

REPORT

D5.1 Tools and methods for synthetic data generation

WP5 AI-Video driven Building and Road Safety: Guarding the Future

Title of the work package: **AI-Driven Video**

Responsible partner: **IT4I@VSB**

Partners: CANEX, Optima Ideas, Asseco CE

Date: November 2025

Confidentiality: Public

www.innovaite.sk



Financované
Európskou úniou
NextGenerationEU

PLÁN [OBNOVY]



ÚRAD PODPREDESEDU VLÁDY
SLOVENSKEJ REPUBLIKY
PRE PLÁN OBNOVY
A ZNALOSTNÚ EKONOMIKU



V Ý S K U M N Á
A I N O V A Č N Á
A U T O R I T A

Content

1	Executive summary	6
2	Introduction	7
2.1.	Purpose of the Report.....	7
2.2.	Background and Motivation	7
2.3.	Scope and Limitations.....	7
3	Requirements for 3D Scene Generation.....	9
3.1.	Application Domains.....	9
3.2.	Objectives.....	9
3.3.	Requirements from AI Algorithms	10
3.4.	Key Quality Indicators	10
4	Specification of 3D Scene Generation	11
4.1.	Modeling Paradigm Overview	11
4.2.	Scene Generation Pipeline.....	13
4.3.	Parameterization and Control	14
4.4.	Scenario Definition.....	16
4.5.	Asset Selection and Management	16
4.6.	Rendering and Output Specification.....	17
5	AI Methods for Extending Variability.....	18
5.1.	Motivation.....	18
5.2.	Overview of AI-Based Augmentation	18
5.3.	State-of-the-Art Techniques	19
5.4.	Techniques for Scene Variability Enhancement	19
5.5.	Evaluation and Selection of Methods.....	20
5.6.	Production Pipeline.....	20
6	Current Development Status	22
6.1.	Current 3D Virtual Environment Progress	22
6.2.	Current AI Component Progress.....	27

6.3.	Identified Challenges and Gaps	28
7	Next Steps and Future Work.....	30
8	Conclusions	31
9	Annexes.....	32

List of figures

Figure 1: Example of manhole covers 3D assets.....	14
Figure 2: Example of building segments	15
Figure 3: Example of building variations created simply by adjusting their floor plan shape	16
Figure 4: Showcase of various parametric floor plan shapes	22
Figure 5: Randomly assembled panel building	23
Figure 6: Showcase of variations of a few simple elements.....	23
Figure 7: Type 1 - A suite of varied building typologies, prepared for subsequent implementation	24
Figure 8: Type 2 - A suite of varied building typologies, prepared for subsequent implementation	24
Figure 9: Type 3 - A suite of varied building typologies, prepared for subsequent implementation	25
Figure 10: Examples of road signs generated by the Geometry Nodes	25
Figure 11: Example of the variety of the created traffic signs.....	26
Figure 12: Test scene with postprocessing.....	26
Figure 13: Example of editing an image to incorporate a hazard into the scene, a missing manhole cover. Generated using Qwen-Image-edit	27
Figure 14: Generated 3D objects using Trellis	28

List of tables

Table 1: Overview of annexes..... 32

Revision history			
Version	Date	Comment	Author
0.1	16.10.2025	Provided template	IT4I@VSB
0.5	16.10.2025	Draft of the content	IT4I@VSB
0.9	7.11.2025	Writing the main content	IT4I@VSB, CANEX, Optima Ideas, Asseco CE
0.9.5	14.11.2025	Internal review	DFKI
1.0	28.11.2025	Final document	IT4I@VSB, CANEX, Optima Ideas, Asseco CE

1 Executive summary

This deliverable, D5.1 “Tools and methods for synthetic data generation”, is a report on the efforts in Task 5.1 Specification for the 3D scene generation, Task 5.2 AI methods for extending variability of generated datasets, and partially also in Task 5.3 Development of the 3D virtual environments, generation of synthetic datasets, and Task 5.4 Development of AI methods for extending dataset variability.

This is the first report to focus on the tools and methods for synthetic data generation. Reported approaches follow the needs and requirements defined by other tasks outlined in WP05 AI-Driven Video, specifically Task 5.5 Development of AI algorithms for increasing traffic safety from camera imagery and Task 5.6 Optimization of AI models and algorithms for multi-source video stream processing adapted to traffic and smart building domain.

In the following sections, we discuss the type of data and how it can be created to enhance AI algorithms for traffic safety and smart buildings.

2 Introduction

2.1. Purpose of the Report

The primary purpose of this delivery, D5.1 „Tools and methods for synthetic data generation” is to define the specifications and conceptual design of a comprehensive pipeline for 3D scene generation that meets the data requirements of AI algorithms in the domains of traffic safety and smart buildings. The report aims to establish a structured framework for creating synthetic visual data with sufficient variability, realism, and semantic richness to support training, testing, and validation of machine learning models. The proposed pipeline integrates procedural and parametric modeling approaches to enable efficient generation of diverse virtual environments while maintaining high levels of control over scene parameters and output quality. In addition, the report explores the application of modern AI-based methods, including generative models and data augmentation networks, to further expand the variability of generated scenes, thereby enhancing their realism and ensuring that they capture a broad spectrum of real-world phenomena and edge cases relevant to safety-critical applications.

