http://dx.doi.org/10.1109/TNNLS.2021.3080276.

Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A., 2020. Albumentations: fast and flexible image augmentations. Information 11 (2), 125. http://dx.doi.org/10.3390/info11020125.

Butilă, E.V., Boboc, R.G., 2022. Urban traffic monitoring and analysis using unmanned aerial vehicles (UAVs): A systematic literature review. Remote. Sens. 14 (3), 620. http://dx.doi.org/10.3390/rs14030620.

Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K., 2023. Observation-centric SORT: Rethinking SORT for robust multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9686–9696. http://dx.doi.org/10.1109/CVPR52729.2023.00934.

Cao, X., Wu, C., Lan, J., Yan, P., Li, X., 2011. Vehicle detection and motion analysis in low-altitude airborne video under urban environment. IEEE Trans. Circuits Syst. Video Technol. 21 (10), 1522–1533. http://dx.doi.org/10.1109/TCSVT.2011.2162274.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: Computer Vision – ECCV 2020. Springer, pp. 213–229. http://dx.doi.org/10.1007/978-3-030-58452-8_13.

Chen, A.Y., Chiu, Y.-L., Hsieh, M.-H., Lin, P.-W., Angah, O., 2020. Conflict analytics through the vehicle safety space in mixed traffic flows using UAV image sequences. Transp. Res. Part C: Emerg. Technol. 119, 102744. http://dx.doi.org/10.1016/j.trc.2020.102744.

Choi, J.-H., Lee, K.-H., Cha, K.-C., Kwon, J.-S., Kim, D.-W., Song, H.-K., 2006. Vehicle tracking using template matching based on feature points. In: 2006 IEEE International Conference on Information Reuse & Integration. IEEE, pp. 573–577. http://dx.doi.org/10.1109/IRI.2006.252477.

Coifman, B., Beymer, D., McLauchlan, P., Malik, J., 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. Transp. Res. Part C: Emerg. Technol. 6 (4), 271–288. http://dx.doi.org/10.1016/S0968-090X(98)00019-9.

Dendorfer, P., Osep, A., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S., Leal-Taixé, L., 2021. Motchallenge: A benchmark for single-camera multiple target tracking. Int. J. Comput. Vis. 129, 845–881. http://dx.doi.org/10.1007/s11263-020-01393-0.

Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L., 2020. MOT20: A benchmark for multi object tracking in crowded scenes. http://dx.doi.org/10.48550/arXiv.2003.09003, arXiv.

DeTone, D., Malisiewicz, T., Rabinovich, A., 2018. SuperPoint: Self-supervised interest point detection and description. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 224–236. http://dx.doi.org/10.1109/CVPRW.2018.00060.

Ding, J., Xue, N., Xia, G.-S., Bai, X., Yang, W., Yang, M.Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., et al., 2021. Object detection in aerial images: A large-scale benchmark and challenges. IEEE Trans. Pattern Anal. Mach. Intell. 44 (11), 7778–7796. http://dx.doi.org/10.1109/TPAMI.2021.3117983.

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H., 2023. Strongsort: Make DeepSORT great again. IEEE Trans. Multimed. http://dx.doi.org/10.1109/TMM.2023.3240881.

Efe, U., Ince, K.G., Alatan, A.A., 2021. Effect of parameter optimization on classical and learning-based image matching methods. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. IEEE Computer Society, Los Alamitos, CA, USA, pp. 2506–2513. http://dx.doi.org/10.1109/ICCVW54120.2021.00283.

Ekaso, D., Nex, F., Kerle, N., 2020. Accuracy assessment of real-time kinematics (RTK) measurements on unmanned aerial vehicles (UAV) for direct geo-referencing. Geo Spat. Inf. Sci. 23 (2), 165–181. http://dx.doi.org/10.1080/10095020.2019.1710437CrossMarkLogoCrossMark.

Espadaler-Clapés, J., Barmpounakis, E., Geroliminis, N., 2023a. Traffic congestion and noise emissions with detailed vehicle trajectories from UAVs. Transp. Res. Part D: Transp. Environ. 121, 103822. http://dx.doi.org/10.1016/j.trd.2023.103822.

