University of Applied Sciences and Arts Northwestern Switzerland
School of Architecture, Civil Engineering and Geomatics

MASTER OF SCIENCE IN ENGINEERING

Evaluation of Game Engines for Labelled Image Generation for Object Detection and Pose Estimation

The generation of labeled image data for the training of machine learning models proves to be time-consuming and cost-intensive (Assadzadeh et al. 2022). Due to the almost realistic rendering of modern game engines (Pavelka and Landa, 2024), their use represents a possible alternative for the cost-effective and time-saving generation of labeled image data. In this master's thesis, the suitability of game engines is tested using Unreal Engine as an example. For this purpose, a general workflow is developed that automates the preparation of spatial data for use in a game engine, enables the simple creation of images and labels in the game engine and ensures subsequent further work by revising the data. The aim is to provide data that can be used directly for training machine learning models.

Initial Situation

Video recordings (Fig. 1) of a track excavator and rail crane from the Innosuisse Real Time Construction Monitoring project of the University of Applied Sciences and Arts Northwestern Switzerland FHNW and Rhomberg Sersa Rail Group serve as the basis for the master's thesis.

Fig. 1: Exemplary selected frames from the video recordings provided.

With the aim of supplementing or even replacing the existing real data with synthetic image data from the game engine, the workflow is designed to create images that are as similar as possible.

Data Preprocessing

To keep costs as low as possible, the workflow is based on open data that already exists and therefore does not need to be captured. More precisely, these are data on height, ground cover, buildings and track axis. The preprocessing is mainly based on the Python programming language, which is used independently or through the programming interface of the Blender modeling software. Thanks to a standardized project structure and the reusability of the code, the generation of new scenes is possible without great expenditure of time. The preprocessing of the data focuses primarily on adapting the data to the modeling area, necessary transformations and spatial modeling. As shown in Fig. 2, the first step is to convert and merge the different data into suitable data formats using Python. The data are transformed to the zero point and cropped to the predefined modeling area. The route for the track construction machines is also defined. Due to the much better modeling functions, Blender is used for the actual modeling. With the help of individual scripts, the prepared data can be processed fully automatically based on parameters. If desired, adjustments can be made directly afterwards. The data is then ready for use in Unreal Engine.