2.2. Background and Motivation

Synthetic data generation has become a crucial enabler for advancing AI systems, particularly in domains where real-world data collection is constrained or costly. In fields such as traffic safety and smart building analytics, acquiring diverse and high-quality datasets require extensive manual annotation, complex sensor setups, and significant time investment. Moreover, privacy concerns often limit access to sensitive visual data containing people, vehicles, or interior spaces. Real-world datasets also tend to lack the variability needed for robust AI model training, such as rare events, edge cases, or specific environmental conditions (e.g., low light, adverse weather), which are difficult to capture in sufficient quantity. Synthetic data, generated within virtual 3D environments, offers a scalable and controllable alternative. Through procedural and parametric modeling techniques, it becomes possible to create diverse, photorealistic scenes with precise ground-truth annotations. Complementing this with AI-based scene enhancement techniques allows for further diversification and photorealistic refinement of rendered data. Together, these approaches bridge the gap between simulation and reality, enabling more robust, generalizable AI solutions.

2.3. Scope and Limitations

This report focuses on the specification and conceptual design aspects of synthetic 3D scene generation, with particular emphasis on procedural and parametric modeling strategies that enable high levels of variability and control. It defines the principles, parameters, and workflows required to produce visually realistic and semantically rich datasets suitable for training and evaluating AI models in the contexts of traffic safety and smart buildings. The scope also includes

an overview of AI-driven techniques for extending the variability and fidelity of generated datasets integrated into the broader scene generation pipeline. However, the report does not address implementation-level details, software development, or the practical training and validation of AI models. Instead, it provides the foundational specifications and design recommendations necessary to ensure that subsequent implementation phases can generate synthetic data of sufficient diversity and quality to meet advanced AI requirements.

3 Requirements for 3D Scene Generation

There are two main areas of interest regarding synthetic data generation for further AI training. It encompasses the areas of **Traffic Safety**, as well as **Security and Safety in Buildings**.

3.1. Application Domains

Traffic Safety – 3D scene generation in the traffic domain will focus on modeling and simulating high-impact safety-related situations such as traffic accidents, animals on the road, and missing manhole covers. These events represent rare yet critical situations with significant potential to endanger road users and infrastructure. As such events are rarely captured in real-world datasets, their synthetic reproduction in a virtual environment is an essential tool for developing and training AI models. Combining real and synthetic data will allow the creation of neural networks that are more robust and better prepared for real-world deployment within traffic information and safety systems. The resulting metadata can then be seamlessly integrated into Cooperative Intelligent Transport System (C-ITS) platforms, supporting real-time information exchange with drivers and enhancing the overall development of intelligent mobility.

Security and Safety in Buildings - In security and surveillance, AI systems trained on synthetic data can detect unauthorized access, suspicious activities, or perimeter breaches in real-time. For occupant safety, they enable the rapid identification of accidents, such as slips, falls, or medical emergencies, as well as the early detection of hazards like smoke or fire. Synthetic data also improves access control and compliance monitoring by helping AI recognize authorized personnel and ensure adherence to safety protocols. Such scenarios will be addressed using synthetic data to create a comprehensive dataset for training robust AI algorithms.

3.2. Objectives

Traffic Safety – the objective is to develop a framework that enables the synthetic generation of visual and metadata-rich data with a level of variability that cannot be achieved through real-world data collection. This approach builds upon existing methods of processing video recordings from mobile and static camera systems already deployed on Slovak highways [1], extending them with new data sources required for advanced artificial intelligence research. While certain traffic phenomena, such as stationary vehicles, wrong-way driving, or congestion, can be sufficiently captured in real-world conditions and used to train neural networks, there are specific situations that occur only rarely. These include missing or open manhole covers, animals on the road, and traffic accidents. Such events are of high relevance to traffic safety and represent critical inputs for early warning and prevention systems within intelligent transport infrastructures.

Security and Safety in Buildings – typically, access to real surveillance footage is limited by privacy and ethical constraints, which hinder the development of robust computer vision models for security applications. As such, simulated surveillance scenes are used and can provide diverse

environmental and behavioral contexts [2], that are otherwise inaccessible in real-world datasets. Synthetic datasets thus represent a scalable and privacy-preserving tool [3], that enhances model performance in anomaly detection and action recognition across real-world surveillance scenarios [4].

3.3. Requirements from AI Algorithms

Traffic Safety - the requirements for synthetic data arise from the needs of artificial intelligence algorithms that must be able to identify, classify, and localize objects and events with high precision in real time. The data must encompass a broad range of traffic scenarios and provide sufficient visual and physical information to support the effective training of neural networks. The key characteristics include variability, realism, physical consistency, balanced class representation, and adequate coverage of critical events. Important objects in the scene should be assigned coordinates, category labels, and timestamps, enabling the automatic generation of accurate annotations without manual intervention. The generated scenes must accommodate both common and exceptional situations, with a particular focus on anomalies that occur with very low frequency.

