AirSim360: A Panoramic Simulation Platform within Drone View

Xian Ge[1*]; Yuling Pan[1,6*], Yuhang Zhang[1*], Xiang Li[1], Weijun Zhang[1‡], Dizhe Zhang[1],
Zhaoliang Wan[1], Xin Lin[1,3], Xiangkai Zhang[1], Juntao Liang[1], Jason Li[4],
Wenjie Jiang[1], Bo Du[2], Ming-Hsuan Yang[5], Lu Qi[1,2†]

[1] Insta360 Research [2] Wuhan University [3] University of California, San Diego
[4] Nanyang Technological University [5] University of California, Merced [6] Shenzhen University

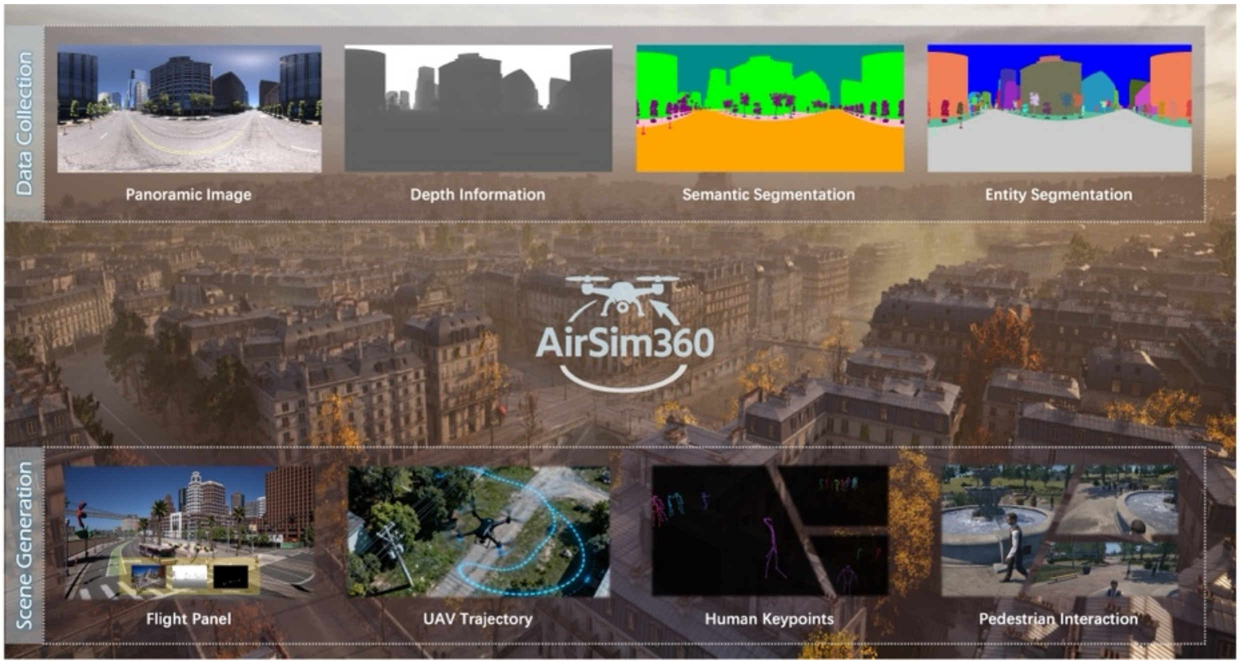


Figure 1. Overview of Airsim360. This work introduces a panoramic UAV simulation platform based on a cutting-edge rendering engine, enabling closed-loop simulation for omnidirectional aerial systems and offering an integrated toolchain for intelligent data acquisition across diverse flight scenarios.

Abstract

The field of 360-degree omnidirectional understanding has been receiving increasing attention for advancing spatial intelligence. However, the lack of large-scale and diverse data remains a major limitation. In this work, we propose AirSim360, a simulation platform for omnidirectional data from aerial viewpoints, enabling wide-ranging scene sampling with drones. Specifically, AirSim360 focuses on three key aspects: a render-aligned data and labeling paradigm for pixel-level geometric, semantic, and instance-level understanding; an interactive pedestrian-aware system for modeling human behavior; and an automated trajectory generation paradigm to support navigation tasks. Furthermore, we collect more than 60K panoramic samples and conduct extensive experiments across various tasks to demonstrate the effectiveness of our simulator. Unlike existing simulators, our work is the first to systematically model the 4D real world under an

*, ‡, † indicate equal contribution, project leader and corresponding author.

omnidirectional setting. The entire platform, including the toolkit, plugins, and collected datasets, will be made publicly available at https://insta360-research-team.github.io/AirSim360-website/.

1. Introduction

Embodied intelligence [7] with omnidirectional perception [32, 47] has gained increasing attention due to the 360° full view in spatial intelligence. It can benefit various robotic applications [13], such as omnidirectional obstacle avoidance [56] during navigation tasks [40].

Different from large-scale perspective image datasets [15, 30], omnidirectional data [14, 44] remain scarce [19] due to the limited usage of 360° cameras in daily life, not to mention the exhaustive human labeling required for many tasks. As a result, most panoramic methods are restricted by small datasets and have scarcely explored data scaling [1].

Inspired by recent advances in simulation platforms [17, 20, 24, 27, 46, 48, 58], a straightforward solution is to rotate agent across multiple angles in simulator to capture an omnidirectional view, including both images and corresponding ground truth. However, this approach introduces two major issues. First, it is computationally inefficient, requiring repeated rendering and significantly increasing data collection time. Second, the definitions of ground-truth signals are not aligned with those in the perspective domain. For example, omnidirectional depth [28] represents slant range along the viewing ray rather than the orthogonal z-axis distance used in perspective projection.

In this work, we focus on building AirSim360, an omnidirectional simulation platform in the drone view to model the 4D real world that consists of high-quality static environments and movable pedestrians. The reason we choose UAV as our agent because it can sample much more data by exploring a wider range of spaces than a ground-based one. Based on the Unreal Engine (UE) 5 series for scene rendering, AirSim360 integrates custom dynamics and communication modules into UE, enabling UAVs to execute actions driven by an external physical modeling engine and serving as the core engine for our entire data-collection toolkit. Table 1 summarizes the core capabilities of Airsim360, which offers a comprehensive API suite ranging from data acquisition (top) to flight control interfaces (bottom). Unlike existing platforms, our simulator supports full runtime interaction, enabling the generation of video-level panoptic segmentation annotations.

Specifically, our AirSim360 has three main characters including render-aligned data and label generation, interactive pedestrian-aware system, and automated trajectory generation paradigm. For data collection, we adopt the equirectangular projection (ERP) as an omnidirectional representation, which can compress multiple perspective views in a single continuous image. However, this representation introduces several challenges for both image content and human- or pixel-level annotations that should be carefully addressed. On the one hand, we propose a GPU-side texture copying mechanism based on Render Hardware Interface (RHI) to enable fast and seamless stitching of six cube-face views captured simultaneously, while maintaining lighting consistency across all faces. On the other hand, the depth information is re-calculated to the slant range along the viewing ray and segmentation annotations have semantic and entity-level labels across frames, ensuring complete panoramic coverage and consistent entity identities over time. Furthermore, we develop point-to-point path planning schemes for automatic data collection and introduce an interactive pedestrian-aware system that simulates movable pedestrians with various actions and annotated keypoints, thus enhancing human-centric perception and analysis.

