The Journal of Open Source Software

Syclops: A Modular Pipeline for Procedural Generation of Synthetic Data

Anton Elmiger [1¶], **Kai von Szadkowski** [1], **and Timo Korthals** [2]

1 German Research Center for Artificial Intelligence (DFKI), Germany **2** CLAAS E-Systems, Germany ¶ Corresponding author

DOI: 10.21105/joss.07854

Software

- Review
- Repository
- Archive

Editor: Sébastien Boisgérault

Reviewers:

- @bstanciulescu
- @sumn2u

Submitted: 28 January 2025
Published: 25 June 2025

Summary

Syclops is an open-source, modular pipeline for generating large-scale, photorealistic synthetic datasets with pixel-perfect ground truth annotations. Built on Blender's Cycles engine (Community, 2019), it offers a flexible framework for researchers in computer vision, robotics, and related fields. Key features include:

- **Plugin-based architecture** for easy extensibility
- **Procedural generation** of diverse, large-scale environments
- **Photorealistic rendering**
- **Multi-modal sensor simulation** (RGB cameras, depth sensors, stereo cameras)
- **Comprehensive ground truth annotations**
- **Dynamic scene configuration** using YAML
- **Scalability** for millions of objects

Syclops is especially useful when collecting real-world data is impractical due to cost or difficulty, making it a valuable tool for generating high-quality synthetic data.

Statement of Need

Machine learning models, particularly in computer vision and robotics, depend on the diversity and quality of training data. Real-world data collection is often expensive and challenging, especially for rare events (Tabkhi, 2022). Synthetic data generation offers an efficient alternative, producing large, annotated datasets (Mumuni et al., 2024).

Syclops addresses this need with its focus on large-scale, procedural scene creation—particularly for outdoor and agricultural scenarios. Compared to tools like Kubric (Greff et al., 2022), Blenderproc2 (Denninger et al., 2023), NViSII, NDDS, and iGibson, Syclops offers a YAML-based scene description that simplifies customization and reproducibility. The following table (Table 1) highlights key differences.

Table 1: Comparison of synthetic data tools with abbreviations: SS=Semantic Segmentation, IS=Instance Segmentation, D=Depth, OF=Optical Flow, SN=Surface Normals, OC=Object Coordinates, BB=Bounding Box, OP=Object Pose, V=Volume, KP=Keypoints, PS=Python Script, C=Camera, SC=Stereo Camera, L=Lidar

Tool	Rendering Engine	Scene Creation	Output Annotations	Sensors
Syclops	Blender Cycles	YAML	SS, IS, D, OF, SN, OC, BB, OP, KP, V	C, SC
Kubric	Blender Cycles	PS	SS, IS, D, OF, SN, OC, BB, OP	C
Blenderproc2	Blender Cycles	PS	SS, IS, D, OF, SN, OC, BB, OP	C, SC

Tool	Rendering Engine	Scene Creation	Output Annotations	Sensors
NViSII	Nvidia Optix	PS	SS, D, OF, SN, OC, BB, OP	C
NDDS	Unreal Engine 4	UE4 GUI	SS, D, BB, OP, KP	C
iGibson	PBR Rastering	PS	SS, IS, D, OF, BB	C, L

Key Features

1. **Large-scale Procedural Generation:**
   Efficiently create vast environments with millions of objects, ideal for outdoor settings such as agricultural fields.
2. **YAML-based Configuration:**
   Define and customize scenes easily with YAML syntax, enhancing reproducibility.
3. **Modular Architecture:**
   Extend functionality with plugins for custom scene elements, sensors, and outputs.
4. **Multi-modal Sensor Simulation:**
   Simulate various sensors (e.g., RGB and stereo cameras with projected light) for versatile data generation.
5. **Comprehensive Annotations:**
   Generate detailed ground truth data including segmentation, depth maps, object coordinates, bounding boxes, poses, keypoints, and volumes.
6. **Off-Highway Focus:**
   Special emphasis on agricultural and off-highway scenarios fills a niche in current synthetic data tools.