Security and Safety in Buildings - AI algorithms require synthetic datasets that are diverse, realistic, physically consistent, and well-balanced to ensure reliable performance in real-world conditions. Variability is essential to capture a wide range of environmental, lighting, and camera conditions, as well as differences in human appearance, movement, and occlusion. The data must exhibit a high degree of realism, both visually, through accurate textures, lighting, and camera artifacts, and behaviorally, with believable human actions and interactions. Physical consistency ensures that all movements and object interactions obey the laws of physics, maintaining plausible continuity across frames. Balanced representation of normal and rare events, such as accidents or security breaches, prevents bias toward the overwhelmingly common “normal” class. Finally, comprehensive coverage of critical phenomena is crucial so that the dataset includes all relevant security and safety events, from unauthorized access and vandalism to medical emergencies and fire detection.

3.4. Key Quality Indicators

To ensure the quality and validity of the synthetic data, measurable indicators are defined to allow comparison with real-world recordings. The main criteria include visual fidelity, geometric diversity, and semantic richness. These parameters assess the level of photorealism and visual accuracy of rendered scenes, the variability of structures and objects, and the level of annotation detail. The goal is to ensure that synthetic datasets provide equivalent or higher information value compared to real data and can be fully utilized for training and validation of neural networks.

4 Specification of 3D Scene Generation

4.1. Modeling Paradigm Overview

In the context of synthetically generating 3D scenes and rendering imagery from them for AI training purposes, it is crucial to adopt modeling paradigms that provide both **high variability** and **tight control** of scene content. Two complementary approaches are particularly relevant: **Parametric Modelling** and **Procedural Modelling**.

Parametric Modelling [5] refers to design workflows in which one defines geometry via a set of parameters or variables (for example, width, height, radius, and number of elements) and the relationships between them. By modifying these parameters, the model updates automatically to reflect the new configuration. This enables the efficient exploration of design variants and the capture of design intent through parameter-driven rules. Parametric Modelling is widely used in CAD and architecture for its flexibility in managing dependencies, constraints, and variational forms.

Procedural Modelling [6], by contrast, emphasizes the use of algorithmic rules, operations, or grammar to generate geometry automatically, often at a large scale. The key idea is that instead of manually positioning individual objects or vertices, one defines a generation procedure (script, node graph, or grammar) that can produce vast amounts of geometry (for example, city streets, landscapes, or building facades) from compact rulesets. Procedural workflows are especially powerful when one needs to generate large, complex, and highly variant scenes, because the rule system can easily yield many variations by sampling different parameters or random seeds.

While there is an overlap, since parametric modelling can form a subset of procedural modelling (i.e., changing parameters within a procedural ruleset), for dataset generation, the two paradigms combine: a procedural generation pipeline might include parametric controls to steer variation [7].

Tools and engines

Several tools and engines support parametric and procedural modelling workflows, each with varying degrees of scalability, automation, and visual fidelity:

- Commercial tools, such as ArcGIS CityEngine [8], specialize in generating procedural urban environments using rule-based grammar (also known as shape grammar) and parametric controls.

- General 3D modeling/animation engines, such as Autodesk Maya [9], Houdini [10], and Cinema 4D [11], provide node-based or scripting-based procedural/parametric capabilities.
- Real-time engines, such as Unreal Engine [12] or Unity [13], can host procedural scene generation modules (via Blueprints, scripting, and runtime variation) and render pipelines optimized for producing synthetic datasets.
- Simulators, such as CARLA [14], an open-source simulator designed for autonomous driving research, built on Unreal Engine, allow users to define roads, actors, and dynamic environmental conditions through Python APIs.

Blender and Geometry Nodes

For a flexible open-source workflow, Blender [15] presents a strong option. Blender's Geometry Nodes system is a node-based procedural/parametric modelling environment embedded in the software. It allows users to build node graphs that define the flow of operations (such as instancing, transformation, and attribute propagation) and expose parameters that can be varied to produce multiple scene variants.

Key characteristics in the context of our pipeline include:

- **Non-destructive node graph:** changes in upstream nodes propagate to downstream geometry, enabling rapid variant generation and parameter sweeps.
- **Parametric control:** nodes may expose sliders, attributes, or fields, allowing a designer or algorithm to systematically vary scene content (for example, the number of trees, building height, or road width).
- **Procedural logic:** branching, instancing, attribute-driven variation, random seeds, and rule-based generation enable large-scale scene generation from compact definitions [16].
- **Integration with export pipelines and rendering:** Blender supports export to industry-standard formats and can incorporate Python scripting to automate dataset generation workflows. It is also equipped with a powerful physically based Cycles renderer to obtain photorealistic image outputs.

In our specification, we propose leveraging Blender, with Geometry Nodes and Cycles renderer, as the **core modelling and rendering engine** because they strike a favorable balance between openness, flexibility, parameterization, and visual fidelity. The parametric controls exposed in the node graphs will enable us to generate a large number of scene variants with controlled variation. In contrast, the procedural logic embedded in the node graphs will automate the heavy lifting of

asset placement, variation, and instancing. At the same time, the Cycles renderer will provide high-fidelity rendered outputs.

