Synthetic Dataset Generation for Object Detection in MAV Applications

Guilherme Santos Gabriel, Bruno Andreoni Sarmento*, Laisa Costa De Biase and Marcelo Knorich Zuffo
University of São Paulo, R. da Reitoria, 374, São Paulo

http://www.imavs.org/

ABSTRACT

The growing demand for accurate object detection across diverse applications has intensified the need for efficient and adaptable dataset generation techniques. In this study, we explore the use of synthetic datasets in the domain of Micro Air Vehicles (MAVs), addressing both indoor and outdoor detection tasks. We propose two procedural geometry-based approaches using Blender, a 3D modeling software, to simulate datasets for drone-based landing pad detection and animal detection in competition scenarios. These datasets were validated through participation in the Brazilian Robotics Competition 2023 and the IMAV 2024. Our results demonstrate that synthetic datasets can significantly reduce annotation effort, offer high adaptability, and maintain dataset quality, provided qualified personnel are available. We compare these synthetic approaches with traditional real-world dataset generation methods and highlight their potential to streamline dataset creation in robotics and AI.

1 INTRODUCTION

With the growing adoption of object detection technologies across various fields, the need for high-quality datasets has become increasingly critical. In particular, applications involving *Micro Aerial Vehicles* (MAVs), commonly known as drones, require task-specific datasets due to variations in operational scenarios and detection distances, thereby intensifying dataset development challenges.

The traditional process of building datasets consist in capturing real world images and manually annotating then, however it is too resource-intensive and time-consuming. This process becomes unfeasible for highly specialized situations (as with the MAV domain), thus some new processes seeks to bypass the real world interaction, with varying degrees of success.

We briefly present these different techniques and investigate the application of synthetic datasets created by 3D modeling and rendering. This investigation led to the development of two datasets validated in MAV competitions and also highlighted some strengths and weaknesses of the methods used.

*Email addresses: guilhermesangabriel@usp.br, bruno_as@usp.br

2 STATE OF THE ART

Recent research has explored the use of synthetic datasets as an alternative to traditional real-world data collection. One notable example is the *FireFly* dataset [1], developed with Unreal Engine 4 to simulate a forest environment with realistic fire-induced embers. Designed for firefighting-drone scenarios, this dataset demonstrated that synthetic data can achieve statistically comparable performance to real-world datasets, thereby validating its use in practical applications.

Synthetic datasets for object detection are typically generated using two main approaches: *image-based* and *procedural geometry-based* methods [2], as presented in Figure 1.

The image-based approach relies on existing images to generate new ones, often using generative techniques such as Transformer models, diffusion models, and autoencoders. While effective, these methods requires a realistic image dataset beforehand.

In contrast, the procedural-geometry approach uses 3-D modeling tools to create fully controllable and realistic scenes, offering greater flexibility in modifying elements such as lighting, backgrounds, camera positions, and object placements.

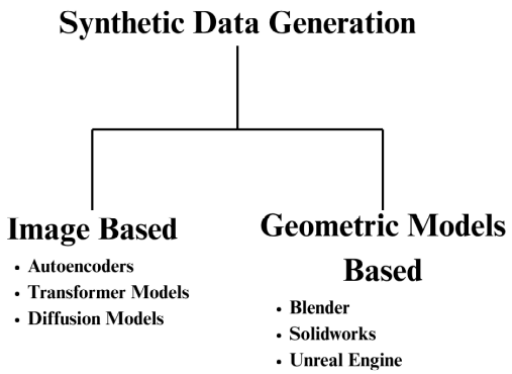

Figure 1: Taxonomy showing some options for synthetic data creation.

Game engines and 3D CAD softwares are required tools for the synthetic data generation:

- **Game engines** — Provide tools for procedural animation and scene variation. For instance, the *PeopleSansPeople* dataset generator [3] built on Unity enables semi-automated creation of labeled datasets for person detection and human-pose estimation.

http://www.imavs.org/

- **3-D CAD tools** — Offer object modeling and realistic rendering capabilities. We could cite the software: Blender, Maya, and SolidWorks Visualize [4].

