



HAL
open science

Simulating Aerial Event-based Environment: Application to Car Detection

Ismail Amessegher, Hajer Fradi, Clémence Liard, Jean-Philippe Diguët,
Panagiotis Papadakis, Matthieu Arzel

► **To cite this version:**

Ismail Amessegher, Hajer Fradi, Clémence Liard, Jean-Philippe Diguët, Panagiotis Papadakis, et al.. Simulating Aerial Event-based Environment: Application to Car Detection. European Robotics Forum 2024, Lorenzo Marconi, Mar 2024, Rimini, Italy. hal-04497648

HAL Id: hal-04497648

<https://hal.science/hal-04497648v1>

Submitted on 10 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Simulating Aerial Event-based Environment: Application to Car Detection

Ismail Amessegher¹, Hajer Fradi¹, Clémence Liard^{1,2}, Jean-Philippe Diguët²,
Panagiotis Papadakis¹, Matthieu Arzel¹

¹IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France

²CNRS, IRL CROSSING, Adelaide, Australia

Abstract. With the primary goal of enhancing the efficiency of drones for research and rescue missions through the exploitation of neuromorphic sensors and event-based vision, our focus in this work lies in setting up a simulated environment that can be used for synthetic data generation. In particular, we employ Unreal Engine to generate scenes suitable for the case of vehicle perception, followed by a dynamic event-based simulation environment in conjunction with AirSim and v2e tools. The synthetic event data acquired in this simulated environment serves as a crucial resource for training Artificial Intelligence (AI) systems, with a specific focus on car detection using YOLOv7.

Keywords: Drones, Simulation, Events, Synthetic Data, Car Detection, YOLOv7

1 Introduction

The increasing frequency of natural disasters calls for new solutions to improve the effectiveness and efficiency of disaster response services. According to a United Nations review, in 2019 only, around 400 climate-related and geophysical events resulted into tens of thousands of human losses. In this context, first-response by Search-And-Rescue (SAR) robots has shown its potential to operate in unstructured and hostile environments. A SAR team typically composed of aerial and/or ground vehicles operate remotely via a control post or center. To accomplish such missions, efforts must be directed towards multiple aspects, including:

- Safe movement: enhancing safety with onboard AI assistance that ensures reliable obstacle detection.
- Fast movement: achieving low-latency environment capture for more effective navigation and perception.
- Preserved energy autonomy: prioritizing artificial vision solutions that are conducive to low energy consumption.

Motivated by the factors mentioned earlier, SAR missions would significantly benefit from any technological shift that would combine higher energy efficiency

and improved navigation autonomy of drones. In this work, we particularly aim to investigate newly emerging bio-inspired sensors known as event or neuromorphic cameras [3]. Thanks to their unique properties including lightweight, energy-efficient, and high-frequency data acquisition, an increasing interest has been shown in exploiting these new sensors in the control of drones [2]. However, event-based vision requires additional training and dataset acquisition, which are more readily accessible in traditional RGB sensors. Since the drone calibration, validation, and AI training processes cannot be overly extensive, it becomes crucial to minimize any drone damage while maximizing exploration opportunities within a simulated environment. On this basis, our primary objective in this work is to create a simulated event-based environment. Throughout this paper, we will present tools that are intelligently integrated to create that. The simulated environment will serve us to generate synthetic event aerial data. An application using this data to train a custom car detector will be described in the last section before concluding.

2 Simulated Event-based Environment

In the literature, object detection or tracking by drones has been mostly performed using conventional frame-based cameras [5], [1]. Event cameras are newly emerging bio-inspired sensors that asynchronously measure per-pixel brightness changes. Known also as neuromorphic sensors, they are different from conventional frame-based sensors, where images at a fixed frame rate are captured. Their resulting output is a stream of events encoding multiple information at once such as time, position, and sign of the brightness changes. Event sensors have several advantages compared to conventional RGB cameras including high microsecond temporal resolution, very high dynamic range (140 dB vs. 60 dB), low power consumption, and high pixel bandwidth (in the order of kHz) [3]. Given all these advantages, these cameras have witnessed an increasing interest in drone applications.

