# TUDelft

Delft University of Technology

A Framework for Fast Prototyping of Photo-realistic Environments with Multiple Pedestrians

Casao, S.; Otero, Andrés ; Serra Gomez, A.; Murillo, Ana C. ; Alonso-Mora, J.; Montijano, Eduardo

**Citation (APA)**
Casao, S., Otero, A., Serra Gomez, A., Murillo, A. C., Alonso-Mora, J., & Montijano, E. (2023). A Framework for Fast Prototyping of Photo-realistic Environments with Multiple Pedestrians. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2023)* (pp. 9083-9089). IEEE. https://doi.org/10.1109/ICRA48891.2023.10160586

**Important note**
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

# A Framework for Fast Prototyping of Photo-realistic Environments with Multiple Pedestrians

Sara Casao[1]    Andrés Otero[1]    Álvaro Serra-Gómez[2]
Ana C. Murillo[1]    Javier Alonso-Mora[2]    Eduardo Montijano[1]

*Abstract*— Robotic applications involving people often require advanced perception systems to better understand complex real-world scenarios. To address this challenge, photo-realistic and physics simulators are gaining popularity as a means of generating accurate data labeling and designing scenarios for evaluating generalization capabilities, e.g., lighting changes, camera movements or different weather conditions. We develop a photo-realistic framework built on Unreal Engine and AirSim to generate easily scenarios with pedestrians and mobile robots. The framework is capable to generate random and customized trajectories for each person and provides up to 50 ready-to-use people models along with an API for their metadata retrieval. We demonstrate the usefulness of the proposed framework with a use case of multi-target tracking, a popular problem in real pedestrian scenarios. The notable feature variability in the obtained perception data is presented and evaluated.

Fig. 1: Example of the scenarios obtained with the proposed framework. In this example, we use the randomness of pedestrian trajectories to obtain diverse data from the same scenario varying the lighting and weather.

## SUPPLEMENTARY MATERIAL

The framework code, models and generated datasets are available at https://github.com/saracasao/Pedestrian_Environment

## I. INTRODUCTION

Multiple problems addressed in robotics, such as tracking, navigation or mapping, entail the presence of people in their real-world applications [1]–[4]. Performing a systematic evaluation of such methods in environments with humans is a major problem. While datasets serve as a traditional option for ranking different approaches under the same testing conditions [5]–[8], their generation entails a resource-intensive process of data gathering and labeling. Besides, robots move, which requires online testing which makes it challenging to perform a rigorous sensitivity analysis. Photo-realistic simulators are becoming an increasingly popular tool to overcome these limitations [9], [10]. Unfortunately, the generation of complex pedestrian scenarios can still be a time-consuming solution with a steep learning curve, in addition to the tedious work of collecting and blending suitable actors (avatars, movements, textures, etc.).

This work presents a framework to easily generate realistic dynamic scenarios involving mobile robots and numerous pedestrians where researchers can mimic their target application conditions. Our framework is built on the photo-realistic

and open-source tools Unreal Engine [11] and AirSim [12]. It includes the necessary elements to make the integration smooth and also contains several ready-to-use examples of interest in different robotic applications, namely:

- A trajectory plugin implemented in Unreal Engine for a user-friendly definition of paths directly on the environment map. The plugin offers the capability to traverse the created paths in a random or customized fashion.
- A compilation of pedestrian models, created using open-source tools, ready to be used by simply dragging and dropping them on the map.
- A Python API to obtain the environment metadata from AirSim, providing automatic annotations of all the pedestrians present in the scene.

Figure 1 shows different examples of the scenes generated with the proposed framework. The randomness of pedestrian trajectories has been leveraged to obtain a large diversity of data within the same environment, exclusively varying light and weather conditions. To demonstrate the usefulness of the developed framework, we adopt the popular use case of multi-target tracking, which consists in determining the position of every person at all times. The evaluation videos are recorded using drones as moving cameras.

The rest of the paper is organized as follows. Section II describes the work related to the proposed framework. Section III gives all the details of the integration of the tools for creating the pedestrian scenario. Section IV demonstrates the large variability of features that could be easily obtained with the proposed framework and presents the evaluation of multi-target tracking methods with the acquired data. Finally, Section V presents the conclusions of the work.

