A Technical Appendices and Supplementary Material

A.1 Coordinate Systems and Transformation

To achieve spatial synchronization between different sensors, vehicle-vehicle-UAV collaboration requires using sensor parameter information to perform coordinate system transformations. The relationships between the coordinate systems are illustrated in Fig. S 1.

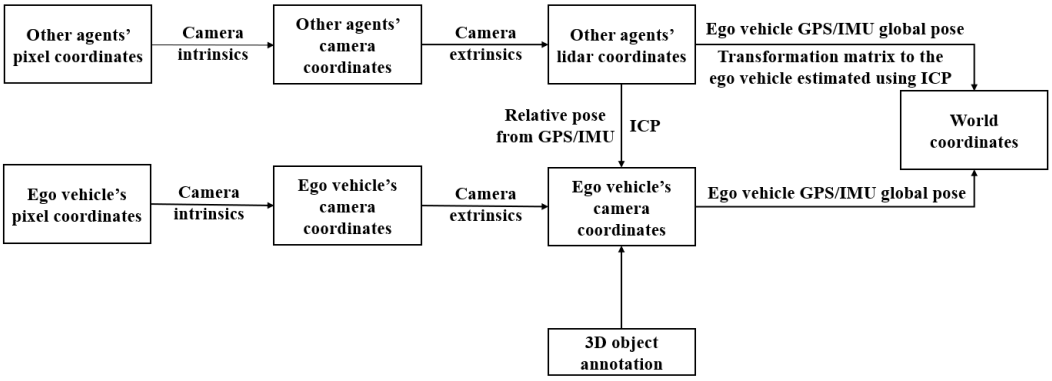


Figure 1: Relationship between coordinate systems.

Pixel Coordinates. The pixel coordinate system refers to a two-dimensional coordinate system defined on the image plane, typically represented as (u, v), with units in pixels. In this system, the origin is located at the top-left corner of the image, the u-axis points to the right along the horizontal direction, and the v-axis points downward along the vertical direction. This coordinate system is used to describe the position of points on the two-dimensional image captured by the camera.

A 3D point in the camera coordinate system, denoted as (x_c, y_c, z_c), can be projected onto the pixel coordinate system through the camera's intrinsic matrix. The transformation process can be expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ 1 \end{bmatrix} \tag{1}$$

where f_x and f_y represent the focal lengths along the image's x and y axes (in pixel units), and (c_x, c_y) denote the principal point (the intersection of the optical axis with the image plane, in pixel coordinates).

Camera Coordinate System and LiDAR-to-Camera Calibration. The camera coordinate system is defined as a three-dimensional right-handed Cartesian coordinate system, with its origin located at the optical center of the camera. In this system, the x-axis points to the right along the image plane, the y-axis points downward along the image plane, and the z-axis extends forward along the optical axis of the camera.

To determine the spatial relationship between the LiDAR and each camera, we employed a point correspondence-based calibration procedure [1, 2]. Specifically, for each individual camera view, several corresponding feature points were manually selected in both the image and the LiDAR point cloud. Based on these correspondences, an initial extrinsic transformation matrix from the camera to the LiDAR was estimated using a least-squares fitting approach.

To improve calibration accuracy, the initial matrix was further refined through iterative manual adjustment and validation by visually checking the alignment of projected LiDAR points on the image plane. In order to ensure long-term calibration reliability, considering possible sensor shifts and mechanical vibrations, this calibration procedure was performed once every four hours during continuous data collection.

The final extrinsic parameters for each camera were stored as a 4×4 homogeneous transformation matrix, representing the coordinate transformation from the LiDAR coordinate system to the corresponding camera coordinate system, as expressed by:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \mathrm{T}_{\text{LiDAR2Cam}} \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} \tag{2}$$

where $\mathbf{T}_{\text{LiDAR2Cam}}$ denotes the extrinsic matrix obtained from the calibration process, and (x_l, y_l, z_l) and (x_c, y_c, z_c) are the point coordinates in the LiDAR and camera coordinate systems, respectively.