At present, there is no aerial event dataset. In this work, we are opting to create data through simulation for multiple compelling reasons. This approach helps in preventing drone crashes, offers a more convenient alternative to creating data with real images, and allows for more accurate annotations in nearly infinite environmental scenarios. To this end, inspired by research works that use AirSim simulator [7] and Unreal Engine (UE), we build a simulated event-based environment by integrating an additional component which is v2e toolbox [4] for synthetic event data creation.

For more details, AirSim, which stands for Aerial Informatics and Robotics Simulation, is an open-source simulation platform developed by Microsoft for unmanned robots. AirSim is built using Unreal Engine as a graphics engine [6], which is highly appreciated for its power and versatility. UE boasts impressive graphics capabilities, stunning visual effects, and strong support for virtual and augmented reality. AirSim comes with photo-realistic camera, depth and thermal sensors. For event-based data, some tools are available to generate such data

from an RGB image capture. An alternative solution is to use v2e toolbox for converting conventional image data into event-based data [4]. This toolbox allows for adjust various parameters to the image conversion. It enables users to control timing, output video dimensions, and noise levels to simulate real-world event data collection conditions.

3 Data Generation

Since aerial images are expensive to be collected, and hard to be manually annotated, synthetic data becomes the most feasible solution to provide large amounts of automatically annotated data. For the task of car detection, the training data is created using the Modular Neighborhood pack ¹. The scene depicts a suburban neighborhood and contains vehicles, which are suitable for our application of car detection. For RGB images, the expected size is 640x480 and images need to be captured at a frequency of 200Hz. Since at this high acquisition frequency, data could be significantly blurred, it is worth making efforts in Unreal Engine and AirSim to slow down the simulation. Initially, we could capture images at a frequency of 20 Hz. By making time flow 10 times slower within the simulation, data capture would reach 200 Hz. AirSim allows us also to select the date at which the simulation takes place. This parameter enables data captures with four types of sunlight conditions: day, night, day followed by night, and night followed by day. This helps to test the recognition capabilities under varying lighting conditions.

To construct an image from raw events, various representations could be investigated [10]. In our approach, we opt to aggregate events within a temporal window of $\Delta t = 5$ ms, aligning with the initial frequency of the simulated environment. The resulting collected dataset comprises a total number of 10982 frames, with 7191 for training and 3791 for testing. The training set consists of 3216 samples acquired under daytime conditions and 3975 during the night. Similarly, the test set includes 2301 samples under daytime conditions and 1490 under nighttime conditions. For the annotations, bounding boxes with a size of less than 240 pixels are excluded. Figure 1 shows two sample images (daytime and nighttime), with their corresponding event frames and annotations.

4 Experiments

As for the task of car detection, we conduct our experiments on YOLOv7 detector [8]. We choose this detector belonging to YOLO family, since it has demonstrated its effectiveness for real-time applications, also more accurate results are obtained compared to previous versions [9]. It outperforms other real-time object detection models in both speed and accuracy. In the training phase, we use a mini-batch size equal to 64 and 30 epochs. The model is trained on NVIDIA

¹ <https://www.unrealengine.com/marketplace/en-US/product/modular-neighborhood-pac>