[1] S. Casao, A. Otero, A.C. Murillo and E. Montijano are with RoPeRt group, at DIIS - I3A, Universidad de Zaragoza, Spain. {scasao, aotero, emonti, acm}@unizar.es
[2] A. Serra-Gómez, J. Alonso-Mora are with Cognitive Robotics, at TU Delft, The Netherlands. {A.SerraGomez@ , J.AlonsoMora@}@tudelft.nl
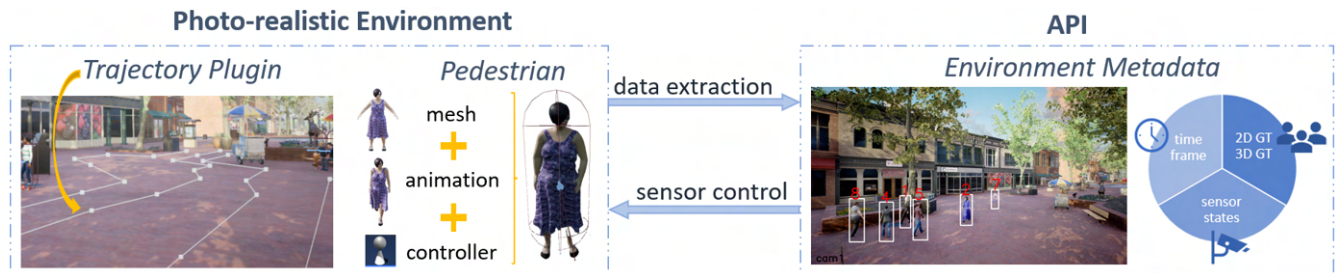
Fig. 2: Overall structure of the proposed framework. In the photo-realistic environment, the trajectory plugin provides a user-friendly way to define the path to follow for the pedestrian models (white lines). In addition, 50 pedestrian models are ready to be used, where the appearance (mesh), the movement (animation) and the pedestrian behavior (controller) are integrated. Finally, the API extract the environment metadata.

## II. RELATED WORK

Most studies that address issues related to autonomous robots concentrate on simulating crowds and individuals by depicting them as points [13], or they do not offer a wide range of appearance models [14], [15]. Unfortunately, these works suffer from a lack of perception information, making it highly challenging to create comprehensive solutions that integrate perception and robotic control. One of the biggest problems in tackling perception issues is the lack of data to work with, thus, slowing down research progress. Multiple benchmarks have been published focusing on the presence of people in the environment, with one of the most popular being the MOT Challenges. The MOT Challenges goal consists of improving the multi-target tracking task on a single static and moving camera [6], [16], [17]. Following this lead, other works, such as [18], make available new dense pedestrian crowd datasets, e.g., from 2 to 2.7 pedestrians per square meter, in addition to novel approaches for multi-target tracking. Aiming to broaden the posed problem of multi-target tracking to consider other issues like re-identification or consensus, a few works provide benchmarks with multiple overlapping cameras [5], [19].

In order to alleviate the huge resources cost of gathering and labeling real-world data, the use of photo-realistic simulators has become extremely popular for a wide variety of problems, from navigation [20] to cinematography [21]. Some works have leveraged the potential of these simulators to address end-to-end active tracking by training navigation policies based on reinforcement learning and giving the RGB image as the only input [1], [22]. Regarding the use of simulators for collecting synthetic data to address novel issues, new benchmarks have been published. The detection and tracking of occluded body joints along with a new dataset obtained from the *Grand Theft Auto V (GTA V)* videogame are presented in [23]. Taking advantage of the same simulator, [7] makes available a benchmark for multi-target tracking in a multi-camera system with and without overlapping cameras, while [8] with *NOVA* focuses on evaluating the robustness of tracking methods under adverse weather conditions. Other works, such as [24], [25], tap into the effortlessness of getting labeled data with simulators and propose domain adaptation algorithms from synthetic to real-world data for action recognition and re-

identification, respectively. The first one [24], renders people in *Blender*, whereas the second one [25] uses *Unreal Engine*, both of them gathering the required animations to create the people models from Mixamo [26]. Previous works on synthetic datasets rarely release the simulator they use, with the exception of [25] and [23]. Unfortunately, Mixamo no longer allows the use of its software for any machine learning or artificial intelligent tasks[1] and *GTA V* uses only static cameras with no option of working with drones inside the simulation. Recent works focus on making the simulator available and flexible to generate datasets [27] and test environments [28] for multiple computer vision applications. However, to the best of our knowledge, none of these works provides neither the required implementations to create fast and easily customized pedestrian scenarios nor releases open-source ready-to-use people models.