Espadaler-Clapés, J., Fonod, R., Barmpounakis, E., Geroliminis, N., 2023b. Continuous monitoring of a signalized intersection using unmanned aerial vehicles. In: 2023 IEEE 26th International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 5338–5343. http://dx.doi.org/10.1109/ITSC57777.2023.10422673.

Fang, J., Meng, H., Zhang, H., Wang, X., 2007. A low-cost vehicle detection and classification system based on unmodulated continuous-wave radar. In: 2007 IEEE Intelligent Transportation Systems Conference. IEEE, pp. 715–720. http://dx.doi.org/10.1109/ITSC.2007.4357739.

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 (6), 381–395. http://dx.doi.org/10.1016/B978-0-08-051581-6.50070-2.

Fonod, R., 2025a. Geo-trax:A Comprehensive Framework for Georeferenced Vehicle Trajectory Extraction from Drone Imagery. Zenodo, http://dx.doi.org/10.5281/zenodo.14589321, URL: https://github.com/rfonod/geo-trax.

Fonod, R., 2025b. Stabilo:A Comprehensive Python Library for Video and Trajectory Stabilization with User-Defined Masks. Zenodo, http://dx.doi.org/10.5281/zenodo.15207016, URL: https://github.com/rfonod/stabilo.

Fonod, R., 2025c. Stabilo Optimize:A Framework for Comprehensive Evaluation and Analysis for the Stabilo Library. http://dx.doi.org/10.5281/zenodo.13828430, URL: https://github.com/rfonod/stabilo-optimize.

Fonod, R., Cho, H., Yeo, H., Geroliminis, N., 2025a. Songdo Traffic: High Accuracy Georeferenced Vehicle Trajectories from a Large-Scale Study in a Smart City. Zenodo, http://dx.doi.org/10.5281/zenodo.13828384, URL: https://doi.org/10.5281/zenodo.13828384.

Fonod, R., Cho, H., Yeo, H., Geroliminis, N., 2025b. Songdo Vision: Vehicle Annotations from High-Altitude BeV Drone Imagery in a Smart City. Zenodo, http://dx.doi.org/10.5281/zenodo.13828408, URL: https://doi.org/10.5281/zenodo.13828408.

Ghahremannezhad, H., Shi, H., Liu, C., 2023. Object detection in traffic videos: A survey. IEEE Trans. Intell. Transp. Syst. http://dx.doi.org/10.1109/TITS.2023.3258683.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587. http://dx.doi.org/10.1109/CVPR.2014.81.

Gloudemans, D., Wang, Y., Ji, J., Zachar, G., Barbour, W., Hall, E., Cebelak, M., Smith, L., Work, D.B., 2023. I-24 motion: An instrument for freeway traffic science. Transp. Res. Part C: Emerg. Technol. 155, 104311. http://dx.doi.org/10.1016/j.trc.2023.104311.

Gohari, A., Ahmad, A., Rahim, R.A., Supa'at, A.S.M., Razak, S.A., Gismalla, M., 2022. Involvement of surveillance drones in smart cities: a systematic review. IEEE Access 10, 56611–56628. http://dx.doi.org/10.1109/access.2022.3177904.

Gokul, L., Adarsh, P., Vinayan, G., Gokuldath, G., Ponmalar, M., Aswini, S., 2023. Lucas kanade based optical flow for vehicle motion tracking and velocity estimation. In: 2023 International Conference on Control, Communication and Computing. ICCC, IEEE, pp. 1–6. http://dx.doi.org/10.1109/ICCC57789.2023.10165227.

Groves, P.D., 2011. Shadow matching: A new GNSS positioning technique for urban canyons. J. Navig. 64 (3), 417–430. http://dx.doi.org/10.1017/S0373463311000087.

Ha, Q., Watanabe, K., Karasawa, T., Ushiku, Y., Harada, T., 2017. MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 5108–5115. http://dx.doi.org/10.1109/IROS.2017.8206396.