Although these tools (Figure 1) are not traditionally intended for dataset creation, most of them exposes scripting interfaces that allow developers to design custom plug-ins and to automate the generation pipeline. In this context, we propose two geometry-based approaches for synthetic dataset generation tailored to indoor and outdoor MAV scenarios, each validated in real-world robotics competitions.

3 GEOMETRY-BASED SYNTHETIC DATASETS FOR MAV DETECTION

Two primary advantages of geometry-based synthetic datasets are (i) the elimination of manual annotation and (ii) the flexibility in scene generation.

Manual annotation is labor-intensive and prone to inconsistencies arising from varying labeling guidelines. In our approach, object dimensions and positions are extracted via the Blender API to automatically generate bounding boxes and metadata, ensuring both large dataset size and labeling consistency without human intervention. Additionally, the ability to manipulate scene elements, lighting, and camera parameters enhances the representativeness of the generated data. Assets such as 3-D objects and textures were sourced from free repositories like PolyHaven and BlenderKit.

3.1 Landing-Pad Detection Dataset (CBR 2023)

In the 2023 Brazilian Robotics Competition (CBR), the precision-landing challenge of the RoboCup Brazil Flying Robots Trial League required drones to do indoor detection of 1 m × 1 m landing pads, at distances ranging from 1 to 8 m. Figure 2 illustrates the landing pad used for the competition.

Figure 2: Landing pad pattern used in the CBR 2023 competition.

The dataset was generated using Blender's procedural-geometry pipeline, controlled via a Python script. Two techniques were employed:

Figure 3: Images created using technique 1.

Figure 4: Images created using technique 2.

1. **Background-texture variation:** The landing-pad model was rotated and repositioned across varying distances and angles relative to the virtual camera. High-dynamic-range imaging (HDRI) textures were proce-

durally rotated and translated to provide contextual variation [5]. Figure 3 presents some samples of the generated images.

2. **Scene-element variation:** The landing pad was placed on various planar terrains with randomized configurations of surrounding objects. Both terrains and objects were procedurally repositioned and rotated. Lighting was modeled with a combination of ambient illumination and directional sources whose color temperatures matched common indoor lights (e.g., fluorescent, halogen, LED). Figure 4 presents samples produced in this process.

Additional frames with close-up views and partial occlusions were rendered to reflect realistic proximity scenarios during landing. All annotations were generated automatically.

3.2 *Animal-Detection Dataset (IMAV 2024)*

The 2024 International Micro Air Vehicle (IMAV) competition posed an outdoor challenge requiring drones to detect zebra-striped animal markers (Figure 5) from a bird's-eye perspective.

Figure 5: Examples of Top-down Images of Zebra for the Census Task: © Yauheniya Piatrouskaya.

Using the presented technique, Blender scenes of zebra markers embedded in outdoor environments were populated with 3-D objects such as vehicles, vegetation, buildings, and other animals. Lighting conditions simulated the sun's trajectory to introduce temporal variation, and select frames included fog elements to emulate low-visibility conditions. All annotations were generated automatically via scripting.

4 RESULTS

4.1 *Hardware and software environment*

Training was executed on a workstation with Ubuntu 22.04 LTS (`x86_64`, kernel 6.5), Python 3.10.12, PyTorch 2.7.1 built with CUDA 12.6, and Ultralytics YOLOv8 v8.3.150 [6].

The available accelerator was an NVIDIA GeForce RTX 4060 providing 7 808 MiB of addressable memory.

The YOLOv8-nano backbone was fine-tuned for 200 epochs on $720 \times 720\,px$ images with a mini-batch of 16 using:

```
yolo detect train data=~/data.yaml \
                  model=yolov8n.pt \
                  epochs=200 imgsz=720 \
                  batch=16
```

A total of 1,883 procedurally rendered images were used for training and 401 real photographs (460 annotated landing-pad instances) for validation. The network converged in 17 epochs.