Our work leverages the benefits of Unreal Engine and AirSim to generate photo-realistic interactive environments with a focus on pedestrian scenarios, providing guidelines to generate challenging conditions for current algorithms. The effort required to develop a photo-realistic environment that enables dealing with any of the problems mentioned above is considerably high. For that reason, we present a framework that significantly reduces the workload for creating scenarios with moving pedestrians.

## III. FRAMEWORK

The framework presented in this work provides the essential tools to easily develop photo-realistic simulated pedestrian scenarios for robotic-oriented applications. Figure 2 shows the overall structure of the developed implementation. In the photo-realistic environment managed via Unreal Engine, the trajectory plugin tool gives the ability to create paths and control pedestrians to follow them in a random or customized fashion. We provide a collection of pedestrian models ready to use by simply dragging and dropping them into the scene. These models are composed of self-generated meshes integrated with a database of existing open-source animations, and controlled via the trajectory plugin. Finally, a Python API developed in AirSim extracts the environment metadata including images, pedestrian information ground truth, sensor states, and timestamps. The open-source feature

---

[1]https://www.adobe.com/legal/terms.html

of the framework allows adding new assets, such as other motion models or new meshes to tailor the scene to user preferences. In the following subsections, we explain in detail the different modules and the relations between them.

### A. Simulation Environment

Our framework is integrated within Unreal Engine to benefit from the extensive community and a large amount of freely available 3D models. Moreover, Unreal has already become a key tool within the robotics community, addressing tasks such as autonomous driving [29] or exploration with drones [27], making it a suitable choice for the application at hand.

*1) World description:* The first element to consider in the simulation is the world to place the pedestrians. For this part, we directly make use of existing environments at the marketplace and the options available within Unreal to configure them, i.e., lighting and weather conditions. For the sake of fast deployment of the proposed framework, the default installation from GitHub already includes one world ready to use. Nevertheless, changing this part is straightforward following the instructions provided by Unreal.

*2) Pedestrians:* The photo-realistic simulation of people entails a high complexity due to the high variability of appearance and intricate movements. Consequently, we focus on providing ready-to-use models of pedestrians. However, it is important to highlight that these models can be exchanged with other agents such as animals or vehicles by simply replacing the mesh and animation shown in Fig. 2. Our framework gathers a set of 50 ready-to-used rigged pedestrian models. These models consist of realistic human characters, encompassing a diverse cast, in terms of gender, ethnicity, height, or clothing, for a truthful simulation of a real-life environment.

The 3D human meshes are produced in *MakeHuman* [30], an open-source application that allows the mass production of people with random characteristics by taking into account different adjustable rules. To increase the diversity of human models, we use a community gallery included in the application to download *Creative Commons* assets, such as topologies, skin tones or clothes. The final people models that may not be realistic or appropriate have been manually filtered out.

Next, the pedestrians must mimic the action of walking within the environment, hence, the animations associated with this action have to be blended with the previous meshes. In our framework, the animations are from the CMU Graphics Lab Motion Capture Dataset [31], which consists of 2605 motion capture segments in different formats. Focusing on the walking animations, multiple motion capture segments have been assessed to select those most suitable and with sufficient quality for a photo-realistic environment. The final adaptation of these animations to a pedestrian movement has been done in *Blender*, where we discard the unrealistic frames in the merge of movement and mesh. Additionally, the frames attributable to the motion capture process are removed, such as the first moments of preparation of the
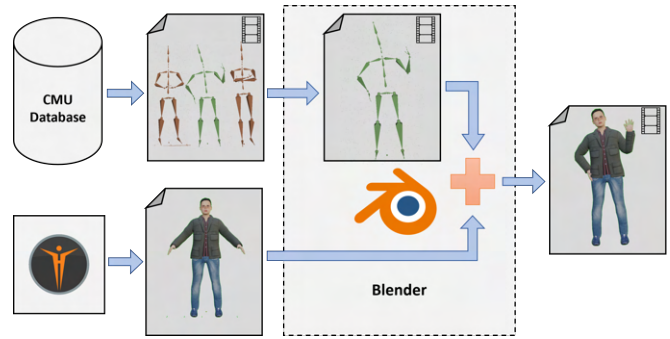