Hao, W., Zhili, S., 2020. Improved mosaic: Algorithms for more complex images. J. Phys.: Conf. Ser. 1684 (1), 012094. http://dx.doi.org/10.1088/1742-6596/1684/1/012094.

Hassanalian, M., Abdelkefi, A., 2017. Classifications, applications, and design challenges of drones: A review. Prog. Aerosp. Sci. 91, 99–131. http://dx.doi.org/10.1016/j.paerosci.2017.04.003.

He, J., Chen, H., Liu, B., Luo, S., Liu, J., 2024. Enhancing YOLO for occluded vehicle detection with grouped orthogonal attention and dense object repulsion. Sci. Rep. 14 (1), 19650. http://dx.doi.org/10.1038/s41598-024-70695-x.

Hsieh, M.-R., Lin, Y.-L., Hsu, W.H., 2017. Drone-based object counting by Spatially Regularized Regional proposal network. In: Proceedings of the IEEE International Conference on Computer Vision. ICCV, IEEE, pp. 4145–4153. http://dx.doi.org/10.1109/ICCV.2017.446.

Hsieh, J.-W., Yu, S.-H., Chen, Y.-S., Hu, W.-F., 2006. Automatic traffic surveillance system for vehicle tracking and classification. IEEE Trans. Intell. Transp. Syst. 7 (2), 175–187. http://dx.doi.org/10.1109/TITS.2006.874722.

Hsu, S.-C., Huang, C.-L., Chuang, C.-H., 2018. Vehicle detection using simplified fast R-CNN. In: 2018 International Workshop on Advanced Image Technology. IWAIT, IEEE, pp. 1–3. http://dx.doi.org/10.1109/IWAIT.2018.8369767.

Jian, L., Li, Z., Yang, X., Wu, W., Ahmad, A., Jeon, G., 2019. Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: electronic eyes in the skies to spot clogged roads. IEEE Consum. Electron. Mag. 8 (3), 81–86. http://dx.doi.org/10.1109/MCE.2019.2892286.

Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E., 2021. Image matching across wide baselines: From paper to practice. Int. J. Comput. Vis. 129 (2), 517–547. http://dx.doi.org/10.1007/s11263-020-01385-0.