4.2 *Performance of the best epoch*

The obtained model achieved the following performance indicators:

- Precision: 0.926
- Recall: 0.895
- mAP @0.5: 0.966
- mAP @0.5:0.95: 0.804
- GPU memory: Under 3 GB
- Inference speed: 1.3 ms per frame on the RTX-4060

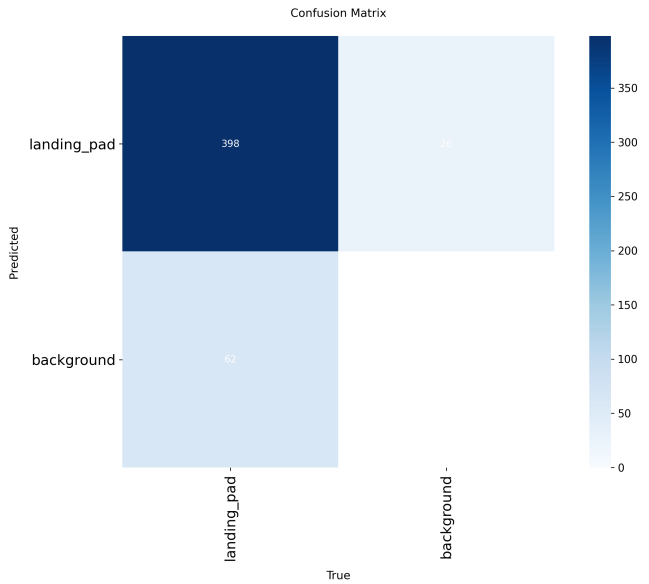

Figure 6: Confusion matrix for the real-image validation set (460 ground-truth instances).

http://www.imavs.org/

http://www.imavs.org/

Figure 7: Sample predictions on real validation images.

The confusion matrix (Fig. 6) indicates that **87** % of ground-truth objects were correctly detected (TP $= 398$ of 460), while **6** % of predictions were false positives (FP $= 26$ of $398 + 26$ predictions). The remaining 62 false negatives stem largely from extreme viewing angles and harsh shadows.

Figure 7 illustrates a subset of the real validation set used for quantitative evaluation. The detection results shown were produced by the model trained solely on synthetic images. Despite variability in lighting conditions, viewing angles, shadows, and partial occlusions, the model consistently localized the landing pads with accuracy and robust box placement.

Mean Average Precision (mAP) is adopted as the principal evaluation metric as it is the community standard for ranking detectors in major challenges [7]. Balancing false positives and false negatives, mAP integrates the entire precision–recall curve. Specifically, the COCO protocol computes AP at ten progressively stricter IoU thresholds (0.50 to 0.95 in 0.05 steps) and then averages them, causing the final score to reward both correct class prediction and tight box placement while avoiding the arbitrariness of a single IoU cut-off.

The choice to validate on real images instead of synthetic ones is deliberate. When training and validation come from the same graphics engine, a detector may achieve high scores simply by exploiting rendering regularities; such scores do not guarantee transfer to flight conditions and encourage over-fitting to synthetic artifacts. A real-image validation set—shot under varied lighting, backgrounds and viewing angles—acts as a proxy for deployment and makes the evaluation stricter. Taken together, the high detection metrics (Fig. **??**), low latency and small memory footprint demonstrate that a lightweight network trained exclusively on procedurally generated data generalizes effectively to real imagery and fulfils the accuracy and timing requirements of micro-air-vehicle applications.

The validation used in the CBR 2023 dataset was not applied to the one used in the IMAV 2024 competition. In this case, the effectiveness of the dataset could be determined anecdotally using the results of the actual competition. The Skyrats drone team, using the dataset in question, was able to detect all the zebra markers that the task positioned in the flying area (Fig. 8), with no false positives, even when other animals could be seen in the drone imaging.