Fig. 3: Scheme of the characters' models obtention. A subset of animations from the CMU Motion Capture Database is downloaded, and a set of human 3D models are created with MakeHuman. The animations are trimmed to only contain valuable frames and then applied on the skeleton of a model via Blender. Finally, the model performing the animation is exported in Unreal Engine.

motion capture actor and those at the end of the movement. Lastly, the models in Blender are exported to Unreal where the integration of all the components, i.e., mesh, animation and controller, is performed. These Unreal pedestrian models are ready to use by simply dragging and dropping them on the world. Figure 3 shows a scheme of the process.

*3) Trajectory plugin:* The developed plugin provides multiple trajectory definition options, enabling the effortless creation of scenes with varying levels of complexity. Firstly, the *continuous path* tool enables the creation of specific trajectories to force characters to walk along them. In more intricate scenarios, i.e., urban areas with countless obstacles, this tool offers a user-friendly fashion to create routes that result in natural, realistic trajectories like walking along the sidewalk. The second tool, named *target points*, defines a set of goal positions that pedestrians can reach by crossing any area of the map, being especially useful for open and unconstrained environments.

Both of these tools are exclusively used to define the routes directly in the world. In order to select and walk towards a *continuous path* or a *target point*, the pedestrian models integrate an *AI Controller* that makes the person perform a trajectory in a random or customized manner. The former method seeks out goals within an area and selects one randomly, prompting the pedestrian to move in that direction until the goal is reached. Then, the search action is repeated to set the next goal. Regarding the customized mode, this option is more time-consuming due to requires specifying the goals for each pedestrian. The controller loads the targets tagged with the current pedestrian's name and scrolls the person through them chronologically from oldest to newest.

Moreover, the developed trajectory plugin includes a user-friendly way of generating scenarios where pedestrians follow the traced path by simply dragging and dropping the *continuous path* or the *target point* directly on the map. This avoids the tedious task of collecting the specific coordinates of the desired trajectory. An example of a different traced route with the *continuous path* can be seen in Figure 2.

### B. API for environment metadata capture.

The API implemented in AirSim performs the image acquisition, the automatic annotation, and the control of the
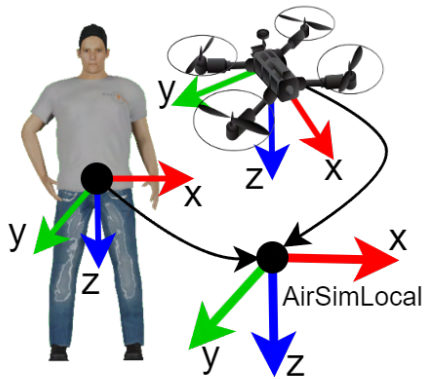
Fig. 4: Coordinate system of the framework. All acquired metadata is referenced in the *AirSimLocal* coordinate system.

cameras at the scene. AirSim is an open-source simulator built on Unreal Engine that aims to narrow the gap between simulation and reality for the development of autonomous vehicles, particularly drones and cars [12]. Different points of view can be used in the framework by defining fixed cameras, using drones as dynamic cameras, or associating cameras with pedestrians to obtain egocentric perspectives. The data annotation is conducted automatically by isolating the pedestrians from the rest of the environment with a unique key structure embedded in their Unreal identity. Every pedestrian model provided in this work shares this key structure to be easily found. Then, pedestrians' ground truth, including instance segmentation, 3D position, and 2D bounding boxes, are gathered in every iteration of metadata acquisition and saved in a JSON file for ease of use.

All the collected information is referenced to the *AirSimLocal* coordinates system, including the dynamic cameras which AirSim references to their initial position, Figure 4.

## IV. EXPERIMENTS

### A. Overview

This section analyzes the multi-target tracking problem as a use case that can benefit from the data generated with the presented framework. The high variability of features in the data generated within the simulated environment shows how it significantly aids in a more comprehensive evaluation of generalization compared to using static benchmarks. The supplementary material demonstrates how the framework is used to generate a new scenario and modify specific features of the scene.