A visualization of the LiDAR-to-camera calibration results for all recording platforms is provided in Fig. S2 S3 S4. The visualizations show the LiDAR point clouds projected onto the corresponding camera images using the estimated extrinsic parameters. Our dataset includes two ground vehicles, each equipped with five cameras providing full $360°$ coverage, and a UAV equipped with a single front-facing camera. The calibration results for each vehicle and the UAV are displayed separately, demonstrating the alignment quality across all viewpoints. The consistency between the projected LiDAR points and the visible object boundaries in the images effectively verifies the accuracy and robustness of our calibration process.

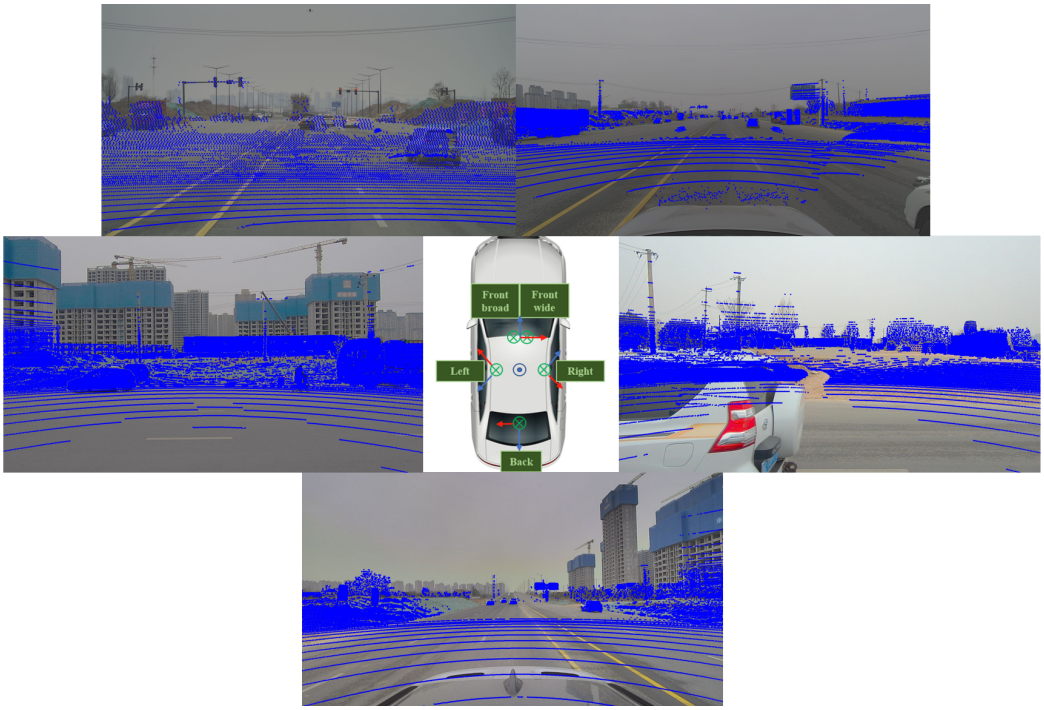


Figure 2: Visualization of the LiDAR-to-camera calibration for Ground Vehicle A equipped with five cameras covering $360°$. Projected LiDAR points align well with image features across all camera views.

LiDAR Coordinate System and World Coordinate System. The LiDAR coordinate system for each platform is defined relative to the sensor's installation frame on that platform. We adopt a right-handed coordinate system, where the geometric center of the LiDAR sensor is set as the origin. The x-axis points forward, the y-axis points to the left, and the z-axis points upward. The world coordinate system is established as a global East-North-Up (ENU) frame derived from GPS measurements, which provides a consistent geodetic reference for all platforms.

Point clouds collected from each platform are initially represented in their respective LiDAR coordinate systems. Using GPS and IMU data, the pose of each platform is obtained relative to the global ENU world coordinate system. In our implementation, we approximate the LiDAR-to-world