Finally, extensive experiments across five panoramic tasks, including depth estimation, semantic/entity segmentation, human keypoint detection, and vision-language navigation, demonstrate that our simulated data transfers effectively to real-world scenarios.

Our contributions are fivefold.

- We propose a simulation platform, Airsim360, with native support for 360-degree aerial scenarios. Built on the UE 5 Series with an aerial dynamics module for high-quality scene rendering, the platform provides offline data generation tools capable of capturing panoramic images and corresponding ground truth, including semantic segmentation, entity segmentation, depth estimation, and 3D human keypoints.
- For efficient data collection, we develop point-to-point path planning schemes that enable both automatic sampling and user-designed flight paths. Moreover, we introduce an interactive pedestrian-aware system that synthesizes pedestrian behaviors with detailed annotations and models realistic interactions.
- We test our simulated data for various perception and navigation tasks. Extensive experiments

Table 1. Comparison of Simulators. This table compares recent simulation platforms across nine key dimensions, including rendering quality, interface support, and data generation flexibility, highlighting the advantages of our generative-AI-oriented simulator. Our **specific characters** are highlighted. "Ent" indicate the entity segmentation that cover the whole image.

Platform	AirSim [46]	CARLA [17]	Cosys-AirSim [24]	OmniGibson [27]	UnrealZoo [58]	OpenFly [20]	AirSim360 (ours)
Realism level	Medium	High	High	Medium	High	High	High
Configurable Dynamics	No	No	No	No	No	No	Yes
API type	Python	Python	Python	Python	Python	Python	Python / Blueprint
Scenario type	Outdoor/Indoor	Outdoor	Outdoor/Indoor	Outdoor/Indoor	Outdoor/Indoor	Outdoor	Outdoor/Indoor
UAV simulator	Yes	No	Yes	No	Yes	Yes	Yes
Annotation capability	Dep/Seg	Dep/Seg/Ins	Dep/Seg/Ins	Dep/Seg/Ins	Dep/Seg/Ins	Dep/Seg/Ins	Dep/Seg/Ent
Custom label support	No	No	Yes	Yes	No	No	Yes
Panoramic image	Yes	No	No	No	No	No	Yes
Rendering engine	UE 4.27	UE 5	UE 5.2	IsaacSim	UE 5	UE 5	UE 4.27 - UE 5.6

demonstrate our data significantly improve performance and robustness when evaluated on real-world validation sets.

- To our knowledge, AirSim360 is the first platform to support omnidirectional navigation for UAVs. Compared with conventional monocular systems, panoramic UAVs provide superior perceptual coverage, enabling more efficient target search with minimal additional motion cost.
- AirSim360 has excellent backward compatibility with ranging from latest UE 5.6 down to 4.27, our toolkit and benchmark tasks can run across these versions.

2. Related Work

UAV Dataset UAV datasets typically contain raw images and task-related ground truth, such as trajectory waypoints, depth maps, semantic labels, and point clouds. In general, such data are obtained through acquisition of real-world flights with manual annotation [18, 26, 51] or all-in-one simulation platforms [33, 35, 45]. However, existing datasets are designed primarily for perspective views with limited fields of view. In contrast, we focus on an omnidirectional setting with 360° coverage in aerial scenarios, which enables panoramic UAV tasks such as omnidirectional obstacle avoidance.

Embodied Simulator Embodied simulators [3, 8, 39] are crucial in robotics, such as autonomous driving and robotic grasping. It is because real-world data collection is often challenging, especially for rare long-tail scenarios. As UAVs play an increasingly important role, a range of UAV-oriented simulation platforms have emerged, including AirSim, UnrealCV [43], OpenFly, UAVScenes [50] and UnrealZoo. Among them, AirSim is an earlier simulator that utilizes UE for aerial scene rendering, but its latest officially supported version remains UE 4.27. UnrealCV and UnrealZoo are plugins built on top of UE that provide socket-based interfaces to capture RGB images, depth estimation, and semantic segmentation. However, they only support conventional perspective views and provide limited geometric and semantic information at relatively low frame rates, while lacking entity-level discrimination and making them insufficient for complex panoramic UAV tasks. In Table 1, our UAV360 platform additionally enables omnidirectional perception with full 360-degree coverage, and offers entity-level segmentation and 3D keypoint annotations to support advanced panoramic UAV applications at high frame rate.

Panoramic Task Panoramic visual tasks in understanding [52, 54] and generation [9, 10, 23] have rapidly advanced in support of spatial intelligence [31], where equirectangular projection (ERP) remains the most prevalent representation due to its straightforward mapping from spherical coordinates to a rectangular image. However, existing methods typically rely on real-world datasets with limited scale and diversity [4, 12, 16, 37, 53, 55], leaving their generalization ability across domains underexplored. In this work, we systematically benchmark depth estimation, pixel-level scene parsing, human keypoint detection, and UAV navigation under both in-domain and out-of-domain settings, demonstrating the merit of the proposed UAV360 platform.

3. AirSim360 Platform

As shown in Figure 2, AirSim360 is a large-scale omnidirectional simulation platform for drone scenarios. Unlike other platforms such as AirSim and UnrealZoo, we emphasize consistency and compatibility among three key components in low-altitude ground environments, including static surroundings, the UAV, and human actors, which aligns

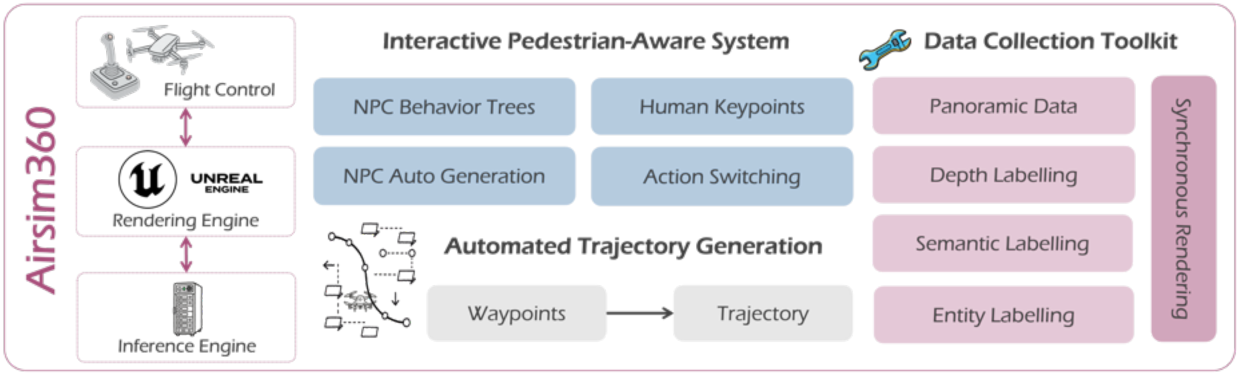


Figure 2. The left panel depicts the core interaction architecture of Airsim360, comprising the flight control module, rendering engine, and inference engine. The middle and right panels showcase three purpose-built data generation modules: the Interactive Pedestrian-Aware System, Automated Trajectory Generator, and Data Collection Toolkit.

closely with the main characteristics of the human-centric real world.

In the following subsections, we first introduce our simulation architecture and its core design, including the simulation environment, flight control, and communication system for efficient panoramic data sampling. Next, we describe the off-line data collection system, which generates panoramic images and pixel-level ground truth such as semantic segmentation, entity segmentation, and human keypoints using an automated trajectory generation tool. Finally, we present the interactive pedestrian-aware system, which enables realistic human behaviors and interactions within the simulation environment.