We select two state-of-the-art tracking methods for the experiments, PHALP [32] and Tracktor++ [33]. Following standard evaluation procedures for object tracking, evaluation is performed offline on the automatically annotated sequences generated with our framework. To measure the tracker performance we use the CLEAR MOT Metrics [34] along with the Identity Metrics [35]. The Multiple Object Tracking Accuracy (MOTA) and ID F1 Score (IDF1) quantify two of the main aspects, namely, object coverage and identity. Note that the goal of the conducted experiments is not to determine the best tracker, but to demonstrate that the diverse features of the acquired set of data allow for isolating and identifying more challenging situations for the tracker.

### B. Acquired Datasets

This subsection details the sample datasets acquired from our framework to evaluate the selected tracking methods.

*Map:* To obtain datasets simulating real-world scenes, we have downloaded a free photo-realistic map with different areas, such as commercial streets or a park. This map has been released by the Unreal community and we have used it to create two different scenarios: *Font* and *Street*.

*Recordings:* Six simulated dynamic scenes of variable length are recorded, getting the labeling automatically from our framework python API. Our reference for the acquired
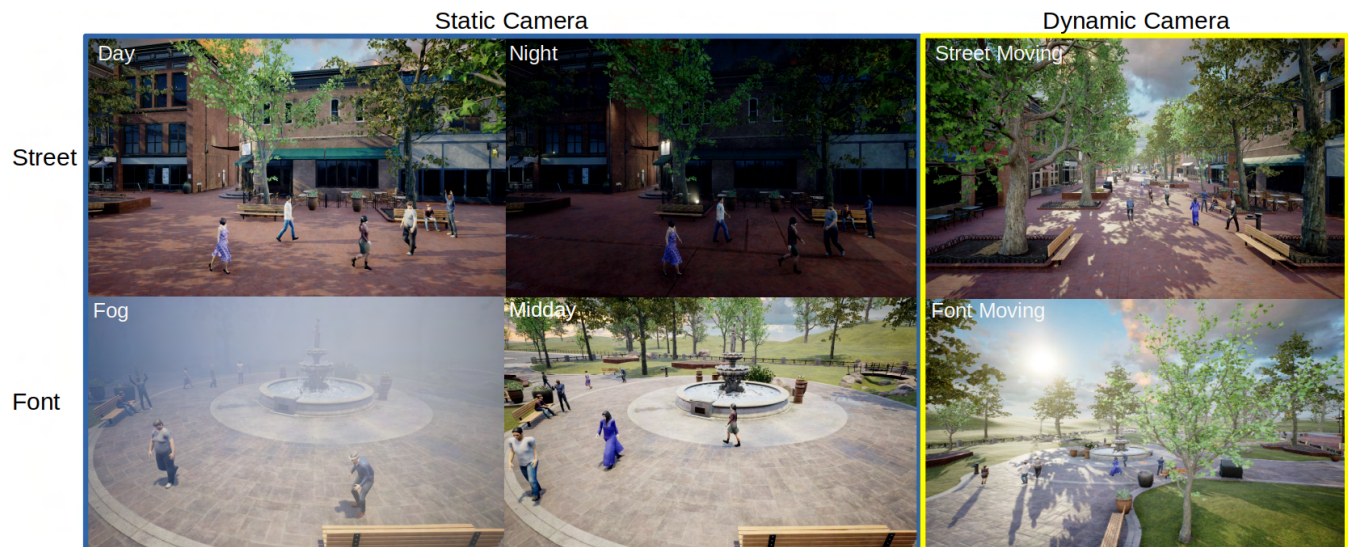


Fig. 5: Example of the acquired data with the presented framework. Each row is a scene used from the free downloaded map. The first and second columns are sequences recorded with static cameras, while the third column is acquired from a drone camera.