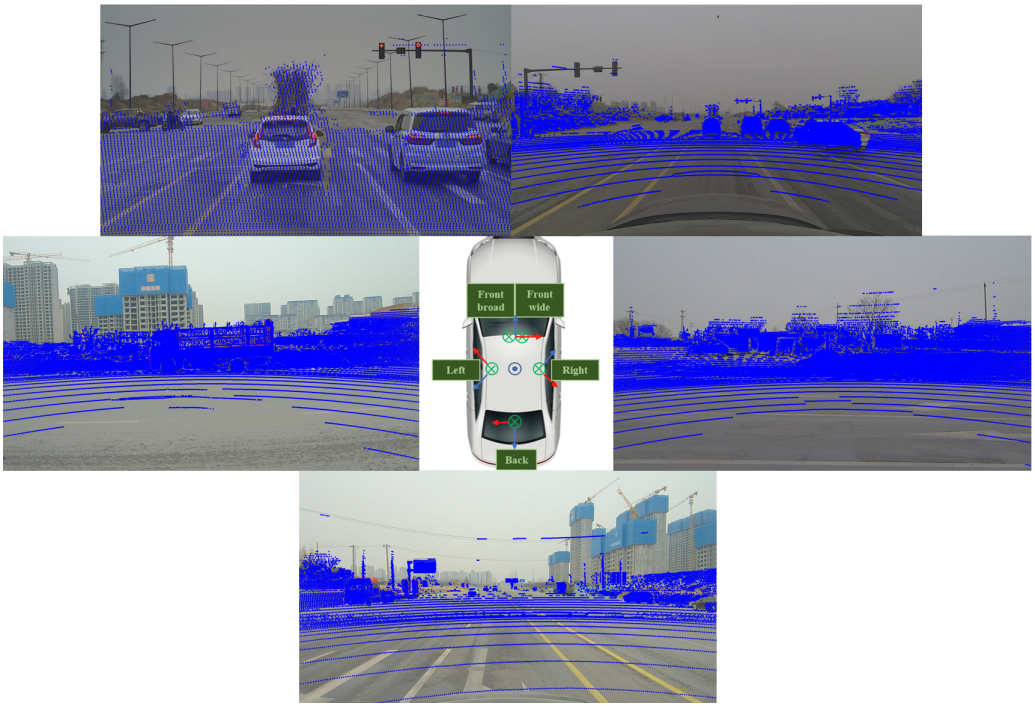
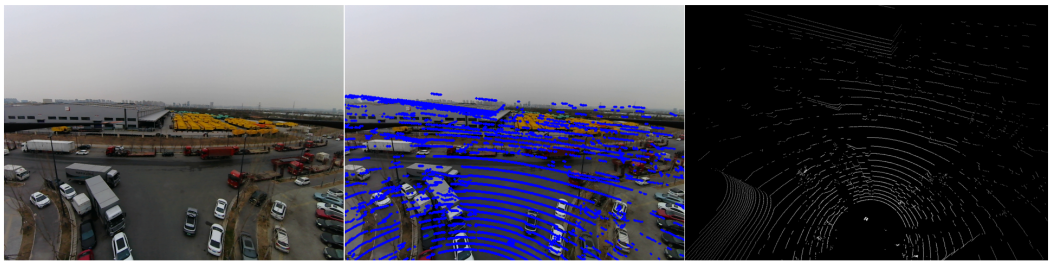


Figure 3: Visualization of the LiDAR-to-camera calibration for Ground Vehicle B equipped with five cameras covering 360°. Projected LiDAR points align well with image features across all camera views.



(a) Image (b) Projected points (c) Point cloud

Figure 4: Multi-modal data alignment from a UAV perspective. (a) Aerial image captured by the UAV-mounted camera. (b) LiDAR point cloud projected onto the image plane for visualizing alignment accuracy. (c) Top-down view of the LiDAR point cloud acquired from the UAV.

transformation using the GPS/IMU-derived vehicle pose, assuming negligible displacement between the LiDAR sensor and the localization reference point.

The transformation of a point $\mathbf{p}_{\text{lidar}}$ in the LiDAR coordinate system to the world coordinate system is performed as:

$$P_{\text{w}} \approx \mathbf{T}_{\text{w}}^{\text{vehicle}} P_{\text{l}} \tag{3}$$

where $\mathbf{T}_{\text{w}}^{\text{vehicle}} \in SE(3)$ is the vehicle pose in the world coordinate frame obtained from GPS/IMU localization.