Figure 8: 2 of 10 zebras detected at IMAV 2024 (conf. 0.73 & 0.76)

5 DISCUSSION

The dataset creation process revealed key distinctions between synthetic and real-world datasets:

- **Dataset localization** — not specifically related to the task presented in the CBR competition but common in other usages such as rescue missions in outdoors scenarios or the conservation situation presented in IMAV 2024 is the varied possible geographical locations. A real life dataset that tries to offer detection in forests may need the actual location, or something close to it, to be effective in its task. If this is not possible, a synthetic dataset can simulate the location needed provided that the designer knows the region well enough to account for the specifics of it;
- **Automated annotation** — as previously noted, the automated annotation process is a huge benefit of the method developed, skipping the need of an individual to not only label the images correctly, but also to establish and follow the dataset labeling guidelines;
- **Adaptability in new situations** — Another big advantage of the synthetic method provided comes from the adaptability component intrinsic of the method.

http://www.imavs.org/

If needed, as when an aspect of the dataset wasn't representative of real life situations, a new patch in the dataset can be developed relatively easily, without the need to do all the normal steps of data collection and labeling after the fact. This is particularly useful when prototyping a system that may need changes that couldn't be accounted for in the beginning or when mixing real life data with the specific needs of the application being currently developed;

- **Need for qualified individuals** — One of the challenges of the method developed is in how specific the toolset required to create it is. The individuals involved in the project had to have been exposed to the tool (Blender) before, and also needed to have a highly niche knowledge such as using the python API, something not part of the main workflow of the tool;
- **Number of individuals needed** — In a real life situation, the need for multiple pictures might force the dataset to be created by a collective of individuals concurrently so the process is faster. In the synthetic method proposed, even though the individuals need some relevant prior training in the tools involved, the actual number of individuals can be massively decreased, leading to a smaller human footprint on the project.

Table 1: Comparison Between Real-World and Synthetic Dataset Methods

Criterion	**Real-World Dataset**	**Synthetic Dataset**
Localization	Geographically constrained	Region-flexible (designer-dependent)
Annotation time	Manual / semi-automated	Fully automated
Adaptability	Slow, labor-intensive	Rapid and modular
Tool qualification	General labelling tools	3-D tools & scripting required
Team size	Larger teams	Smaller, specialised teams

6 CONCLUSION

This work demonstrates the feasibility and advantages of synthetic dataset generation for MAV applications using procedural geometry and open-source tools. The developed datasets addressed indoor precision-landing and outdoor marker-detection tasks and were validated in robotics competitions. Results show that synthetic datasets are not only viable substitutes but can surpass traditional methods in flexibility, annotation efficiency, and adaptability—despite requiring specific technical expertise.

Next steps to the work presented come with the development of a plug-in inside Blender specialized on helping with synthetic dataset creation. To better validate the results in these new cases, a comparison between the synthetic and a fully real-life based dataset could help elucidate how well both approaches function.

REFERENCES

[1] Yue Hu, Xin-Yu Ye, Yifei Liu, Souvik Kundu, Gourav Datta, Srikar Mutnuri, Namo Asavisanu, Nora Ayanian, Konstantinos Psounis, and Peter Anthony Beerel. Firefly: A synthetic dataset for ember detection in wildfire. *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3767–3771, 2023.

[2] Nishanth Nandakumar and Jörg Eberhardt. Overview of synthetic data generation for computer vision in industry. In *2023 8th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*, pages 31–35, 2023.

[3] Salehe Erfanian Ebadi, Y. C. Jhang, Alexander Zook, Saurav Dhakad, Adam Crespi, Pete Parisi, Steve Borkman, Jonathan Hogins, and Sujoy Ganguly. Peoplesanspeople: A synthetic data generator for human-centric computer vision. *ArXiv*, abs/2112.09290, 2021.

[4] Matej Arlovic, Davor Damjanovic, Franko Hrzic, and Josip Balen. Synthetic dataset generation methods for computer vision application. In *2024 International Conference on Smart Systems and Technologies (SST)*, pages 69–74, 2024.

[5] Ahmad Rafi, Musstanser Tinauli, and Mohd Izani. High dynamic range images: Evolution, applications and suggested processes. In *2007 11th International Conference Information Visualization (IV '07)*, pages 877–882, 2007.

[6] Rejin Varghese and Sambath M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6, 2024.

[7] R. Padilla, S. L. Netto, and E. A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *Proceedings of the 2020 International Workshop on Systems, Signal Processing and their Applications (IWSSIP)*, pages 237–242, 2020.