| Dataset | Camera | Length | Resolution | Pedestrian Trajectory | Scene | Description |
|---|---|---|---|---|---|---|
| *Day* | Static | 500 | 1920x1080 | Customized | Street | Pedestrian street at day time |
| *Night* | Static | 500 | 1920x1080 | Customized | Street | Pedestrian street at night time |
| *Fog* | Static | 900 | 1920x1080 | Random | Font | Fog weather in a small park |
| *Street Moving* | Dynamic | 500 | 1920x1080 | Customized | Street | Drone as moving camera flying over people |
| *Midday* | Static | 900 | 1920x1080 | Random | Font | Small park at midday |
| *Font Moving* | Dynamic | 600 | 640x480 | Random | Font | Low quality dataset with a drone as moving camera |

TABLE I: Details of the sample datasets acquired with the proposed framework for the tracking use case evaluation.

sets of data is the single-view MOT Challenge benchmarks [6], [16], [17], which release short sequences of fixed and moving cameras in real environments. Nevertheless, the photo-realistic simulator has the ability to generate data by combining its resources, such as multiple overlapping and non-overlapping cameras, or using dynamic cameras at the same time as static ones. The supplementary material includes a set of videos where the data acquisition on the proposed framework leverages these combined resources.

Therefore, we include sequences acquired from fixed and moving cameras, which are simulated with drones flying over pedestrians to elude the problem of obstacle avoidance. While the drone moves toward a target position, the API obtains the environment metadata including the extrinsic and intrinsic parameters of the camera. The presented set of data is summarized in Table I where each of the datasets is acquired under different light or weather conditions. The *Fog*, *Midday* and *Font Moving* datasets use the random trajectory of pedestrians while in *Day*, *Night* and *Day Moving* we create customized trajectories. All datasets except *Font Moving* capture high-quality images (1920x1080 resolution). *Font Moving* is captured to represent several challenging features and conditions for tracking algorithms: low-quality images, a moving camera, and a distant camera from the pedestrians.

Figure 5 shows images of each dataset where we can appreciate the high variability in terms of light, weather and viewpoint.

*C. Evaluation*

This evaluation focuses on analyzing the benefits that the proposed framework can bring to the tracking algorithms assessment. To perform the proposed analysis, we select two methods that address the object tracking task.

The first one is the state-of-the-art PHALP [32] that tracks people in a single view by predicting the 3D appearance, location and pose. Table II shows the comparison between their previous results and those obtained with the data acquired with our framework. The datasets PoseTrack [36], MuPoTS [37] and AVA [38] capture a diverse set of high-quality real sequences, including sports, casual interactions and movies. The drop in performance in some of our generated sequences, e.g., *Fog*, *Street Moving* and *Midday*, demonstrate that the method is able to deal with space-limited environments where pedestrians are close to the camera but its performance decrease in broad areas up to 11 points in MOTA and IDF1 for static cameras. This effect especially grows when a moving camera is used (*Street Moving*) causing a fall performance of up to 22 points in

MOTA. Figure 6 shows a comparison of the qualitative images resulting from this tracker on two scenes. On the left, is an example of the high performance of the method in *Day* dataset, and on the right, is an example on *Midday* where the failures are highlighted with red bounding boxes.



Fig. 6: Example of qualitative results with the tracker PHALP. Left: results on *Day* data. Right: results on *Midday* data. The failures (missed people) are highlighted with red bounding boxes. In this case, correspond with people with low resolution.

| Dataset* | PHALP [32] | | |
|---|---|---|---|
| | MOTA↑ | IDF1↑ | IDs↓ |
| PoseTrack [36] | 58.9 | 76.4 | 541 |
| MuPoTS [37] | 66.2 | 81.4 | 22 |
| AVA [38] | - | 62.7 | 227 |
| Day | 84.4 | 76.2 | 8 |
| Night | 83 | 82 | 2 |
| Fog | 48.2 | 52 | 21 |
| Street Moving | 36.3 | 52.8 | 4 |
| Midday | 48.8 | 51.5 | 15 |

TABLE II: PHALP tracking results in multiple datasets. *PoseTrack*, *MuPoTS* and *AVA* are the benchmarks used for its official evaluation. *Day*, *Night*, *Fog*, *Street Moving* and *Midday* are the datasets acquired with our framework. We can observe how these sets present a broader range of challenges that enables a more thorough evaluation of the tracker. * Sequence *Font Moving*, with poor quality images, is not in this analysis because it makes the algorithm produce unsatisfactory results.