Architecture and Implementation

Syclops is implemented in Python and leverages Blender's Python API for scene creation and rendering (Figure 1 for an overview). Its architecture comprises:

- **Job Configuration:**
  YAML-based files define scene composition, sensor properties, and outputs.
- **Asset Management:**
  A module for organizing and accessing 3D models, textures, and materials.
- **Scene Generation:**
  Plugins efficiently place and manipulate large numbers of objects using Blender's Geometry nodes and object instancing.
- **Sensor Simulation:**
  Modules replicate various sensor modalities.
- **Output Generation:**
  Plugins produce sensor outputs and ground truth annotations.
- **Postprocessing:**
  Tools refine and process the generated data, enabling additional annotations and data augmentation.

For instance, Syclops can use convex decomposition for efficient rigid body simulation, allowing for dynamic scene interactions.

The Journal of Open Source Software

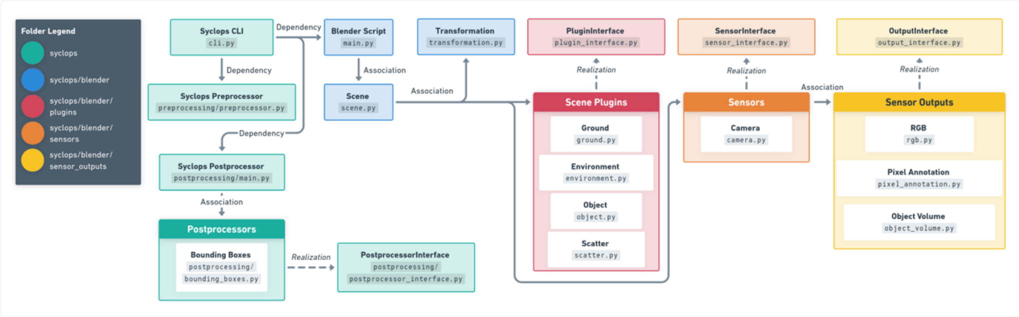

Figure 1: Architecture overview showing Syclops' components and their relationships.

Example Usage

A simple YAML configuration below demonstrates how to generate a synthetic dataset of RGB and depth images of trees scattered on a flat ground:

```yaml
# job_config.yaml
general:
  steps: 100
  seeds:
    numpy: 42
    cycles: 42

scene:
  syclops_plugin_ground:
    - name: "Ground"
      size: 50
      texture: "Example Assets/Muddy Dry Ground"
      class_id: 1

  syclops_plugin_scatter:
    - name: "Trees"
      models: "Example Assets/Trees"
      floor_object: "Ground"
      density_max: 0.1
      class_id: 2

sensor:
  syclops_sensor_camera:
    - name: "main_camera"
      frame_id: "camera_link"
      resolution: [1280, 720]
      focal_length: 35
      outputs:
        syclops_output_rgb:
          - id: "main_rgb"
            samples: 256
        syclops_output_pixel_annotation:
          - semantic_segmentation:
              id: "main_semantic"
          - depth:
              id: "main_depth"
```

Run the dataset generation with:

```
syclops -j job_config.yaml
```

The graphical assets included in the repository demonstrate the tool's capabilities.

Use Cases

Syclops has been applied in various real-world scenarios (see Figure 2 for an example). It has generated datasets for:

- **Semantic segmentation** of crop and weed plants in agricultural fields, achieving a mIoU of 80.7 on the Phenobench Benchmark (Weyler et al., 2024) compared to 85.97 with real images.
- **Volume estimation** of vegetables on a conveyor belt with physics simulation, showcasing its industrial automation potential.

These applications underline Syclops' versatility across outdoor and indoor settings, simulating complex object interactions.