4.2. Scene Generation Pipeline

In Blender, users should be able to generate one of the predefined scenarios in 3D scenes with ease, simply by selecting and clicking. Depending on the scenario, the user should be able to update:

1. scenario-specific parameters,
2. scene-specific parameters, and/or
3. 3D assets/static objects.

The parametrized parts (1 & 2) should use the user-friendly interface of “Tabs” in the 3D Viewport of Blender. Further updates to 3D assets (3) may be found in other parts of Blender, such as the Asset Browser, where you can drag and drop prepared 3D assets and materials.

One such example of a scenario-specific (1) geometry node for the “Manhole Covers” scenario should allow the user to update the placement of the manhole covers, varying their density, randomness, and the ratio of open to covered manholes. Examples of created manhole covers are shown in Figure 1. This manhole geometry node should be linked to the street object, so that when the street object is updated, the manhole placement is recalculated according to the chosen parameters.

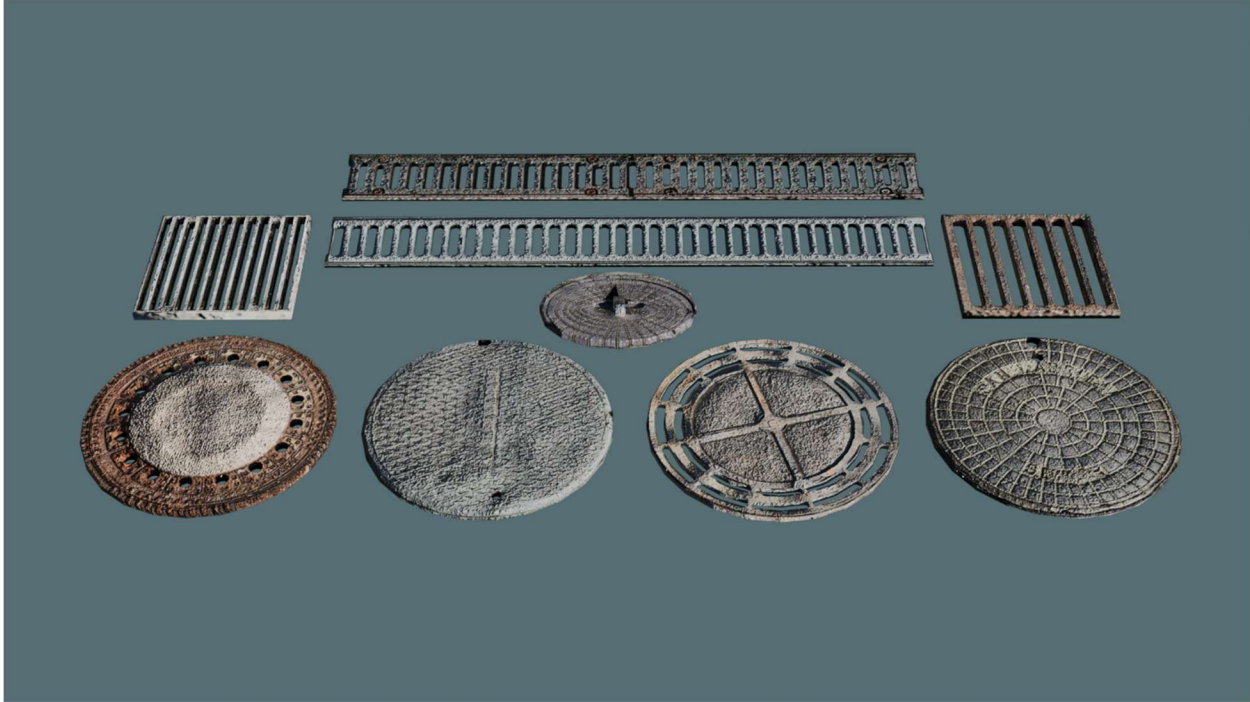


Figure 1: Example of manhole covers 3D assets

Scene-specific parameters (2) might refer to the parameters of geometry nodes in the scene. Let's consider the in-city scene, featuring geometry node-based buildings along the street. Again, this node should be linked to the street, so after updating the street, the buildings around are recalculated according to the chosen parameters. Parameters should include the density of buildings, their height range (i.e., the number of floors), width range (i.e., the number of windows horizontally), materials, and possibly a wider selection of accessories, such as windows, outer AC units, and balconies.

Another example of a scene-specific (2) parametrized geometry node is the greenery around the streets, with user-defined density, a subset of greenery (bushes and/or trees and/or grass).

3D assets (3) should be all located, clearly categorized, and tagged in the Asset Browser in Blender. The user can drag and drop each asset into the 3D Viewport.

4.3. Parameterization and Control

We describe parametrization and control concepts using an asset example, as well as other adjustable tools integrated into Blender.

We can take a building as an asset example that can be constructed from multiple segments, as shown in Figure 2, which, when randomly assembled, generates a wide variety of unique structures. Additional variation is achieved through procedural materials, which can be easily adjusted by changing material types, colors, patterns, or even adding signs of wear and damage.