The second method is the single view tracker Tracktor++ [33], whose results on our set of sequences and the MOT Challenges [16], [17] are shown in Table III. In this case, the lack of visibility is the major cause of failure, Figure7 shows a comparison of the qualitative results obtained with this method between *Fog* (on the left) and *Midday* (on the right). We include a second row with the ground truth to make it easier for the reader to find the pedestrians. In the sunny frame, most people are tracked satisfactorily, however in the foggy image, this precision drops, and only those close to the camera can be tracked.

We can deduce from the analyzed results that the proposed framework aids in the identification of open problems and the robustness evaluation against the open world.
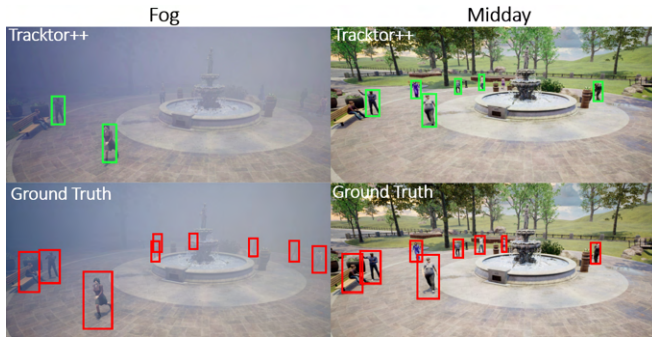
Fig. 7: Example of qualitative results with the tracker Tracktor++. First row: the bounding boxes depict the tracked pedestrians. Note the difference in performance between *Fog* (left) and *Midday* (right) due to the lack of visibility. Second row shows the ground truth to make the comparison easier for the reader.

| Dataset | Tracktor++ [33] | | |
| --- | --- | --- | --- |
| | MOTA ↑ | IDF1 ↑ | IDs ↓ |
| 2D MOT2015 [16] | 44.1 | 46.7 | 1318 |
| MOT16 [17] | 54.4 | 52.5 | 682 |
| MOT17 [17] | 53.5 | 52.3 | 2072 |
| Day | 31.5 | 44.0 | 18 |
| Night | 36.9 | 46.4 | 11 |
| Fog | 23.6 | 32.1 | 20 |
| Street Moving | 56.3 | 57.2 | 12 |
| Midday | 39.3 | 36.9 | 41 |
| Font Moving | 3.6 | 9 | 8 |

TABLE III: Tracktor++ results in multiple datasets. *2D MOT2015*, *MOT16* and *MOT17* are the benchmarks used for its official evaluation. *Day*, *Night*, *Fog*, *Street Moving* and *Midday* are the datasets acquired with our framework. We can also observe how these sets enable a more thorough evaluation of the tracker.

## V. CONCLUSIONS

The framework presented in this work provides all the essential tools to easily develop pedestrian scenarios for robotic-oriented applications. We believe that the released framework will significantly reduce the workload for robotic researchers to develop such environments. The implemented trajectory plugin is a user-friendly tool to create pedestrian paths and control simulated people to follow them in a random or customized fashion. Besides, to alleviate the tedious task of collecting and adapting people models, the framework compiles 50 pedestrian models ready to use by simply dragging and dropping. The provided API automatically gathers and annotates the pedestrians and cameras' metadata. The developed use case highlights how we can easily develop new benchmarks to perform more exhaustive evaluations of applications involving mobile robots and pedestrians, in particular towards robustness to real-world variations.

## VI. ACKNOWLEDGMENTS

## REFERENCES

[1] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. H. Bülthoff, M. J. Black, and A. Ahmad, "Active perception based formation control for multiple aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4491–4498, 2019.

[2] C. Cao, P. Trautman, and S. Iba, "Dynamic channel: A planning framework for crowd navigation," in *International Conference on Robotics and Automation*. IEEE, 2019, pp. 5551–5557.

[3] Y. Yue, C. Yang, J. Zhang, M. Wen, Z. Wu, H. Zhang, and D. Wang, "Day and night collaborative dynamic mapping in unstructured environment based on multimodal sensors," in *International Conference on Robotics and Automation*. IEEE, 2020, pp. 2981–2987.

[4] S. Casao, A. Naya, A. C. Murillo, and E. Montijano, "Distributed multi-target tracking in camera networks," in *International Conference on Robotics and Automation*. IEEE, 2021, pp. 1903–1909.