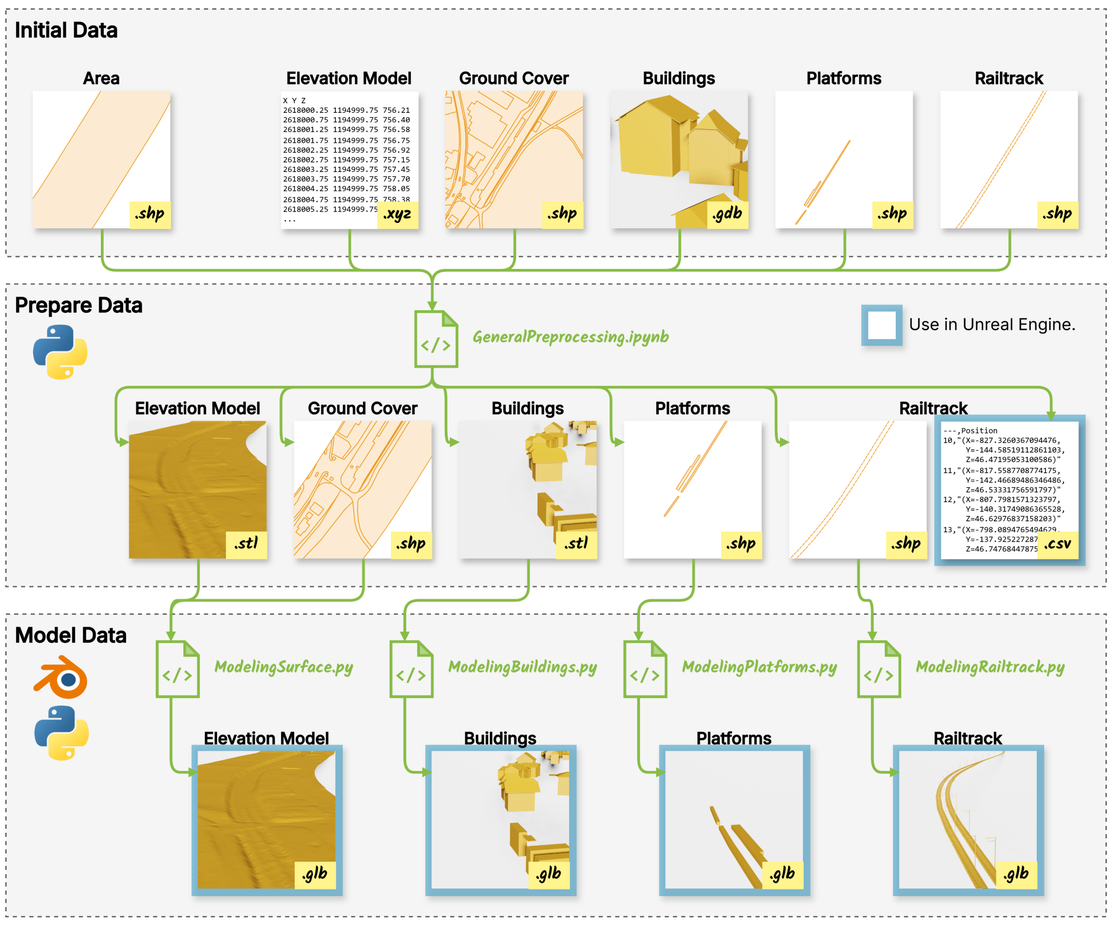

Fig. 2: Simplified representation of the developed pipeline for the preprocessing of spatial data.

Image and Label Generation

The Unreal Engine project is built in a way that data can be imported directly by specifying the project folder. Predefined assignments are used to assign surface materials and place vegetation automatically. The construction machine is placed on the movement path according to the selected project and can then be controlled using the defined movement options. The creation of images and labels can be triggered using the keyboard. In addition to the synthetic image, pose points and a segmentation mask are extracted. The pose is retrieved using hidden auxiliary points, the segmentation is created as presented by Grenier (2018) by switching to flat materials per object class and ignoring lighting. With the help of the implemented weather and time of day variations, the game engine can be used to create synthetic images as shown in Fig. 3.

Fig. 3: Selection of synthetic images generated with the created Unreal Engine project.

Label Postprocessing

After creation in Unreal Engine, the labels are not ready to be used directly for training machine learning models. Depending on the task, a segmentation mask is too detailed or the pose points are not available in the appropriate format. A revision using Python is much more user-friendly than a revision of the export function in Unreal Engine. Postprocessing can also be used to create a bounding box for object detection so that pose points, segmentation masks and bounding boxes can be automatically created for all synthetic images (Fig. 4).

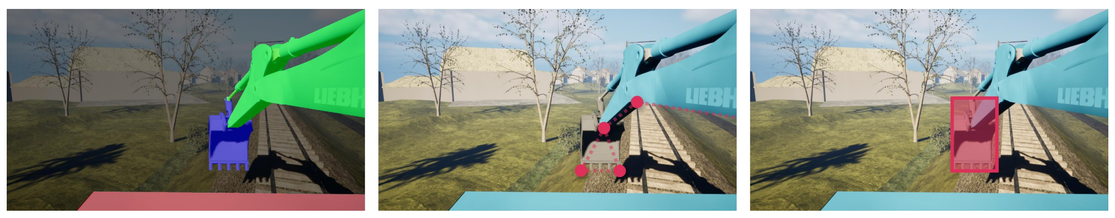

Fig. 4: Visualization of the three label types that can be created with the implemented solution.

Usability

Different training experiments with various amounts of real and/or synthetic data for object detection or image segmentation show that the presented synthetic data can provide better results in combination with the real data. However, the synthetic images do not come up to the performance of real image data. This does not mean that the synthetic data is useless. Especially in imbalanced data sets or in situations where it is not possible or very difficult to obtain additional real training data, they can offer a useful alternative. It is also conceivable to label only a small part of the real training data manually and to generate the remaining part artificially with the help of the game engine, thus simultaneously compensating for underrepresented configurations, which is not possible with the real data. The workload is lower due to the automatization of data preparation and label generation. Furthermore, the experiments show that small amounts of synthetic training data (less than 50% of the total training data) have hardly any negative impact on the performance.

Conclusion

The work shows a simple way to create labeled synthetic image data using a game engine and automatic data pre- and post-processing. Thus, it proves the usability of game engines for the generation of labeled synthetic image data. Even if the synthetic image data presented does not have the necessary depth of detail to completely replace real data, it can be used to support the real data and supplement underrepresented configurations.

Student: Luca Reber, luca.reber@students.fhnw.ch

Supervisor: Prof. Dr. Pia Bereuter, pia.bereuter@fhnw.ch
Tutor: Manuela Ammann, manuela.amman@fhnw.ch
Expert: Prof. Dr. Denis Jordan, denis.jordan@fhnw.ch

Literature:

Assadzadeh, Amin, Arashpour, Mehrdad, Brilakis, Ioannis, Ngo, Tuan, and Konstantinou, Eirini (2022). “Vision-based excavator pose estimation using synthetically generated datasets with domain randomization”. In: Automation in Construction 134. DOI: https://doi.org/10. 1016/j.autcon.2021.104089.

Grenier, Jeff (2018). Generating image segmentation datasets with Unreal Engine 4. Medium. URL: https://medium.com/@jeff_97181/generating-image-segmentation-datasets-withunreal-engine-4-2b5b9f75da34 (visited on 12/05/2024).

Pavelka, Karel and Landa, Martin (2024). “Using Virtual and Augmented Reality with GIS Data”. In: ISPRS International Journal of Geo Information 13.7. DOI: https://doi.org/10.3390/ijgi13070241.