3.1. Overview

Built upon high-quality rendering capabilities of Unreal Engine, AirSim360 is an online closed-loop simulator compatible with multiple versions ranging from UE 4.27 to UE 5.6. By default, we adopt UE5 as our setting, considering its significant improvements in dynamic illumination, geometric detail, and overall scalability.

To improve user accessibility, we further optimize the communication module between virtual sensors and the UAV flight controller while providing an open external interface such as Vision-Language-Action (VLA) [25] for flexible integration. In the current system, the external model can return high-level control commands or a direct target position. These values are then parsed by our custom flight control module into the corresponding thrust and torque for each of the four rotors.

Compared with other platforms such as AirSim and UnrealZoo, we independently compile a flight control module and integrate it deeply into UE. This design allows us to simulate various types of drone through a simple user prompt that can enhance the generalization and flexibility of our system. More details of the implementation are provided in the appendix, as they fall beyond our primary focus on omnidirectional simulation.

3.2. Data Collection Toolkit

Apart from online simulation, AirSim360 provides a powerful toolkit for offline data collection to overcome scale limitations of existing omnidirectional datasets, particularly in pixel-level annotations that are closely related to semantic and geometric understanding.

3.2.1. Render-Aligned Data and Label Generation

Inspired by the stitching process of omnidirectional cameras, our data, whether the original images or their corresponding ground truth, are generated by stitching six cube-face views together. Each view corresponds to one of the six directions, namely front, back, left, right, up, and down, with a 90-degree field of view. Specifically, we obtain six non-overlapped cube images $I_c^{6\times H_c\times W_c}$ as input and then produce an equirectangular projection map (ERP) $I_e^{H_e\times W_e}$. Here, H_c, W_c, H_e and W_e are the height and width of the cube and erp images, respectively. For clear illustration of such a process, we use the spherical projection which serves as a bridge between the input and output representations in Figure 3. The detailed equation is in our appendix.

In the following, we introduce the unique challenges posed by our stitching process for each data type, along with the corresponding solutions.

Panoramic Image: Capturing multimodal data simultaneously from six directions significantly increases GPU rendering load and storage pressure,

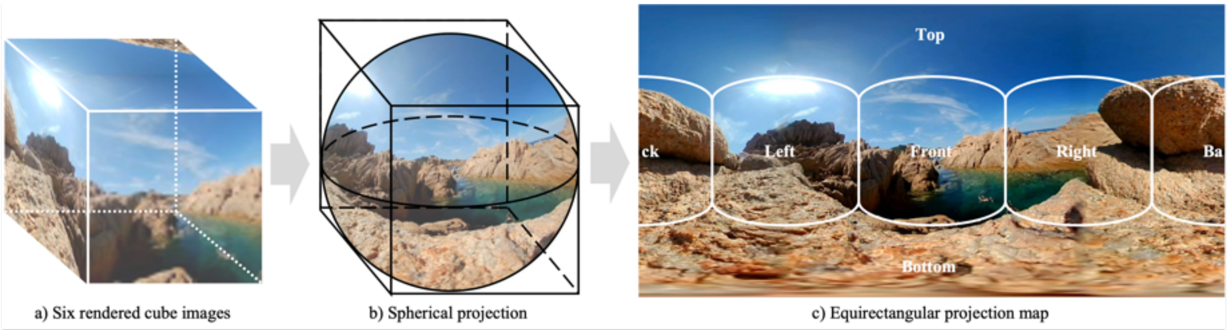

Top
ck
Left
Front
Right
Ba
Bottom
a) Six rendered cube images
b) Spherical projection
c) Equirectangular projection map

Figure 3. The equirectangular projection serves as a standard representation for panoramic imagery. In our implementation, we configured six identically calibrated cameras facing different directions. Since these are idealized pinhole cameras without lens distortion in the simulation environment, their outputs can be seamlessly stitched into a corresponding panoramic image through camera stitching algorithms.

resulting in substantial frame rate drops and difficulty in maintaining high throughput at high resolutions. For multi-view image rendering, we have redesigned the composition mechanism for six camera-associated images from ground up. After establishing corresponding camera rendering views, we utilize internal functions of Render Hardware Interface (RHI) to perform GPU-side texture resource copying, enabling one-time stitching of the six images. This approach thus avoids the drawbacks of secondary stitching within material nodes via blueprints.

Depth Information: In a perspective image, depth is defined as the distance between a 3D point and the camera center along the optical axis. This definition differs in the omnidirectional view, where no tangent plane exists. Therefore, in our setting, depth is defined as the distance from the center of the camera to the point along the viewing ray.

Since Z-Depth can be directly acquired from the precomputed depth buffer (Z-Buffer) in the Unreal Engine, we proposed a material-based pipeline to extract depth data and render the results into a Render Target. This approach allows writing the depth information directly into the alpha channel of the corresponding image. During the actual rendering process, with known camera world coordinates and corresponding intrinsic and extrinsic parameters, the precise distance from the camera to each spatial point can be computationally determined.

Semantic Segmentation: We assign a semantic label to each pixel to enhance scene understanding. In our implementation within Unreal Engine, we represent a set of semantic categories by assigning specific RGB values. To achieve this, we utilize the Stencil Buffer from the graphics rendering pipeline to assign colors to the static mesh actuator. The Stencil Buffer stores an integer value between 0 and 255 for each pixel in the scene. Through custom-designed post-process material, these values are transformed into designated color outputs, facilitating the acquisition of semantic labels.

Entity Segmentation: Unlike other simulators that focus only on partial instances, we consider all entities present in the scene, ensuring complete coverage across the entire image. However, this design presents a challenge related to quantity limitations. The Stencil Buffer mechanism restricts the total number of assignable categories to 256, which is insufficient for the large number of objects typically involved in entity segmentation tasks, particularly in complex scenarios. To overcome this limitation, we develop a dedicated entity segmentation method capable of labeling all static mesh actors, skeleton mesh actors, and landscape elements within the scene. This approach allows us to obtain complete and fine-grained segmentation results for any given frame.

Synchronous Rendering among Various Sensors: Various sensors should capture data synchronously without latency. Considering that UAV-mounted cameras are in constant motion, we have deactivated the *Capture Every Frame* option for all rendering cameras, significantly reducing GPU resource consumption. Since we have custom-designed the cameras and various image sensors, we introduced an Event Dispatcher to synchronize all sensors with a unified trigger signal, enabling simultaneous acquisition of multiple data types.