Figure 2: Example of building segments

In Blender, there is also a natural way to add and work with realistic lighting in the scene. One of the available global lights is represented by Sky Texture. It is a physically based lighting system, allowing for easy adjustment of sun elevation, intensity, air pollution, and other parameters to achieve a wide range of atmospheric conditions.

Regarding the 3D assets, these will be distributed randomly, with the possibility of controlling the seed, but based on real-life layouts that reflect typical, mainly urban arrangements. A Geometry Nodes-based system is needed to enable the batch generation of architectural models and their variations in a single operation. Utilizing existing floor plans sourced from real map data, the system will produce these models in a single, efficient process. Furthermore, it can generate completely randomized city configurations. Key building parameters, including story count, facade composition, and floor plan morphology, are subject to procedural variation, allowing for nearly infinite unique outputs (see Figure 3). This methodology guarantees the extensive volume and diversity of environmental data required for robust machine learning.

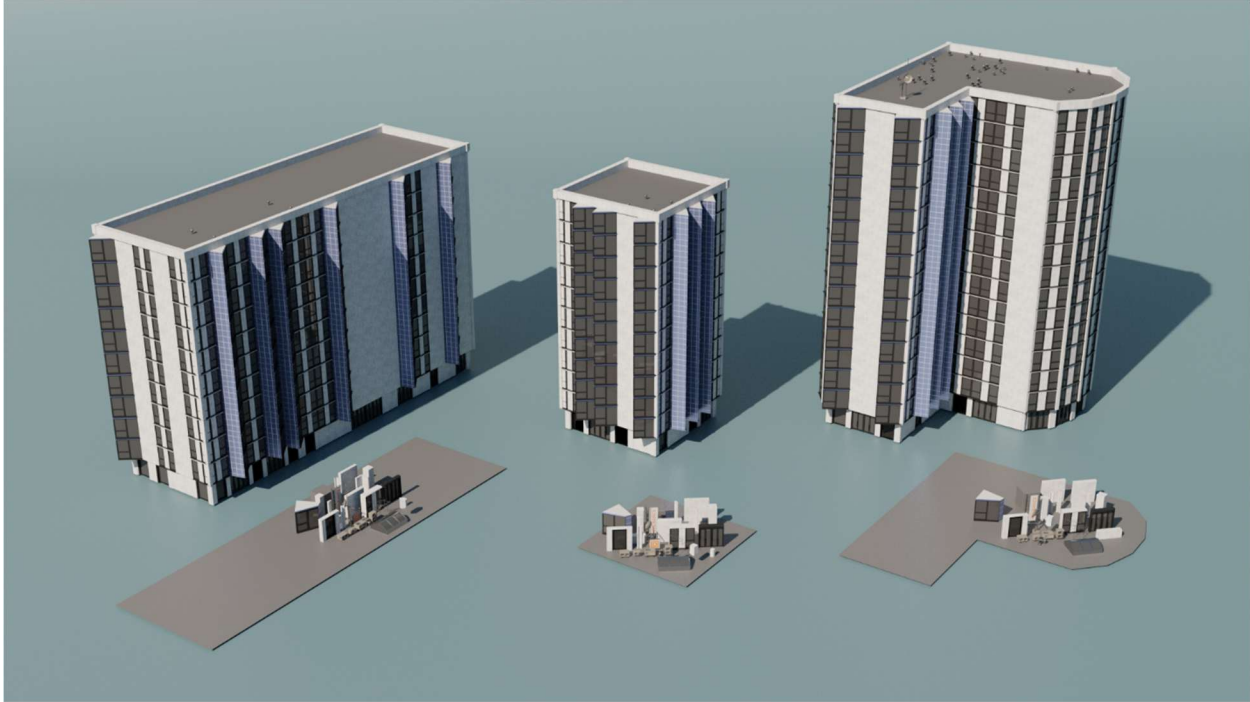


Figure 3: Example of building variations created simply by adjusting their floor plan shape

4.4. Scenario Definition

Scenario definition involves systematic design and generation of virtual scenes that replicate critical, safety-relevant events that are difficult or dangerous to capture in the real world. In the Traffic Safety domain, this includes creating realistic 3D simulations of rare yet high-impact situations such as vehicle collisions, sudden animal crossings, or infrastructure hazards like missing manhole covers. These synthetic scenarios enable AI models to learn how to recognize and respond to complex, unpredictable conditions that are seldom represented in real data. Similarly, in Building Security and Safety, scenario definition focuses on constructing detailed synthetic environments that simulate diverse incidents, such as unauthorized access, suspicious movements, accidents, medical emergencies, or fire hazards, under varying lighting, environmental, and occupancy conditions. By combining synthetic and real-world data, these defined scenarios provide the foundation for developing robust AI algorithms.

4.5. Asset Selection and Management

Criteria for selecting or creating 3D assets align with the requirements for developing AI algorithms. Demands are on a high level of realism, optimization – assets will be lightweight on geometry complexity, availability – assets can be easily purchased and used, and compatibility – assets will be compatible with Blender, i.e., will use native or standardized transport formats (BLEND, OBJ, FBX, glTF, etc.).