Jocher, G., 2020. YOLOv5 by Ultralytics. http://dx.doi.org/10.5281/zenodo.7347926, URL: https://github.com/ultralytics/yolov5.
Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLO. http://dx.doi.org/10.5281/zenodo.13942966, URL: https://github.com/ultralytics/ultralytics.
Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. J. Basic Eng. 82 (1), 35–45. http://dx.doi.org/10.1115/1.3662552.
Kaplan, E., Hegarty, C., 2017. Understanding GPS/GNSS: Principles and Applications, third ed. Artech House, USA.
Ke, R., Li, Z., Kim, S., Ash, J., Cui, Z., Wang, Y., 2016. Real-time bidirectional traffic flow parameter estimation from aerial videos. IEEE Trans. Intell. Transp. Syst. 18 (4), 890–901. http://dx.doi.org/10.1109/TITS.2016.2595526.
Khan, M.A., Ectors, W., Bellemans, T., Janssens, D., Wets, G., 2017. Unmanned aerial vehicle–based traffic analysis: Methodological framework for automated multivehicle trajectory extraction. Transp. Res. Rec. 2626 (1), 25–33. http://dx.doi.org/10.3141/2626-04.
Khan, M.A., Ectors, W., Bellemans, T., Ruichek, Y., Janssens, D., Wets, G., et al., 2018. Unmanned aerial vehicle-based traffic analysis: A case study to analyze traffic streams at urban roundabouts. Procedia Comput. Sci. 130, 636–643. http://dx.doi.org/10.1016/j.procs.2018.04.114.
Khan, N.A., Jhanjhi, N., Brohi, S.N., Usmani, R.S.A., Nayyar, A., 2020. Smart traffic monitoring system using unmanned aerial vehicles (UAVs). Comput. Commun. 157, 434–443. http://dx.doi.org/10.1016/j.comcom.2020.04.049.
Kim, D., 2020. Pedestrian and bicycle volume data collection using drone technology. J. Urban Technol. 27, 45–60. http://dx.doi.org/10.1080/10630732.2020.1715158.
Kim, S., Anagnostopoulos, G., Barmpounakis, E., Geroliminis, N., 2023. Visual extensions and anomaly detection in the pNEUMA experiment with a swarm of drones. Transp. Res. Part C: Emerg. Technol. 147, 103966. http://dx.doi.org/10.1016/j.trc.2022.103966.
Kim, Z., Cao, M., 2010. Evaluation of feature-based vehicle trajectory extraction algorithms. In: 13th International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 99–104. http://dx.doi.org/10.1109/ITSC.2010.5625278.
Kim, W., Jung, C., 2017. Illumination-invariant background subtraction: Comparative review, models, and prospects. IEEE Access 5, 8369–8384. http://dx.doi.org/10.1109/ACCESS.2017.2699227.
Kim, S., Tak, Y., Barmpounakis, E., Geroliminis, N., 2024. Monitoring outdoor parking in urban areas with unmanned aerial vehicles. IEEE Trans. Intell. Transp. Syst. http://dx.doi.org/10.1109/TITS.2024.3397588.
Kouris, A., Kyrkou, C., Bouganis, C.-S., 2019. Informed region selection for efficient UAV-based object detectors: Altitude-aware vehicle detection with CyCAR dataset. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 51–58. http://dx.doi.org/10.1109/IROS40897.2019.8967722.
Kouvelas, A., Saeedmanesh, M., Geroliminis, N., 2023. A linear-parameter-varying formulation for model predictive perimeter control in multi-region MFD urban networks. Transp. Sci. 57 (6), 1496–1515. http://dx.doi.org/10.1287/trsc.2022.0103.
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, IEEE, pp. 2118–2125. http://dx.doi.org/10.1109/ITSC.2018.8569552.
Kuhn, H.W., 1955. The hungarian method for the assignment problem. Nav. Res. Logist. Q. 2 (1–2), 83–97. http://dx.doi.org/10.1002/nav.3800020109.
Kumar, R., Sawhney, H.S., Asmuth, J.C., Pope, A., Hsu, S., 1998. Registration of video to geo-referenced imagery. In: Proceedings of the 14th International Conference on Pattern Recognition. 2, IEEE, pp. 1393–1400. http://dx.doi.org/10.1109/ICPR.1998.711963.
Kumar, R., Sawhney, H., Samarasekera, S., Hsu, S., Tao, H., Guo, Y., Hanna, K., Pope, A., Wildes, R., Hirvonen, D., et al., 2001. Aerial video surveillance and exploitation. Proc. IEEE 89 (10), 1518–1539. http://dx.doi.org/10.1109/5.959344.