To compensate for residual misalignments caused by the approximation, an Iterative Closest Point (ICP) [3] algorithm is applied to refine the registration of point clouds from different platforms relative to the ego vehicle's LiDAR frame before transforming them to the world coordinate system.

The final transformation for a point cloud from another platform is given by:

$$P_{\text{w}}^{i} = \mathbf{T}_{\text{w}}^{\text{ego}} \mathbf{T}_{\text{ego}}^{i} P_{i} \tag{4}$$

where $\mathbf{T}_{\text{ego}}^{i}$ is the ICP-refined relative pose between platform i and the ego vehicle.

A.2 Multi-agent Time Synchronization

Time Source Synchronization. In our multi-agent system, all platforms achieve unified time source synchronization through GPS-based timing signals. Each platform's onboard clock is disciplined by the GPS receiver, providing a highly accurate and stable global time reference. This approach effectively eliminates clock drift and offset among different agents, ensuring that all sensors across vehicles and the UAV are synchronized to the same absolute time base. As a result, temporal consistency is maintained across heterogeneous sensors and platforms, which is critical for tasks such as sensor fusion, data alignment, and multi-agent cooperative perception.

Timestamp Synchronization. Although all platforms in our system share a common GPS-based time source, the sensors operate at different sampling frequencies, and their measurements are not necessarily captured at exactly the same timestamps. To address this, we employ the `message_filters` package in ROS to perform precise timestamp synchronization. This framework matches sensor messages based on their timestamps by finding the temporally nearest frames across heterogeneous data streams. In doing so, it compensates for both acquisition frequency differences and minor delays, ensuring accurate temporal alignment for multi-sensor fusion and multi-agent cooperative perception.

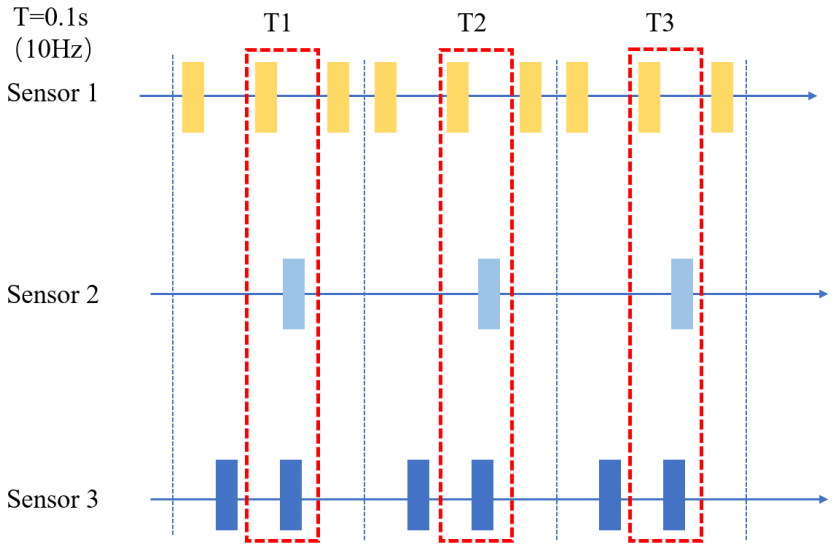


Figure 5: At 10 Hz, timestamp synchronization is performed for sensor data with different frequencies. The nearest frame within the red dashed box is regarded as the data corresponding to the same timestamp within this sampling period.

The combination of GPS-based time source synchronization and message-level timestamp synchronization enables reliable multi-sensor fusion and cooperative perception across heterogeneous platforms.

A.3 AGC-Drive Dataset Statistics

3D Bounding Box Category Distribution. To provide a comprehensive overview of the dataset, we present the number of annotated 3D bounding boxes for each object category. The dataset defines a total of 13 categories, which we group into two main groups: Vehicle and Other. The Vehicle group includes four subcategories: *Car*, *Bus*, *Truck*, and *Van*, while the Other group covers nine subcategories: *Person*, *Bicycle*, *Tricycle*, *Motorcycle*, *Rider*, *Traffic Sign*, *Barrier*, *Cone*, and *Others*.