Current models used in the project have been primarily derived from existing models, such as traffic signs, roads, and some buildings, or created from scratch. All externally sourced models have been refined and adjusted to fit the project's requirements. All models aim to achieve realism while maintaining a limited polygon count and using procedural materials. The types of building assets and their details are selected based on their usability in the procedural assembly of complete buildings. The remaining 3D assets are chosen based on real-life objects typically found on the streets of Bratislava, ensuring they match well with Slovakian urban areas. Large scenes, such as those being developed, must be optimized to provide smooth viewport performance and quick rendering.

4.6. Rendering and Output Specification

The rendering workflow utilizes the EEVEE [17] renderer for fast, semi-realistic previews, allowing for quick changes and tweaks. For the final output, the Cycles renderer will be used to achieve physically accurate lighting and realistic materials. We will use a version of the Cycles renderer deployed on HPC clusters at IT4Innovations. This will allow for fast rendering at the highest possible quality.

The camera setup in Blender enables detailed settings that simulate the optical properties of a real camera, including accurate focal length and focus distance. These settings are crucial for replicating the realistic look of real dashcam footage.

To further enhance the rendered outputs, post-processing will be handled directly in Blender's Compositor Editor, where effects such as lens distortion, glare, depth maps for haze and fog, and color correction will be applied to refine the final image.

The final renders will be exported in one of the standard image formats supported by Blender. For a lossless compression format that can be directly digested by following the AI algorithm development, PNG might be the optimal format.

5 AI Methods for Extending Variability

5.1. Motivation

Procedural generation in our 3D environment (Blender) is a powerful starting point for creating diverse scene layouts. However, it may sometimes struggle to achieve the perceptual fidelity and vast combinatorial variety required for training robust AI algorithms for tasks such as traffic or building safety. These procedurally generated scenes, while structurally diverse, are often stylistically uniform, creating a "reality gap" that hinders the generalization of AI models to real-world camera imagery.

To address this, we leverage a multi-faceted AI strategy. Our primary focus is on **enhancing Blender renders** to inject photorealistic details—such as complex lighting, atmospheric effects, and material weathering—and to programmatically generate **vast environmental variations** like different weather and lighting conditions. This allows us to scale dataset creation far beyond manual methods.

Furthermore, our AI approach extends beyond render enhancement. We are exploring methods to **expedite asset creation** by generating 3D models from real-world imagery of Slovakian streets. We also see potential in using generative models for **direct scene manipulation**, such as using inpainting to modify existing camera footage or even generating entirely novel scenes, offering alternative pathways for rapid dataset expansion.

5.2. Overview of AI-Based Augmentation

Our strategy involves leveraging several classes of generative models to synthesize and enhance imagery. These models serve as a post-processing and enhancement layer for the output generated from the 3D engine.

- **Generative Adversarial Networks (GANs):** Primarily for style and domain translation. We can use GANs to learn the "style" of a real-world dataset (e.g., imagery from Bratislava) and apply it to our synthetic renders.
- **Diffusion Models:** State-of-the-art models for high-fidelity image synthesis and editing. These are exceptionally powerful for prompt-based img2img enhancement, where a base render is transformed into a photorealistic output, and for inpainting specific details or rare events into a scene.
- **Neural Radiance Fields (NeRFs) & 3D Gaussian Splatting:** While primarily for novel view synthesis, we are exploring these techniques (e.g., Trellis [18]) for generating high-fidelity, varied 3D assets from real-world photos (e.g., from Google Street View), which can then be processed by 3D designer and integrated back into our Blender scenes to increase asset diversity.

5.3. State-of-the-Art Techniques

The field of generative AI provides a powerful toolkit for generating synthetic data. The following technologies dominate the current state-of-the-art:

- **Diffusion Models:** As the current leading paradigm for image generation, these models (e.g., Stable Diffusion [19], DALL-E [20], Imagen [21]) excel at creating high-fidelity, diverse imagery from text or image inputs. Their core techniques, such as image-to-image translation and inpainting, are exceptionally well-suited for refining CG renders and introducing specific scene elements.
- **Control and Conditioning Layers:** A critical advancement for practical applications is the development of control layers. ControlNet [22] is a prominent example, enabling Diffusion Models to be conditioned on spatial inputs such as depth maps, Canny edges, or human poses. This enforces structural consistency, ensuring that generative enhancements do not compromise the underlying integrity of the source scene—a vital requirement for creating valid training data.
- **Generative Video Synthesis:** Extending generation to dynamic scenes is a key area of active research. The primary challenge is maintaining temporal consistency to avoid flickering. State-of-the-art approaches include dedicated video diffusion models (Text2Video-Zero [23], NVIDIA Cosmos [24], AnimateDiff [25]) that incorporate motion-aware mechanisms to produce coherent video sequences.
- **Generative Adversarial Networks (GANs):** Though less dominant for flexible image editing, GANs remain a foundational technology. CycleGAN [26] established a powerful method for unpaired image-to-image translation, a classic approach to sim-to-real domain adaptation. For asset creation, StyleGAN3 [27] is still a benchmark for generating seamless, high-resolution procedural textures with remarkable realism.
- **Neural 3D Representations:** For creating assets from real-world captures, NeRFs and the more recent 3D Gaussian Splatting [28] have revolutionized the field. They can generate photorealistic, navigable 3D scenes and objects from a set of photographs, offering a powerful method for digitizing real-world assets.