Leng, J., Ye, Y., Mo, M., Gao, C., Gan, J., Xiao, B., Gao, X., 2024. Recent advances for aerial object detection: A survey. ACM Comput. Surv. 56 (12), 1–36. http://dx.doi.org/10.1145/3664598.
Leutenegger, S., Chli, M., Siegwart, R.Y., 2011. BRISK: Binary robust invariant scalable keypoints. In: 2011 International Conference on Computer Vision. IEEE, pp. 2548–2555. http://dx.doi.org/10.1109/ICCV.2011.6126542.
Li, B., 2017. 3D fully convolutional network for vehicle detection in point cloud. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 1513–1518. http://dx.doi.org/10.1109/IROS.2017.8205955.
Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., Wei, X., 2022. YOLOv6: A single-stage object detection framework for industrial applications. arXiv arXiv:2209.02976v1.
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, pp. 740–755. http://dx.doi.org/10.1007/978-3-319-10602-1_48.
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: Computer Vision – ECCV 2016. Springer, pp. 21–37. http://dx.doi.org/10.1007/978-3-319-46448-0_2.
Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision. ICCV, IEEE Computer Society, Los Alamitos, CA, USA, pp. 9992–10002. http://dx.doi.org/10.1109/ICCV48922.2021.00986.
Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60, 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.
Mahajan, V., Barmpounakis, E., Alam, M.R., Geroliminis, N., Antoniou, C., 2023. Treating noise and anomalies in vehicle trajectories from an experiment with a swarm of drones. IEEE Trans. Intell. Transp. Syst. 24 (9), 9055–9067. http://dx.doi.org/10.1109/TITS.2023.3268712.
Makrigiorgis, R., Kolios, P., Kyrkou, C., 2022. Aerial Multi-Vehicle Detection Dataset. Zenodo, http://dx.doi.org/10.5281/zenodo.7053442.
Makrigiorgis, R., Kolios, P., Timotheou, S., Theocharides, T., Panayiotou, C.G., 2020. Extracting the fundamental diagram from aerial footage. In: 2020 IEEE 91st Vehicular Technology Conference. IEEE, pp. 1–5. http://dx.doi.org/10.1109/VTC2020-Spring48590.2020.9128534.
Mandellos, N.A., Keramitsoglou, I., Kiranoudis, C.T., 2011. A background subtraction algorithm for detecting and tracking vehicles. Expert Syst. Appl. 38 (3), 1619–1631. http://dx.doi.org/10.1016/j.eswa.2010.07.083.
Mao, Q.-C., Sun, H.-M., Zuo, L.-Q., Jia, R.-S., 2020. Finding every car: a traffic surveillance multi-scale vehicle object detection method. Appl. Intell. 50, 3125–3136. http://dx.doi.org/10.1007/s10489-020-01704-5.
Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K., 2016. MOT16: A benchmark for multi-object tracking. http://dx.doi.org/10.48550/arXiv.1603.00831, arXiv.
Mingkhwan, E., Khawsuk, W., 2017. Digital image stabilization technique for fixed camera on small size drone. In: 2017 Third Asian Conference on Defence Technology. ACDT, IEEE, pp. 12–19. http://dx.doi.org/10.1109/ACDT.2017.7886149.
Muja, M., Lowe, D.G., 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Applications. pp. 331–340. http://dx.doi.org/10.5220/0001787803310340.
Novikov, D., Sotirelis, P., Yilmaz, A., 2024. Vehicle geolocalization from drone imagery. ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. X-2, 171–181. http://dx.doi.org/10.5194/isprs-annals-X-2-2024-171-2024.
Outay, F., Mengash, H.A., Adnan, M., 2020. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. Transp. Res. Part A: Policy Pr. 141, 116–129. http://dx.doi.org/10.1016/j.tra.2020.09.018.
Paipuri, M., Barmpounakis, E., Geroliminis, N., Leclercq, L., 2021. Empirical observations of multi-modal network-level models: Insights from the pNEUMA experiment. Transp. Res. Part C: Emerg. Technol. 131, 103300. http://dx.doi.org/10.1016/j.trc.2021.103300.