The detailed number of 3D bounding boxes for each subcategory is illustrated in Fig. S6. As shown in the figure, *Car* is the most frequently annotated category with over 650K instances, followed by *Sign*, *Truck*, and *Rider*. This distribution reflects the typical composition of cooperative driving environments, which feature a high density of vehicles and static traffic infrastructures like traffic

signs and barriers. In comparison, dynamic vulnerable road users such as *Bicycles*, *Motorcycles*, and *Persons* are less commonly observed.

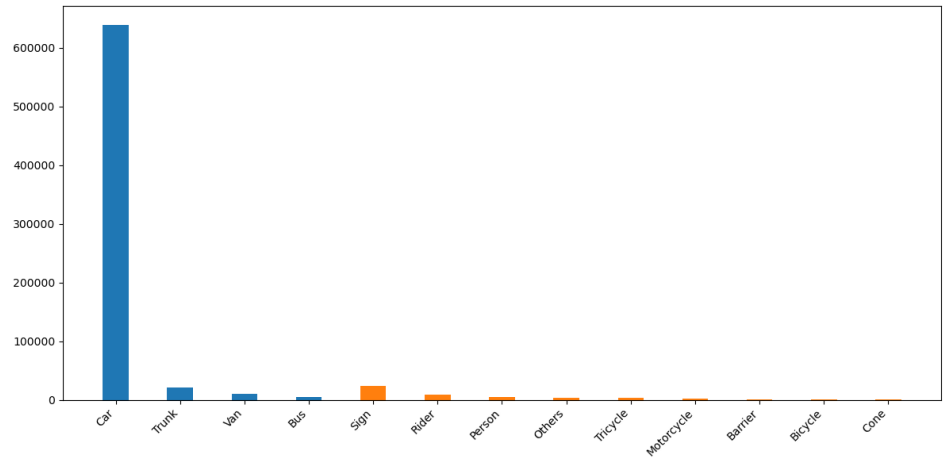


Figure 6: The number of annotated 3D bounding boxes for each object subcategory in our dataset.

A.4 AGC-Drive vs. CoPeD

Table S1 summarizes the comparison between AGC-Drive and CoPeD [4] datasets. Both datasets provide real-world, multi-agent cooperative perception data, integrating LiDAR, camera, and GNSS/IMU sensors to support collaborative tasks in diverse environments. Additionally, both support ground and aerial agents, enabling cross-platform multi-robot cooperation.

However, significant differences exist. AGC-Drive focuses on real driving environments (rural, urban, highway) with higher vehicle speeds, while CoPeD covers mixed indoor and outdoor robot scenarios at lower speeds. AGC-Drive uniquely offers aerial LiDAR data from UAVs, in-cabin cameras, and 3D bounding box annotations with occlusion labels, providing richer multi-view and multi-modal data. In contrast, CoPeD provides 2D bounding boxes only and relies on automatic annotation methods. Furthermore, AGC-Drive contributes a larger scale of point clouds and images, with available source code, enhancing its value as an open benchmark for autonomous driving research.

Table 1: Comparison between CoPeD and AGC-Drive.

	AGC-Drive	CoPeD
Source	Real	Real
scenario types	14 Diverse driving scenarios	Mixed indoor and outdoor environments
Agents	2*Veh & 1*UAV	3*Ground robots & 2*Aerial robots
Sensors	Camera, Lidar, IMU/GPS, Radar, In-cabin camera	Camera, Lidar, IMU/GPS
Aerial LiDAR Support	✓	×
Cams (/Agent)	Multiple	Single
Height	15 to 20m	2m, 2 to 10m
Vehicle speed	30(Rural), 30 to 50(Urban), 80(highway) km/h	1.8(Indoor), 5.4(Outdoor) km/h
Categories	13	-
Labels	3D Boxes & Occlusion	2D Boxes
Images	360,000	203,400
Pointclouds	80,000	-
Source code	✓	only calibration

References

[1] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE, 2018.

[2] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna. LiDAR-Camera Calibration using 3D-3D Point correspondences. *ArXiv e-prints*, May 2017.

[3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[4] Yang Zhou, Long Quang, Carlos Nieto-Granda, and Giuseppe Loianno. Coped-advancing multi-robot collaborative perception: A comprehensive dataset in real-world environments. *IEEE Robotics and Automation Letters*, 9(7):6416–6423, 2024.