3.2.2. Interactive Pedestrian-Aware System

Scenarios involving humans are critically important in low-altitude real-world environments. Therefore, simulating realistic human behavior is a key aspect of our platform. We begin it from three perspectives. First, we allow users to create a customizable number of pedestrians within a defined active area, automatically assigning various behaviors to

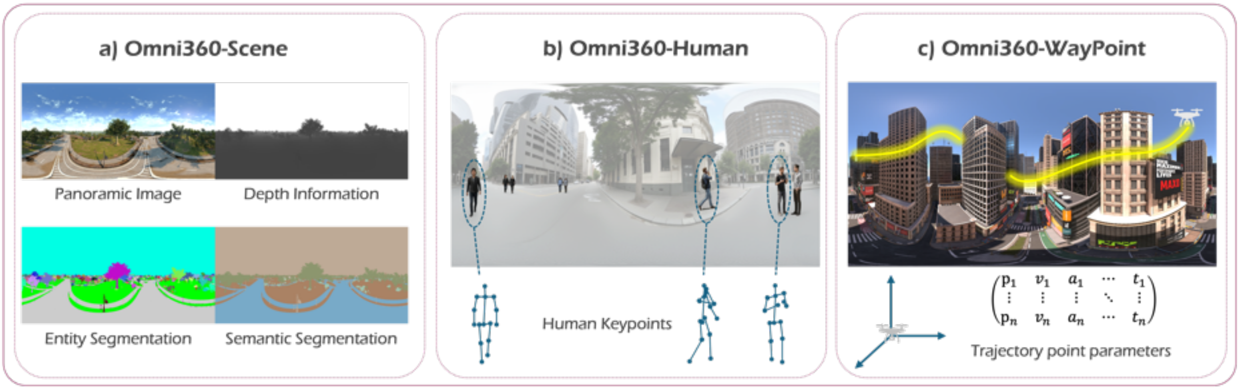


Figure 4. Visualization of Omni360-X. a) Visualizes Render-Aligned Data with corresponding labels (Sec. 3.2.1). b) Demonstrates pedestrian-aware label generation under panoramic views (Sec. 3.2.2). c) Shows trajectory synthesis from sparse waypoints using the Minimum Snap (Sec. 3.2.3).

them. Second, we enable autonomous interactions by combining NPC Behavior Trees with State Machines. For example, a pedestrian state machine can trigger transitions based on a multi-actor message dispatch and receive mechanism, allowing an agent to switch from a walking state to a chatting state when meeting another agent, or to randomly activate an OnPhoneCall state while walking (see Figure 5). Third, human keypoints are generated in real time as pedestrians interact, ensuring temporally consistent annotations for downstream perception tasks.

Generating pedestrian keypoint data presents two main challenges. First, a framework capable of supporting autonomous pedestrian motion is required to accommodate diversity in body movements. Second, a method for binding identical skeletal keypoints across different characters is essential to avoid positional inaccuracies introduced by manual keypoint annotation.

To address these issues, we design Interactive Pedestrian-aware System (IPAS), a system that enables autonomous and interactive movements among pedestrians. Additionally, we implement a Blueprint-based approach that invokes pre-existing universal skeletal points in the Skeletal Mesh via blueprint functions. For keypoints not included in the standard skeletal framework, we use the *Add Socke* method to incorporate them into the Skeleton Tree, thereby enabling users to access and output all designed body keypoint coordinates.

3.2.3. Automated Trajectory Generation Paradigm

The ground truth trajectories in previous platforms have largely relied on human control of the drone. To make the sampling process more efficient, we adopt Minimum Snap trajectory planning [34] into our simulation pipeline. A collector only needs to specify a few key waypoints within the scene, after which the system automatically generates a smooth and realistic trajectory that adheres to the dynamic constraints of the UAV. By adjusting parameters such as maximum velocity and acceleration, the resulting polynomial coefficients of the trajectories can be applied directly to real quadrotors.

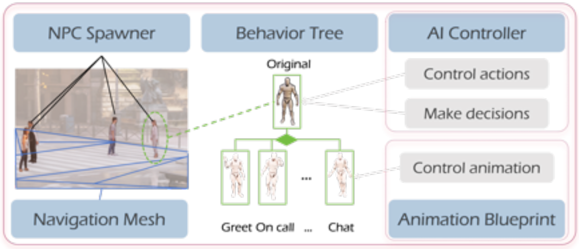


Figure 5. Conceptual diagram of the IPAS logic, showing the interplay between the Behavior Tree, State Machine, and the message-passing system for autonomous agent interaction.

4. Datasets Collection

Based on our proposed AirSim360 simulator, we collect several large-scale datasets for omnidirectional scene understanding. Then, we introduce the datasets gathered from a variety of Unreal Engine 5 (UE5) scenarios. Specifically, Omni360-X is a large-scale panoramic dataset containing 60K nonduplicated frames filtered using SSCD [38]. To serve diverse research objectives, Omni360-X is organized into three subsets, including Scene, Pedestrian, and Trajectory, where each emphasizes a specific aspect of 360-degree understanding. The visualization is shown in Figure 4.

More detailed information about the Omni360-X dataset is provided in the supplementary material due to page limitations.

Table 2. Composition of the Omni360-Scene Dataset. The dataset includes four Unreal Engine scenes, each depicting a unique urban environment. For scenes of varying scales, we collected panoramic data with corresponding labels, while the final column reports the total number of labels per scene.

Scenarios Name	Area (m^2)	Nums	Label types	Sem. Cat.
City Park	800,000	25,600	Dep/Seg/Ins	25
Downtown West	60,000	6,800	Dep/Seg/Ins	29
SF City	250,000	22,000	Dep/Seg/Ins	20
New York City	44,800	6,600	Dep/Seg/Ins	25

Omni360-Scene We focus primarily on scene parsing within Omni360-Scene. Inspired by ADE20K [59], Omni360-Scene contains more than 60K images annotated with depth and panoptic segmentation. Panoptic segmentation has two components: semantic and entity segmentation. For semantic labeling, we first extract semantic nouns from UE 5, which can be directly used for understanding open-world or open-vocabulary scenes. We then design a hierarchical semantic tree to ensure semantic consistency across different scenes. or entity segmentation, we further decompose stuff categories, such as trees and buildings, into individual entities. This is made possible by the simulator, which enables detailed entity separation that would be extremely challenging to achieve manually.

In Table 2, we present the data statistics for the four scenarios within Omni360-Scene. The dataset contains a total of 61,000 images captured from four scenarios, which demonstrate significant diversity in both physical scale (from 44,800 m^2 to 800,000 m^2) and semantic complexity (20 to 29 categories). All scenarios provide comprehensive ground truth for Depth, Semantic Segmentation, and Instance Segmentation

Omni360-Human To better understand pedestrian behavior, we adopt 3D monocular human localization as our primary task, as it measures both human positions and postures in the 3D world, which are closely related to behavioral understanding. Accordingly, Omni360-Human contains approximately 100K samples covering more than 10 pedestrian behaviors across diverse camera distances and viewpoints in about 6 scenes.

As shown in Figure 4 (b), each sample contains both camera information, including the camera's absolute position and rotation angles in the world coordinate system, and pedestrian information, such as the locations and rotation angles of human keypoints.

Table 3. Summary of the Omni360-WayPoint dataset. Flight paths of different lengths and sampling rates were generated across four outdoor scenes with physically consistent UAV dynamics.

Thumbnail	Scenario	Scene type	Count	Spacing	Length range
	City Park	outdoor	20 000	0.5	[50, 150]
	Downtown West	outdoor	5 000	0.2	[20, 50]
	New York City	outdoor	5 000	0.2	[20, 50]
	SF City	outdoor	20 000	0.5	[50, 150]

Omni360-WayPoint Building upon our proposed automatic trajectory synthesis scheme, we will release Omni360-WayPoint, an open dataset containing over 100,000 UAV waypoints, as described in Table 3. The trajectories adhere to realistic flight dynamics and include route variants parameterized by different maximum level-flight speeds, enabling analyses across diverse kinematic regimes. Such datasets can serve as various targets. For example, they can provide physics-consistent supervision for perception and state estimation, and enable trajectory prediction and system identification for model-based control and reinforcement learning. Moreover, they align instruction–video–action for VLA training, strengthen 3D reconstruction and mapping with accurate poses.

5. Experiments

In this section, we conduct several ablation studies on the AirSim360 platform to demonstrate the effectiveness of our design choices. We then present and benchmark the sampled dataset in multiple tasks, indicating the benefits of our proposed simulator for real-world scenarios.