5.4. Techniques for Scene Variability Enhancement

By applying the state-of-the-art methods above, we can systematically enhance the variability of our generated scenes across several critical axes:

- **Weather & Illumination:** Transforming a single, neutrally lit base scene into a multitude of scenarios (rain, fog, snow, dawn, dusk, harsh sun) by applying stylistic image-to-image translation.

- **Material & Texture Variation:** Augmenting clean, procedural textures with AI-generated details like pavement cracks, road wear, graffiti, and rust to break stylistic uniformity.
- **Rare Event Simulation:** Using inpainting to programmatically insert critical but infrequent hazards, such as missing manhole covers, road debris, or stalled vehicles, directly addressing the need to train models on atypical situations.
- **Sensor Emulation:** Applying a final post-processing layer to simulate specific camera sensor noise, lens distortion, motion blur, and compression artifacts to more closely match the target hardware.

5.5. Evaluation and Selection of Methods

The criteria for selecting the optimal AI methods for our production pipeline include:

- **Fidelity & Realism:** The perceptual quality of the generated output, assessed both qualitatively and quantitatively with metrics like Fréchet Inception Distance (FID) and VBench-like evaluations.
- **Controllability & Ground-Truth Preservation:** The ability to guide the AI output precisely and, most importantly, to ensure enhancements do not invalidate the corresponding ground-truth labels (e.g., bounding boxes, segmentation maps) with consistency checks.
- **Temporal Consistency:** For video generation, the method's ability to produce stable, flicker-free sequences is paramount.
- **Performance & Scalability:** The models should be computationally efficient and suitable for integration into an automated, large-scale data generation pipeline on our HPC resources.

5.6. Production Pipeline

Our production pipeline is designed as a flexible, multi-stage workflow that deeply integrates our 3D environment with powerful AI modules, enabling the production of both static images and complete video sequences:

1. **Hybrid Scene Generation (Blender):** The process begins in Blender, where scenes are constructed using a combination of procedural generation and **AI-generated assets**. These assets, created from real-world captures, provide a baseline level of realism and location-specific detail. The environment is set up to export not only visual output but also a suite of ground-truth and control data (e.g., bounding boxes, depth maps, segmentation masks, etc.).
2. **"AI Render" Trigger (Blender Interface):** We are developing an integrated interface within Blender that enables users to trigger an "AI Render." This process packages the base render (either a single image or an image sequence) and its associated control data, sending it as a job to the appropriate AI module.

3. **HPC-Powered AI Enhancement Module:** The job is processed by our AI backend, which will run on our HPC cluster for scalability.
 - **For Images:** A Diffusion Model pipeline (e.g., a ComfyUI [29] graph) uses image-to-image and ControlNet to apply the desired photorealistic style or add specific elements.
 - **For Videos:** A dedicated video synthesis model (e.g., our planned NVIDIA Cosmos-based implementation) processes the entire image sequence, leveraging control maps to ensure temporal consistency and structural integrity throughout the clip.
4. **Final Composition & Augmentation (Optional):** The enhanced render (image or video) can be returned to Blender for final compositing or undergo a final AI-driven modification step, such as inpainting to add late-stage, dynamic elements like specific types of road debris or hazards that are more efficiently handled in the 2D domain.
5. **Dataset Compilation:** The final, enhanced media is packaged with its verified ground-truth data and stored, forming a complete entry in our training dataset.

6 Current Development Status

6.1. Current 3D Virtual Environment Progress

Here, we summarize the current state of development in 3D virtual environments. Our focus was mainly on parametric buildings, as they cover a significant portion of the scene in urban areas. The second focus was on creating optimized and fully parametric vertical traffic signs. Lastly, we have also proceeded with initial rendering tests of a populated scene with realistic global lighting and shadows.