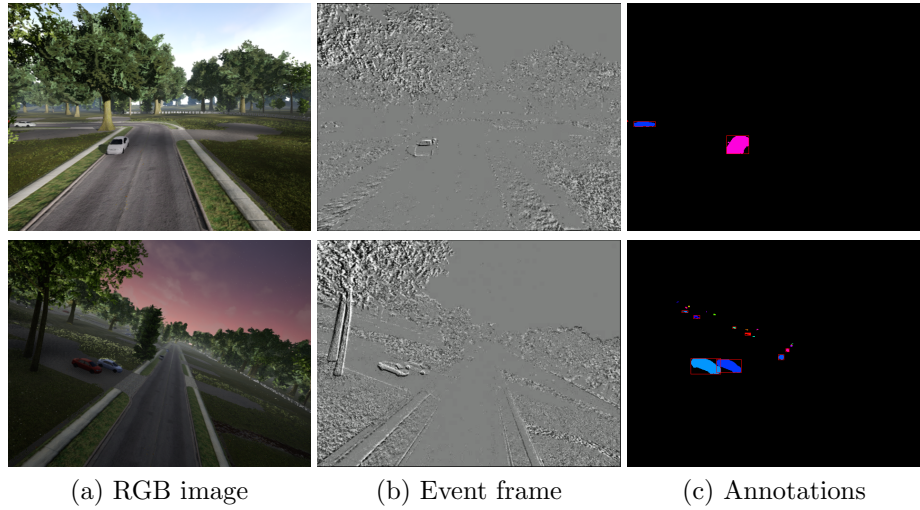


Fig. 1. Two sample images of the simulated dataset with their corresponding generated event frames and annotation boxes in the segmentation map. In the first row, a frame during the day, and in the second row, another sample under nighttime conditions.

RTX A6000 GPU. The performance of car detector in the test set of our created synthetic event data is evaluated in terms of mean Average Precision (mAP) of detections at Intersection Over Union (IOU) equal to 0.5 regarding the ground truth boxes. The achieved mAP stands at 0.63, with precision and recall both reaching 0.93 and 0.63, respectively. These outcomes affirm that we succeed in precisely detecting cars within the simulated data. However, there is room for improvement in recall. Enhancing True Positives by detecting more objects could be achieved by expanding the size of the training data.

5 Conclusion

In this work, we build a simulation environment based on Unreal Engine, AirSim and v2e toolbox to simulate dynamic aerial events. These tools are assessed in generating event-based data, which is used to train a custom car detection model. By means of experiments using YOLOv7 detector, the relevance of the simulated event data for training a custom model under different lighting conditions has been shown. As perspective, for the chosen Modular Neighborhood environment, despite its relevance to our case, but the overall visual variety of objects is relatively limited. Thus, this work could be potentially extended using other environments.

6 Acknowledgement

This work has received a French government support granted to the Labex CominLabs excellence laboratory and managed by the National Research Agency in the “Investing for the Future” program under reference ANR-10-LABX-07-01 from September 2022 to December 2024. In this program, the project associated to this work is *LEASARD* (Low-Energy deep neural networks for Autonomous Search-And-Rescue Drones).

References

1. Yaru Cao, Zhijian He, Lujia Wang, Wenguan Wang, Yixuan Yuan, Dingwen Zhang, Jinglin Zhang, Pengfei Zhu, Luc Van Gool, Junwei Han, et al. Visdrone-det2021: The vision meets drone object detection challenge results. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 2847–2854, 2021.
2. Rika Sugimoto Dimitrova, Mathias Gehrig, Dario Brescianini, and Davide Scaramuzza. Towards low-latency high-bandwidth control of quadrotors using event cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4294–4300. IEEE, 2020.
3. Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.
4. Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1312–1321, 2021.
5. Anitha Ramachandran and Arun Kumar Sangaiah. A review on object detection in unmanned aerial vehicle surveillance. *International Journal of Cognitive Computing in Engineering*, 2:215–228, 2021.
6. Andrew Sanders. *An introduction to Unreal engine 4*. CRC Press, 2016.
7. 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*, pages 621–635. Springer, 2018.
8. Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.
9. Hangyue Zhao, Hongpu Zhang, and Yanyun Zhao. Yolov7-sea: Object detection of maritime uav images based on improved yolov7. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 233–238, 2023.
10. Nikola Zubić, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. From chaos comes order: Ordering event representations for object recognition and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12846–12856, 2023.