5.1. Ablation Study on AirSim360

We conduct several comparison experiments on a workstation equipped with an NVIDIA RTX 4060 Ti (8 GB), an Intel i7-14700, and 32 GB of RAM. The evaluation covers two operational modes: perspective camera rendering and panoramic image stitching and transmission. Across all settings, our system consistently achieves higher data throughput and improved frame rates than conventional approaches.

Each camera in the rendering engine is processed as an independent rendering pass. Because the cameras move in real time, we optimize their scene-

Table 4. Performance metrics for frame capture settings.

Capture Every Frame	FPS	GPU Time
Enable	20	54 ms
Disable	29	35 ms

Table 5. Comparison of frame rates across different platforms with 6 cameras.

Platform	Nums of Camera	FPS
Not Optimized	6	14
Airsim360	6	18

scanning pattern to improve efficiency. As shown in Table 4, this optimized scanning strategy yields a 45% increase in per-camera frame rate.

Given the panoramic design of our platform, we restructure the composition and data-transmission pipeline for the six camera feeds in the rendering engine. Leveraging the Render Hardware Interface (RHI) described in Section 3.2.1, we reimplement the C++ pipeline, increasing the frame rate from 14 to 18 FPS.

5.2. Monocular Pedestrian Distance Estimation

To assess the impact of synthetic data, we apply IPAS (view in Sec. 3.2.2) to Monocular Pedestrian Distance Estimation (MPDE) in Table 6. The generated Omni360-Human dataset provides 3D keypoint annotations in the world coordinate system. Using diverse simulated scenes for data augmentation, our approach achieves improved results on several MPDE benchmarks. The model is trained with Omni360-Human and nuScenes [6], and evaluated on Omni360-Human, nuScenes, KITTI [22], and FreeMan [49], reporting both Euclidean and angular distance errors following MonoLoco++ [5].

As shown in Table 6, training with the Omni360-Human dataset consistently improves performance across all three public test sets. The average angular error decrease from 21.21° to 17.02°, and the mean distance error from 0.484 m to 0.458 m. To better reflect real-world camera configurations, synthetic data are generated with a 20° pitch angle. The largest test set, FreeMan, exhibit the most significant improvement, with the angular error reduced from 17° to 11.6°. These results highlight the effectiveness of synthetic data in enhancing monocular pedestrian distance estimation.

Table 6. The MPDE results trained on two datasets, evaluated across three public benchmarks and one control set, *Omni360*. The *Omni360* refers to the dataset Omni360-Human.

Training Set	Test Set	Dist. Err	Ang. Err	Ang. Err (Pub)	Dist. Err (Pub)
nuScenes	*nuScenes*	1.078	31.90	21.21	0.484
	KITTI	0.822	31.50		
	FreeMan	0.260	17.00		
	Omni360	2.439	33.30		
nuScenes* + *Omni360	*nuScenes*	1.068	30.70	**17.02**	**0.458**
	KITTI	0.809	31.20		
	FreeMan	0.228	11.60		
	Omni360	1.779	15.20		

Table 7. Quantitative comparison for depth estimation. We fine-tune on the UniK3D model and compare the performance of training on Deep360 versus our Omni360 dataset. ↓ and ↑ denote lower or higher is better.Best results are in bold.

Experiment Type	Training Data	Evaluation Data	AbsRel ↓	RMSE ↓	δ_1 ↑
Out-of-Domain	Deep360	SphereCraft	8.2570	0.0566	0.3490
	Omni360 (Ours)	SphereCraft	**5.4372**	**0.0435**	**0.3990**
Cross-Domain	Deep360	Omni360 (Ours)	0.3600	0.0714	0.4896
	Omni360 (Ours)	Deep360	**0.1762**	**0.0229**	**0.6672**

Table 8. Results of panoramic segmentation. The mIoU and mAP metrics are used to evaluate performance on the semantic and entity segmentation tasks, respectively.

Task	WildPASS	Omni360-Scene	Performance
Semantic	✓	○	58.0
	✓	✓	67.4
Entity	✓	○	24.6
	✓	✓	38.9

5.3. Panoramic Depth Estimation

To evaluate the generalization and domain adaptability of our Omni360 dataset, we conduct out-of-domain and cross-domain experiments for panoramic depth estimation. All datasets consist of outdoor synthetic data. The UniK3D model [36] is used as the baseline and fine-tuned after its official implementation. We compare the results with the Deep360 dataset [29], using Absolute Relative Error(AbsRel), Root Mean Squared Error(RMSE), and a percentage metrics δ_1, where i = 1.25, as evaluation metrics.

Out-of-Domain: Models trained in Deep360 and Omni360 are tested on SphereCraft [21]. In Table 7, the model trained on Omni360 achieves better results, showing improved generalization to unseen environments.

Cross-Domain: We perform cross-domain tests between Deep360 and Omni360. The model trained on Omni360 performs best, indicating stronger robustness and more transferable representations.

Table 9. Results of panoramic VLN. Three standard metrics, Success Rate (SR), Success weighted by Path Length (SPL), and Navigation Error (NE), are used to evaluate model performance.

Model	SR	SPL	NE
qwen2.5-vl-72b-instruct	0.4	0.3843	18099.73
qwen3-vl-plus	0.0	0.0	11436.97
qwen3-vl-flash	0.2	0.1945	9506.26
doubao-seed-1-6-251015	0.5	0.4813	10573.89

5.4. Panoramic Segmentation

In Table 8, we evaluate semantic and entity segmentation [41, 42] in the WildPASS validation set [55] using OOOPS [57] and Mask2Former [11]. With the incorporation of our Omni360-Scene segmentation data, both tasks achieve substantial performance gains, highlighting the effectiveness of large-scale data scaling for pixel-level semantic understanding.

5.5. Benchmark on Omni360-X

5.5.1. Panoramic Vision-Language Navigation

Vision-Language Navigation (VLN) [2] for unmanned aerial vehicles (UAVs) represents a challenging yet crucial direction toward embodied intelligence in aerial agents, , requiring joint understanding of spatial scenes and language instructions. Most existing UAV-VLN methods mostly rely on forward-facing cameras, resulting in a narrow field of view and large blind areas that limit target search efficiency and situational awareness.

We introduce a panoramic UAV-VLN task built on our simulation platform, where the UAV perceives the environment through panoramic vision instead of a single forward camera.

A set of preliminary experiments is conducted using several vision–language models to assess their ability to interpret panoramic observations for navigation.

In this section, to demonstrate the You Only Move Once (YOMO) capability of panoramic UAVs in decision-making tasks, we place most targets as salient objects and designed short flight trajectories. Under these conditions, the panoramic UAV can directly move toward the goal without additional yaw rotations or exploration. The similar SPL and SR values reported in Table 9 confirm that panoramic perception enables near-optimal, single-step decision making when the UAV is close to the target, validating the YOMO principle.

6. Conclusion

To address the issue of lacking large-scale omnidirectional data, we propose AirSim360, a panoramic simulation platform built on Unreal Engine. Inspired by the key elements in 4D real world, AirSim360 focuses on three aspects. First, we introduce a render-aligned data pipeline for generating 360-degree images with pixel-level depth and segmentation annotations in ERP representation. Second, we develop an interactive pedestrian-aware system that enables the study of human behavior through 3D monocular human localization. Third, we present an automated trajectory-generation paradigm that produces realistic aerial trajectories for navigation tasks. Built on top of AirSim360, we collect more than 60K non-duplicate frames. Ablation studies on the UE 5-based design, along with extensive experiments on the collected data, demonstrate the effectiveness of our platform and the benefits of the resulting dataset.