Park, S.J., Kim, T.Y., Kang, S.M., Koo, K.H., 2003. A Novel Signal Processing Technique for Vehicle Detection Radar. In: IEEE MTT-S International Microwave Symposium Digest, 2003, vol. 1, IEEE, pp. 607–610. http://dx.doi.org/10.1109/MWSYM.2003.1211012.
Petersson, M., Kifle Solomon, N., 2024. Object Tracking Evaluation:BoT-SORT &ByteTrack with YOLOv8:A Comparison of Accuracy and Computational Efficiency. (Dissertation. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-351280.
Puertas, E., De-Las-Heras, G., Fernández-Andrés, J., Sánchez-Soriano, J., 2022. Dataset: Roundabout aerial images for vehicle detection. Data 7 (4), 47. http://dx.doi.org/10.3390/data7040047.
Rauch, H.E., Tung, F., Striebel, C.T., 1965. Maximum likelihood estimates of linear dynamic systems. AIAA J. 3 (8), 1445–1450. http://dx.doi.org/10.2514/3.3166.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788. http://dx.doi.org/10.1109/CVPR.2016.91.
Redmon, J., Farhadi, A., 2018. YOLOv3: An incremental improvement. http://dx.doi.org/10.48550/arXiv.1804.02767, arXiv preprint arXiv:1804.02767.
Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (06), 1137–1149. http://dx.doi.org/10.1109/TPAMI.2016.2577031.
Ren, Y., Zhu, C., Xiao, S., 2018. Small object detection in optical remote sensing images via modified faster R-CNN. Appl. Sci. 8 (5), 813. http://dx.doi.org/10.3390/app8050813.
Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision. IEEE, pp. 2564–2571. http://dx.doi.org/10.1109/ICCV.2011.6126544.
Sarlin, P.-E., DeTone, D., Malisiewicz, T., Rabinovich, A., 2020. SuperGlue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4938–4947. http://dx.doi.org/10.1109/CVPR42600.2020.00499.
Savitzky, A., Golay, M.J., 1964b. Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. 36 (8), 1627–1639. http://dx.doi.org/10.1021/ac60214a047.
Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. Annu. Rev. Control. 43, 128–151. http://dx.doi.org/10.1016/j.arcontrol.2017.03.005.
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. http://dx.doi.org/10.1016/j.commtr.2021.100014.
Shi, J., et al., 1994. Good features to track. In: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 593–600. http://dx.doi.org/10.1109/CVPR.1994.323794.
Sifuentes, E., Casas, O., Pallas-Areny, R., 2011. Wireless magnetic sensor node for vehicle detection with optical wake-up. IEEE Sensors J. 11 (8), 1669–1676. http://dx.doi.org/10.1109/JSEN.2010.2103937.
Sivaraman, S., Trivedi, M.M., 2013. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Trans. Intell. Transp. Syst. 14 (4), 1773–1795. http://dx.doi.org/10.1109/TITS.2013.2266661.
Su, H., Liu, S., Zheng, B., Zhou, X., Zheng, K., 2020. A survey of trajectory distance measures and performance evaluation. VLDB J. 29, 3–32. http://dx.doi.org/10.1007/s00778-019-00574-9.
Toutin, T., 2004. Review article: Geometric processing of remote sensing images: models, algorithms and methods. Int. J. Remote Sens. 25 (10), 1893–1924. http://dx.doi.org/10.1080/0143116031000101611.
Tran, T.M., Vu, T.N., Nguyen, T.V., Nguyen, K., 2023. UIT-ADrone: A novel drone dataset for traffic anomaly detection. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 16, 5590–5601. http://dx.doi.org/10.1109/JSTARS.2023.3285905.
Tsitsokas, D., Kouvelas, A., Geroliminis, N., 2023. Two-layer adaptive signal control framework for large-scale dynamically-congested networks: Combining efficient max pressure with perimeter control. Transp. Res. Part C: Emerg. Technol. 152, 104128. http://dx.doi.org/10.1016/j.trc.2023.104128.
Venthuruthiyil, S.P., Chunchu, M., 2020. Vehicle path reconstruction using recursively ensembled low-pass filter (RELP) and adaptive tri-cubic kernel smoother. Transp. Res. Part C: Emerg. Technol. 120, 102847. http://dx.doi.org/10.1016/j.trc.2020.102847.
Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M., 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7464–7475. http://dx.doi.org/10.1109/CVPR52729.2023.00721.
Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G., 2024a. YOLOv10: Real-time end-to-end object detection. arXiv arXiv:2405.14458.
Wang, Y., Gloudemans, D., Ji, J., Teoh, Z.N., Liu, L., Zachár, G., Barbour, W., Work, D., 2024b. Automatic vehicle trajectory data reconstruction at scale. Transp. Res. Part C: Emerg. Technol. 160, 104520. http://dx.doi.org/10.1016/j.trc.2024.104520.