[5] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret, "Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5030–5039.

[6] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "Motchallenge: A benchmark for single-camera multiple target tracking," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 845–881, 2021.

[7] P. Kohl, A. Specker, A. Schumann, and J. Beyerer, "The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 4489–4498.

[8] A. Kerim, U. Celikcan, E. Erdem, and A. Erdem, "Using synthetic data for person tracking under adverse weather conditions," *Image and Vision Computing*, vol. 111, 2021.

[9] A. Devo, A. Dionigi, and G. Costante, "Enhancing continuous control of mobile robots for end-to-end visual active tracking," *Robotics and Autonomous Systems*, vol. 142, 2021.

[10] L. Chen, F. Liu, Y. Zhao, W. Wang, X. Yuan, and J. Zhu, "Valid: A comprehensive virtual aerial image dataset," in *International Conference on Robotics and Automation*. IEEE, 2020, pp. 2009–2016.

[11] "Epic Games Incorporated. unreal engine, 2022." https://www.unrealengine.com.

[12] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.

[13] F. Grzeskowiak, D. Gonon, D. Dugas, D. Paez-Granados, J. J. Chung, J. Nieto, R. Siegwart, A. Billard, M. Babel, and J. Pettré, "Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd," in *International Conference on Robotics and Automation*. IEEE, 2021, pp. 3879–3885.

[14] R. Alami *et al.*, "Hateb-2: Reactive planning and decision making in human-robot co-navigation," in *International conference on robot and human interactive communication (RO-MAN)*. IEEE, 2020, pp. 179–186.

[15] A. Favier, P.-T. Singamaneni, and R. Alami, "Simulating intelligent human agents for intricate social robot navigation," in *RSS Workshop on Social Robot Navigation*, 2021.

[16] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[17] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[18] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Densepeds: Pedestrian tracking in dense crowds using front-rvo and sparse features," in *RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2019, pp. 468–475.

[19] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[20] C. Vorbach, R. Hasani, A. Amini, M. Lechner, and D. Rus, "Causal navigation by continuous-time neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 425–12 440, 2021.

[21] P. Pueyo, E. Montijano, A. C. Murillo, and M. Schwager, "Cinempc: Controlling camera intrinsics and extrinsics for autonomous cinematography," in *International Conference on Robotics and Automation*. IEEE, 2022, pp. 4058–4064.

[22] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang, "End-to-end active object tracking and its real-world deployment via reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1317–1332, 2019.

[23] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *European Conference on Computer Vision*, 2018, pp. 430–446.

[24] V. G. T. da Costa, G. Zara, P. Rota, T. Oliveira-Santos, N. Sebe, V. Murino, and E. Ricci, "Dual-head contrastive domain adaptation for video action recognition," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1181–1190.

[25] T. Zhang, L. Xie, L. Wei, Z. Zhuang, Y. Zhang, B. Li, and Q. Tian, "Unrealperson: An adaptive pipeline towards costless person re-identification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 506–11 515.

[26] "Adobe Systems Incorporated. mixamo, 2022." https://www.mixamo.com.

[27] B. Alvey, D. T. Anderson, A. Buck, M. Deardorff, G. Scott, and J. M. Keller, "Simulated photorealistic deep learning framework and workflows to accelerate computer vision and unmanned aerial vehicle research," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3889–3898.

[28] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 902–919, 2018.

[29] B. Osiński, A. Jakubowski, P. Ziecina, P. Miłoś, C. Galias, S. Homoceanu, and H. Michalewski, "Simulation-based reinforcement learning for real-world autonomous driving," in *International Conference on Robotics and Automation*. IEEE, 2020, pp. 6411–6418.

[30] "Makehuman community. makehuman, 2022." http://www.makehumancommunity.org.

[31] Carnegie Mellon University Graphics Lab, "Cmu graphics lab motion capture database," The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

[32] J. Rajasegaran, G. Pavlakos, A. Kanazawa, and J. Malik, "Tracking people by predicting 3d appearance, location and pose," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2740–2749.

[33] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.

[34] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[35] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 17–35.

[36] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5167–5176.

[37] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 120–130.

[38] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6047–6056.