References

[1] Md Manjurul Ahsan, MA Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta, and Zahed Siddique. Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, 9(3):52, 2021. 2

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 9

[3] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010. 3

[4] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 3

[5] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Perceiving humans: from monocular 3d localization to social distancing. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 8

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 8

[7] Angelo Cangelosi, Josh Bongard, Martin H Fischer, and Stefano Nolfi. Embodied intelligence. In *Springer handbook of computational intelligence*, pages 697–714. Springer, 2015. 2

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 3

[9] Shih-Hsiu Chang, Ching-Ya Chiu, Chia-Sheng Chang, Kuo-Wei Chen, Chih-Yuan Yao, Ruen-Rone Lee, and Hung-Kuo Chu. Generating 360 outdoor panorama dataset with reliable sun position estimation. In *SIGGRAPH Asia*. 2018. 3

[10] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 3

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 9

[12] Shih-Han Chou, Cheng Sun, Wen-Yen Chang, Wan-Ting Hsu, Min Sun, and Jianlong Fu. 360-indoor: Towards learning real-world objects in 360° indoor equirectangular images. In *WACV*, 2020. 3

[13] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021. 2

[14] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 518–533, 2018. 2

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[16] Xin Deng, Hao Wang, Mai Xu, Yichen Guo, Yuhang Song, and Li Yang. Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution. In *CVPR*, 2021. 3

[17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, 2017. 2, 3

[18] Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Wang. Aerial vision-and-dialog navigation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3043–3061, 2023. 3

[19] Haoran Feng, Dizhe Zhang, Xiangtai Li, Bo Du, and Lu Qi. Dit360: High-fidelity panoramic image generation via hybrid training. *arXiv preprint arXiv:2510.11712*, 2025. 2

[20] Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, et al. Openfly: A comprehensive platform for aerial vision-language navigation. In *arXiv*, 2025. 2, 3

[21] Christiano Gava, Yunmin Cho, Federico Raue, Sebastian Palacio, Alain Pagani, and Andreas Dengel. Spherecraft: A dataset for spherical keypoint detection, matching and camera pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4408–4417, 2024. 8

[22] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 8

[23] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6927–6935, 2019. 3

[24] Wouter Jansen, Erik Verreycken, Anthony Schenck, Jean-Edouard Blanquart, Connor Verhulst, Nico Huebel, and Jan Steckel. Cosys-airsim: a real-time simulation framework expanded for complex industrial applications. *arXiv preprint arXiv:2303.13381*, 2023. 2, 3

[25] Sicong Jiang, Zilin Huang, Kangan Qian, Ziang Luo, Tianze Zhu, Yang Zhong, Yihong Tang, Menglin Kong, Yunlong Wang, Siwen Jiao, et al. A survey on vision-language-action models for autonomous driving. *arXiv preprint arXiv:2506.24044*, 2025. 4

[26] Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. Citynav: Language-goal aerial navigation dataset with geographic information. *arXiv preprint arXiv:2406.14240*, 2024. 3

[27] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024. 2, 3

[28] Haodong Li, Wangguangdong Zheng, Jing He, Yuhao Liu, Xin Lin, Xin Yang, Ying-Cong Chen, and Chunchao Guo. Da: Depth anything in any direction. *arXiv preprint arXiv:2509.26618*, 2025. 2

[29] Ming Li, Xueqian Jin, Xuejiao Hu, Jingzhao Dai, Sidan Du, and Yang Li. Mode: Multi-view

omnidirectional depth estimation with 360° cameras. In *European Conference on Computer Vision (ECCV)*, 2022. 8

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[31] Xin Lin, Xian Ge, Dizhe Zhang, Zhaoliang Wan, Xianshun Wang, Xiangtai Li, Wenjie Jiang, Bo Du, Dacheng Tao, Ming-Hsuan Yang, and Lu Qi. One flight over the gap: A survey from perspective to panoramic vision, 2025. 3

[32] Xin Lin, Xian Ge, Dizhe Zhang, Zhaoliang Wan, Xianshun Wang, Xiangtai Li, Wenjie Jiang, Bo Du, Dacheng Tao, Ming-Hsuan Yang, et al. One flight over the gap: A survey from perspective to panoramic vision. *arXiv preprint arXiv:2509.04444*, 2025. 2

[33] Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15384–15394, 2023. 3

[34] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011. 6

[35] Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3d environments with visual goal prediction. *arXiv preprint arXiv:1809.00786*, 2018. 3

[36] Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unik3d: Universal camera monocular 3d estimation, 2025. 8

[37] Giovanni Pintore, Eva Almansa, Marco Agus, and Enrico Gobbetti. Deep3dlayout: 3d reconstruction of an indoor layout from a spherical panoramic image. *TOG*, 2021. 3

[38] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *CVPR*, 2022. 6

[39] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023. 3

[40] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019. 2

[41] Lu Qi, Jason Kuen, Weidong Guo, Tiancheng Shen, Jiuxiang Gu, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High-quality entity segmentation. *arXiv preprint arXiv:2211.05776*, 2022. 9

[42] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8743–8756, 2022. 9

[43] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *ACM MM*, 2017. 3

[44] Ahmed Rida Sekkat, Yohan Dupuis, Pascal Vasseur, and Paul Honeine. The omniscape dataset. In *2020 IEEE International conference on robotics and automation (ICRA)*, pages 1603–1608. IEEE, 2020. 2

[45] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017. 3

[46] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics: Results of the 11th international conference*, 2017. 2, 3

[47] Qing Shi, Chang Li, Chunbao Wang, Haibo Luo, Qiang Huang, and Toshio Fukuda. Design and implementation of an omnidirectional vision system for robot perception. *Mechatronics*, 41:58–66, 2017. 2

[48] Oleksandra Sobchyshak, Santiago Berrezueta-Guzman, and Stefan Wagner. Pushing the boundaries of immersion and storytelling: A technical review of unreal engine. *Displays*, page 103268, 2025. 2

[49] Jiong Wang, Fengyu Yang, Bingliang Li, Wenbo Gou, Danqi Yan, Ailing Zeng, Yijun Gao, Junle Wang, Yanqing Jing, and Ruimao Zhang. Freeman: Towards benchmarking 3d human pose estimation under real-world conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21978–21988, 2024. 8

[50] Sijie Wang, Siqi Li, Yawei Zhang, Shangshu Yu, Shenghai Yuan, Rui She, Quanjiang Guo, JinXuan Zheng, Ong Kang Howe, Leonrich Chandra, et al. Uavscenes: A multi-modal dataset for uavs. In *ICCV*, 2025. 3

[51] Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Wenjun Wu, Hongsheng Li, Yue Liao, and Si Liu. Towards realistic uav vision-language navigation: Platform, benchmark, and methodology. *arXiv preprint arXiv:2410.07087*, 2024. 3

[52] Hang Xu, Qiang Zhao, Yike Ma, Xiaodong Li, Peng Yuan, Bailan Feng, Chenggang Yan, and Feng Dai. Pandora: A panoramic detection dataset for object

with orientation. In *European conference on computer vision*, pages 237–252. Springer, 2022. 3

[53] Menghan Xu et al. Predicting head movement in panoramic video: A deep reinforcement learning approach. In *CVPR*, 2018. 3

[54] Kailun Yang, Xinxin Hu, Luis M Bergasa, Eduardo Romera, and Kaiwei Wang. Pass: Panoramic annular semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 21(10):4171–4185, 2019. 3