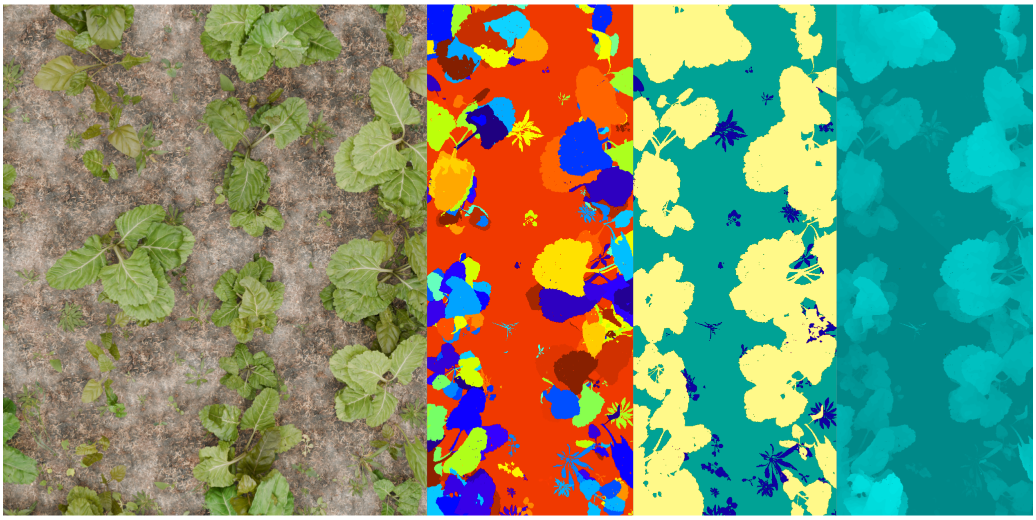

Figure 2: Data synthesized by Syclops for selective weeding in sugarbeets. Left to right: RGB image, instance segmentation, semantic segmentation, depth.

Limitations and Future Work

Syclops currently does not support the procedural generation of individual graphical assets. High-quality assets are essential for realistic data synthesis, and future work will address this limitation. Planned enhancements include:

- Developing tools for procedural asset generation.
- Expanding sensor simulation capabilities.
- Improving rendering realism and scene generation efficiency.

Conclusion

Syclops is a powerful tool for generating high-quality synthetic datasets in computer vision and robotics. Its modular, YAML-based architecture and focus on large-scale procedural generation make it especially suitable for off-highway applications such as agriculture. By providing extensive annotations and flexible configuration, Syclops supports accelerated research and development in challenging data collection scenarios.

Acknowledgements

We thank Henning Wübben, Florian Rahe, Thilo Steckel, and Stefan Stiene for their valuable feedback. Syclops was developed as part of the Agri-Gaia project, supported by the German Federal Ministry for Economic Affairs and Climate Action (grant number: 01MK21004A) and sponsored by the Ministry of Science and Culture of Lower Saxony and the VolkswagenStiftung.

References

Community, B. O. (2019). *Blender - a 3D modelling and rendering package*. Blender Foundation. http://www.blender.org

Denninger, M., Winkelbauer, D., Sundermeyer, M., Boerdijk, W., Knauer, M. W., Strobl, K. H., Humt, M., & Triebel, R. (2023). Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, *8*(82), 4901. https://doi.org/10.21105/joss.04901

Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D. J., Gnanapragasam, D., Golemo, F., Herrmann, C., & others. (2022). Kubric: A scalable dataset generator. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3749–3761. https://doi.org/10.1109/CVPR52688.2022.00373

Mumuni, A., Mumuni, F., & Gerrar, N. K. (2024). A survey of synthetic data augmentation methods in computer vision. *arXiv Preprint arXiv:2403.10075*. https://doi.org/10.1007/s11633-022-1411-7

Tabkhi, H. (2022). Real-world computer vision for real-world applications: Challenges and directions. *Proceedings of SAI Intelligent Systems Conference*, 727–750. https://doi.org/10.1007/978-3-031-16072-1_53

Weyler, J., Magistri, F., Marks, E., Chong, Y. L., Sodano, M., Roggiolani, G., Chebrolu, N., Stachniss, C., & Behley, J. (2024). PhenoBench: A large dataset and benchmarks for semantic image interpretation in the agricultural domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2024.3419548