Wang, C.-Y., Yeh, I.-H., Liao, H.-Y.M., 2024. YOLOv9: Learning what you want to learn using programmable gradient information. arXiv arXiv:2402.13616.
Wang, Q., Zheng, J., Xu, H., Xu, B., Chen, R., 2017. Roadside magnetic sensor system for vehicle detection in urban environments. IEEE Trans. Intell. Transp. Syst. 19 (5), 1365–1374. http://dx.doi.org/10.1109/TITS.2017.2723908.
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, IEEE, pp. 3645–3649. http://dx.doi.org/10.1109/ICIP.2017.8296962.
Wu, B.-F., Kao, C.-C., Jen, C.-L., Li, Y.-F., Chen, Y.-H., Juang, J.-H., 2013. A relative-discriminative-histogram-of-oriented-gradients-based particle filter approach to vehicle occlusion handling and tracking. IEEE Trans. Ind. Electron. 61 (8), 4228–4237. http://dx.doi.org/10.1109/TIE.2013.2284131.
Wu, J., Man, Q., Yang, X., Dong, P., Ma, X., Liu, C., Han, C., 2024. Fine classification of urban tree species based on UAV-based RGB imagery and LiDAR data. Forests 15 (2), 390. http://dx.doi.org/10.3390/f15020390.
Yang, Z., Pun-Cheng, L.S., 2018. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. Image Vis. Comput. 69, 143–154. http://dx.doi.org/10.1016/j.imavis.2017.09.008.
Yilmaz, A., Javed, O., Shah, M., 2006. Object tracking: A survey. ACM Comput. Surv. 38 (4), 13–es. http://dx.doi.org/10.1145/1177352.117735.
Yu, H., Li, G., Zhang, W., Huang, Q., Du, D., Tian, Q., Sebe, N., 2020. The unmanned aerial vehicle benchmark: Object detection, tracking and baseline. Int. J. Comput. Vis. 128, 1141–1159. http://dx.doi.org/10.1007/s11263-019-01266-1.
Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., de La Fortelle, A., et al., 2019. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. http://dx.doi.org/10.48550/arXiv.1910.03088, arXiv:1910.03088 [Cs, Eess].
Zhang, Y., Guo, Z., Wu, J., Tian, Y., Tang, H., Guo, X., 2022a. Real-time vehicle detection based on improved YOLO v5. Sustainability 14 (19), 12274. http://dx.doi.org/10.3390/su141912274.
Zhang, T., Jin, P.J., 2019. A longitudinal scanline based vehicle trajectory reconstruction method for high-angle traffic video. Transp. Res. Part C: Emerg. Technol. 103, 104–128. http://dx.doi.org/10.1016/j.trc.2019.03.015.
Zhang, R., Newman, S., Ortolani, M., Silvestri, S., 2018. A network tomography approach for traffic monitoring in smart cities. IEEE Trans. Intell. Transp. Syst. 19 (7), 2268–2278. http://dx.doi.org/10.1109/TITS.2018.2829086.
Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X., 2022b. Bytetrack: Multi-object tracking by associating every detection box. In: European Conference on Computer Vision. Springer, pp. 1–21. http://dx.doi.org/10.1007/978-3-031-20047-2_1.
Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W., 2021. Fairmot: On the fairness of detection and re-identification in multiple object tracking. Int. J. Comput. Vis. 129, 3069–3087. http://dx.doi.org/10.1007/s11263-021-01513-4.
Zhang, J., Yang, H., Chen, Z., Yu, T., Liu, H., 2022c. Probability distribution and recommended values of passenger car external dimensions in mechanical parking garage design. J. Asian Archit. Build. Eng. 21 (5), 1942–1958. http://dx.doi.org/10.1080/13467581.2021.1971995.

Zhang, Z., Zheng, J., Xu, H., Wang, X., 2019. Vehicle detection and tracking in complex traffic circumstances with roadside LiDAR. Transp. Res. Rec. 2673 (9), 62–71. http://dx.doi.org/10.1177/03611981198444.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J., 2024. DETRs beat YOLOs on real-time object detection. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE Computer Society, Los Alamitos, CA, USA, pp. 16965–16974. http://dx.doi.org/10.1109/CVPR52733.2024.01605.

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. 2678 (4), 606–621. http://dx.doi.org/10.1177/03611981231185768.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., 2021a. Deformable DETR: Deformable transformers for end-to-end object detection. arXiv arXiv:2010.04159v4.

Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H., 2021. Detection and tracking meet drones challenge. IEEE Trans. Pattern Anal. Mach. Intell. 44 (11), 7380–7399. http://dx.doi.org/10.1109/TPAMI.2021.3119563.