[55] Kailun Yang, Jiaming Zhang, Simon Reiß, Xinxin Hu, and Rainer Stiefelhagen. Capturing omni-range context for omnidirectional segmentation. In *CVPR*, 2021. 3, 9

[56] Panagiotis G Zavlangas, Spyros G Tzafestas, and Kasper Althoefer. Fuzzy obstacle avoidance and navigation for omnidirectional mobile robots. In *European Symposium on Intelligent Techniques, Aachen, Germany*, pages 375–382, 2000. 2

[57] Junwei Zheng, Ruiping Liu, Yufan Chen, Kunyu Peng, Chengzhi Wu, Kailun Yang, Jiaming Zhang, and Rainer Stiefelhagen. Open panoramic segmentation. In *ECCV*, 2024. 9

[58] Fangwei Zhong, Kui Wu, Churan Wang, Hao Chen, Hai Ci, Zhoujun Li, and Yizhou Wang. Unrealzoo: Enriching photo-realistic virtual worlds for embodied ai. In *ICCV*, 2025. 2, 3

[59] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 7

Appendix

This Supplementary Material provides technical details, comprehensive dataset statistics, and additional implementation specifics that were omitted from the main paper due to space constraints. Specifically, we present the following information:

- In Section A, we provide some details of the three subsets of the Omni360-X dataset: Omni360-Scene, Omni360-Human, and Omni360-WayPoint, including the semantic categories and pedestrian behaviors.
- In Section B, we elaborate on the mathematical model and constraints of the Minimum Snap trajectory planning method used for automated trajectory generation.
- In Section C, we provide comprehensive experimental configurations for the Monocular Pedestrian Distance Estimation (MPDE) and Panoramic Vision-Language Navigation (VLN) tasks.

Also, please check *a recorded video* to obtain a brief description of our paper.

A. More Details of Omni360-X Dataset

A.1. Omni360-Scene Statistics

Omni360-Scene provides pixel-level annotations for depth information, semantic segmentation, and entity segmentation across diverse environments. As semantic complexity varies by scene, Table 10 provides a visual overview including the ERP image and semantic segmentation mask for each scenario, followed by a detailed breakdown of the semantic categories.

A.2. Omni360-Human Statistics

The Omni360-Human subset is dedicated to human-centric perception tasks, primarily monocular pedestrian distance estimation.

Data Statistics: Table 11 presents a detailed breakdown of the dataset composition. The data was collected across 6 distinct scenarios , covering a wide range of crowd densities and area sizes. The dataset includes over 100K frames in total, with varying numbers of NPCs to simulate realistic crowd dynamics.

A.3. Omni360-WayPoint Statistics

Omni360-WayPoint provides physics-consistent UAV flight paths for navigation, trajectory prediction, and control. The trajectories adhere to realistic flight dynamics derived from Minimum Snap planning. Table 13 and Table 12 details the key kinematic parameters and scale of the waypoint data.

$$\mathbf{S}(t) = \begin{bmatrix} \mathbf{p}(t)^T \\ \mathbf{v}(t)^T \\ \mathbf{a}(t)^T \end{bmatrix} = \begin{bmatrix} x(t) & y(t) & z(t) \\ v_x(t) & v_y(t) & v_z(t) \\ a_x(t) & a_y(t) & a_z(t) \end{bmatrix} \quad (1)$$

B. Minimum Snap Trajectory Planning Implementation Details

The Automated Trajectory Generation Paradigm employs Minimum Snap trajectory planning to produce smooth, dynamically feasible UAV flight paths from sparse user-defined waypoints. This method minimizes the integrated square of the fourth derivative of position (Snap), effectively ensuring trajectory smoothness and reduced control effort.

Polynomial Representation. Given a sequence of key waypoints $\{p_0, p_1, \ldots, p_M\}$, each segment of the trajectory is modeled as a fifth-order polynomial:

$$p_i(t) = a_{i,0} + a_{i,1}t + a_{i,2}t^2 + a_{i,3}t^3 + a_{i,4}t^4 + a_{i,5}t^5, \quad (2)$$

where $\mathbf{a}_i = [a_{i,0}, \ldots, a_{i,5}]^\top$ are the polynomial coefficients for segment i.

Optimization Objective. Following the Minimum Snap formulation, the smoothness of the trajectory is achieved by minimizing the integral of the squared fourth derivative (snap):

$$J = \int_{t_0}^{t_M} \left\| \frac{d^4 p(t)}{dt^4} \right\|^2 dt. \quad (3)$$

Quadratic Programming Formulation. The optimization problem can be expressed as a quadratic program:

$$\begin{aligned} \min_{\mathbf{a}} \quad & \mathbf{a}^\top Q \mathbf{a}, \\ \text{s.t.} \quad & A\mathbf{a} = b, \end{aligned} \quad (4)$$

where Q is derived from the cost in (3), and A, b encode waypoint and continuity constraints up to the third derivative. Solving this system yields the polynomial coefficients $\mathbf{a}$ defining the minimum-snap trajectory.

Table 10. Visualization of semantic segmentation and list of semantic categories for each scene.

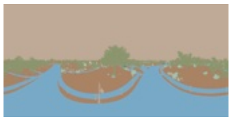
Scene Name	ERP Image	Semantic Vis	Semantic Categories
City Park			Building, Rock, AmurCork, Bush, Elm, Ivy, Maple, WeepingWillow, PlayGround, Bench, LampPost, FoodStalls, Cafechair, Roadblock, Trashcan, Trafficbarrel, Circlefence, Trafficlight, Water Plane, Road, Cafetable, Umbrella, Pool Sidewalk, Sky, Landscape
Downtown West			Building, Awning, Roof, Tree Generic, Tree Narrowleaf, Tree Pine, Prop Dining Table, Umbrella, Prop Dining Chair, Pot, Car Pillar, Recycle Bin, Food Cart, Bench Wood, Poster Stand, Ground Mod, Road, Lightpost Light Post, Tarppost, Light Streetlight Complete, Tarponly, Ground Park Walkway, Rock Rock, Background Mountains, Wood Fence Wood Fence, Prop Park Railing Rail, Prop Park Railing Pillar, Sky, Landscape
SF City			Building, Sidewalk, Road, Bus, Fence, Cone, Hydrant, Parkingmeter, Stopstation, Elecbox, Trash, Traffictube, Barrier, Alamppost, Blamppost, Lake, Bollardrope, Barricademetal, Tree, Sky
New York City			Concreteblock, Streetprops, Plasticcone, Metalfence, Pillar, Lamp, Trashcan, Postbox, Umbrella, Table, Chair, Greenpot, Roadcolumn, Adplane, Buidlingawning, Scaffolding, Usaflag, Plant, Ventilationtube, Building, Hotdogpot, Road, Sidewalk, Grounddirt, Sky

Table 11. Detailed Statistics of the Omni360-Human Dataset.

Scene	Subsets	Area Range $(m \times m)$	NPC Count (Min-Max)	Total Frames
New York City	14	$12 \times 12 \sim 30 \times 30$	$15 \sim 45$	29,000
LisbonDowtown	10	$12 \times 12 \sim 30 \times 50$	$10 \sim 45$	9,000
Downtown City	17	$12 \times 12 \sim 30 \times 30$	$8 \sim 30$	27,000
Roof	7	$12 \times 12 \sim 45 \times 20$	$5 \sim 30$	11,200
Rural Cabins	2	$15 \times 15 \sim 15 \times 30$	$7 \sim 14$	4,000
Rome	11	$8 \times 10 \sim 50 \times 30$	$4 \sim 30$	20,500
Total	**61**	-	-	**100,700**

Table 12. Introduction of Omni360-WayPoint. The Kinematic Parameters include two distinct sets of $a_{\max}$, $v_{\max}$, sampling interval t, each representing a typical UAV flight condition. The Total Number of Flight Paths is computed as the product of the number of Kinematic Parameter sets and the Number of Routes.

Scenario	Length Range	Kinematic Parameters	Number of Routes	Total Number of Flight Paths
City Park	[50, 150]	[(3, 16, 0.5), (5, 21, 1)]	20000	40000
Downtown West	[20, 50]	[(3, 16, 0.5), (5, 21, 1)]	5000	10000
New York City	[20, 50]	[(3, 16, 0.5), (5, 21, 1)]	5000	10000
SF City	[50, 150]	[(3, 16, 0.5), (5, 21, 1)]	20000	40000

Table 13. Inputs and outputs of the trajectory waypoints generation algorithm. The input $v_{\max}$ denotes the predefined maximum flight speed, $a_{\max}$ represents the maximum aircraft acceleration, and t is the sampling interval. The output parameters are also described in Eq. (1).

Minimum Snap	Parameters		
Input	$v_{\max}$	$a_{\max}$	t
Output	$\mathbf{p}(t)$	$\mathbf{v}(t)$	$\mathbf{a}(t)$

Dynamic Feasibility. To ensure physical feasibility, the trajectory is further constrained by dynamic limits on velocity and acceleration:

$$\|\dot{p}(t)\| \leq v_{\max}, \qquad \|\ddot{p}(t)\| \leq a_{\max}. \tag{5}$$

Each segment duration ΔT_i is automatically adjusted according to these limits, as well as the chosen sampling interval Δt, ensuring that the resulting trajectory remains dynamically executable by

Table 14. MPDE results across different training and testing datasets. **Dist. Err (All)** denotes the weighted average over all four datasets, where **Dist** refers to the Euclidean distance. **Ang. Err (All)** denotes the weighted average over all four datasets, where **Ang** refers to the angle. The **Pub** column in the last two columns indicates that the weighted average is computed over only the top three public datasets.

Training Set	Test Set	Dist. Err	Samples	Dist. Err (All)	Ang. Err	Ang. Err (All)	Ang. Err (Pub)	Dist. Err (Pub)
nuScenes	nuScenes	1.078	15369	0.80	31.90	23.14	21.207	0.484
	KITTI	0.822	1759		31.50			
	FreeMan	0.260	43361		17.00			
	Omni360-Human	2.439	11496		33.30			
nuScenes + Omni360-Human-all	nuScenes	1.073	15369	**0.43**	30.70	**16.25**	17.282	0.449
	KITTI	0.802	1759		32.70			
	FreeMan	0.213	43361		11.90			
	Omni360-Human	0.313	11496		10.80			
nuScenes + Omni360-Human-pitch_0	nuScenes	1.071	15369	0.50	30.70	19.08	19.194	**0.433**
	KITTI	0.812	1759		31.90			
	FreeMan	0.191	43361		14.60			
	Omni360-Human	0.868	11496		18.50			
nuScenes + Omni360-Human-pitch_20	nuScenes	1.068	15369	0.67	30.70	16.73	**17.023**	0.458
	KITTI	0.809	1759		31.20			
	FreeMan	0.228	43361		11.60			
	Omni360-Human	1.779	11496		15.20			

the UAV controller.

C. Experimental Details

This section provides additional implementation details and experimental settings for the Monocular Pedestrian Distance Estimation (MPDE) and Panoramic Vision-Language Navigation (VLN) tasks presented in the main paper.

C.1. Monocular Pedestrian Distance Estimation (MPDE)

In the Monocular Pedestrian Distance Estimation experiments, we design four sets of evaluations to demonstrate the effectiveness of our data. We first report the results on all test sets using only the nuScenes dataset. We then conduct a series of comparative experiments on three configurations of the Omni360-Human dataset: the full dataset, the subset with a pitch angle of 0°, and the subset with a pitch angle of 20°.

All models are trained using the AdamW optimizer with an initial learning rate of 0.002 and a weight decay coefficient of 0.01. The learning rate is multiplied by 0.98 every 300 steps during training.

The Omni360-Human training set is curated to exclude any samples from the Omni360-Human test set used in the experiments.

C.2. Panoramic Vision-Language Navigation (VLN)

In Visual Language Navigation (VLN), the formulation of prompts plays a critical role in determining evaluation metrics such as the Success Rate (SR). To ensure transparency and fairness in our experiments, we publicly release all prompts used in Table 15. It should be noted that, in addition to prompt formulation, factors including the frame rate of the simulator platform and the latency of online model invocation may also influence SR. The central aim of this work, however, is to introduce a highly challenging and promising new task based on the Airsim360 platform. Therefore, our experimental design prioritizes the most impactful factor, namely the formulation of prompts, while a comprehensive analysis of other variables remains outside the scope of this study.

Table 15. List of Prompts. The following prompts are used in two different environments, namely the New York City scene and the 1950s NYC Environment Megapack scene.

Prompt
Find the nearest traffic light and stop when you reach it.
Find the nearest blue mailbox and stop when you reach it.
Find the nearest tall building straight ahead and stop when you reach its rooftop.
Move forward, then at the intersection, you'll see a red telephone booth on your right. Stop near the closest one.
Fly across the lake in front of you, reach the opposite neighborhood, and stop on the street.
Find the nearby lake surrounded by woods and stop at the nearest shore.
Locate the building nearby with a giant Coca-Cola bottle decoration on its roof and fly close to the decoration.
Cross the zebra crossing and fly over this section of the road.
Fly to the small island in the center of the lake and land.
Fly straight ahead, turn right at the intersection, and stop near the bridge.
Fly along the current road and stop when you reach the second tree.
Stop near the small fountain located downstairs in the nearby building.
You are currently on the left side of the bridge. Now move to the right side and stop.
Climb over the fence in front of you and stop on the path in the park ahead.
Fly to the blue billboard on the building ahead and to your left, then stop nearby.
Fly to the tree with red leaves on the left side of the street and stop nearby.
Fly to the vicinity of the three very similar buildings straight ahead and stop.
Locate the billboard straight ahead featuring a person in a blue suit, fly to it, and stop nearby.
Fly to the billboard with the red car on the building ahead above you and stop nearby.
Find the floor in the building in front of you with red curtains and stop nearby.
Locate the billboard with the black and white portrait on the building ahead and stop nearby.
Arrive at the bank with the purple sign and stop downstairs.
Find the nearest yellow sunshade among the many downstairs and stop there.
Fly along the crosswalk over the intersection and stop on the opposite side of the road.
Find the nearest American flag and stop there.
Continue flying straight ahead and stop when you reach the intersection with the main road.
Fly to the blue barrier ahead and stop.
Fly to the red phone booth behind of you and stop when you are close the red phone booth.
Fly to the blue billboard on your right rear and stop when you're close to it.
Fly to the lake surrounded by trees on your right and stop when you're close to it.
Fly to the red bridge on your right and stop when you're close to it.
Stop near the nearest lawn.
Find the nearest traffic light and stop near it.
Navigate to the nearest green bike lane and stop.
Fly to the nearest billboard that shows BLACK & WHITE and stop nearby.
Navigate to the orange mailbox ahead and stop nearby.
Fly to the nearest food truck and stop.