- Existing scene prototypes or 3D assets for buildings

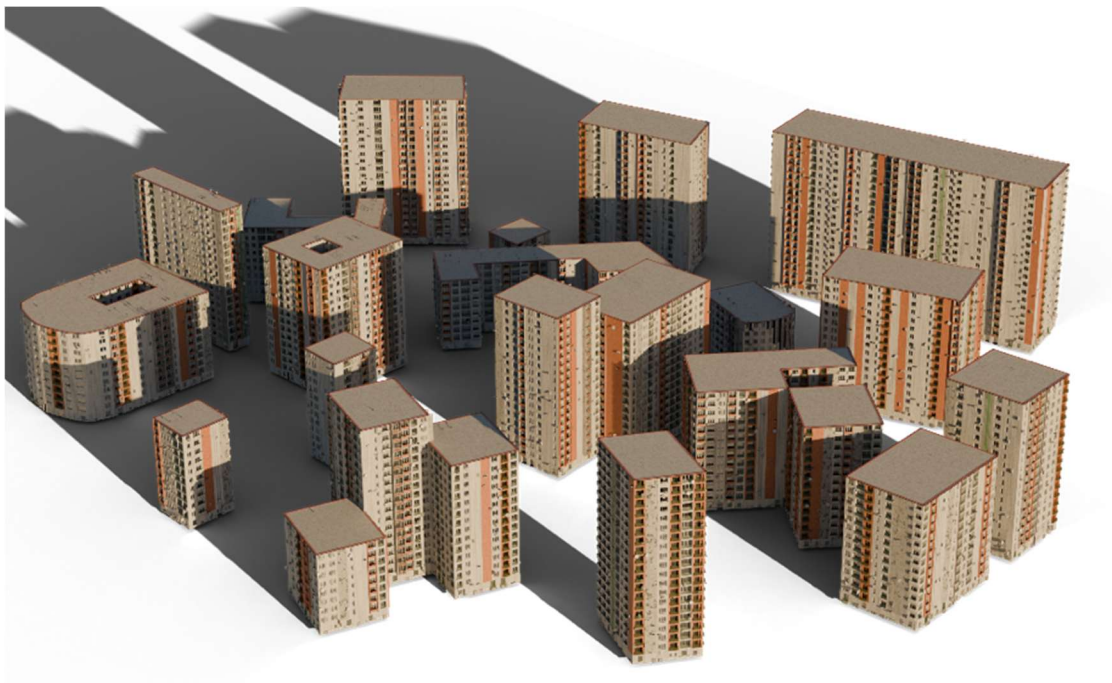


Figure 4: Showcase of various parametric floor plan shapes



Figure 5: Randomly assembled panel building



Figure 6: Showcase of variations of a few simple elements

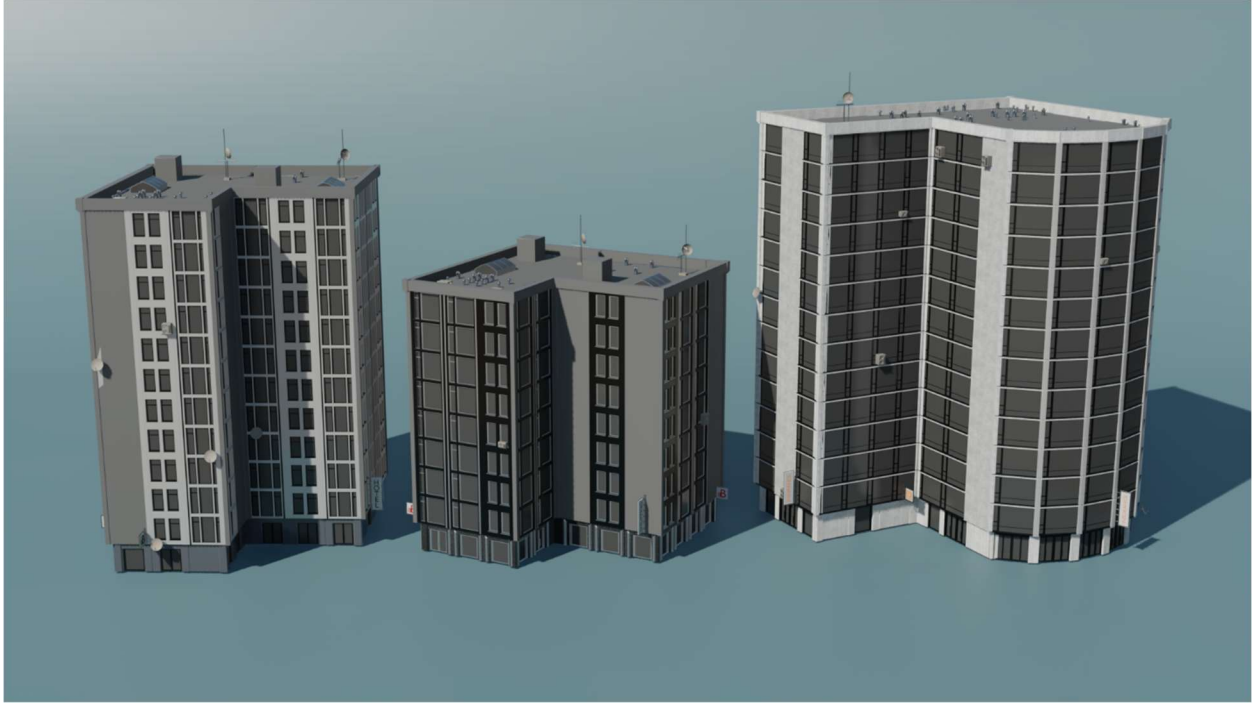


Figure 7: Type 1 - A suite of varied building typologies, prepared for subsequent implementation



Figure 8: Type 2 - A suite of varied building typologies, prepared for subsequent implementation



Figure 9: Type 3 - A suite of varied building typologies, prepared for subsequent implementation

- Functional Geometry Node setups for traffic signs – the design and color scheme of the signs comply with Slovak regulations as defined by the Ministry of Transport.



Figure 10: Examples of road signs generated by the Geometry Nodes

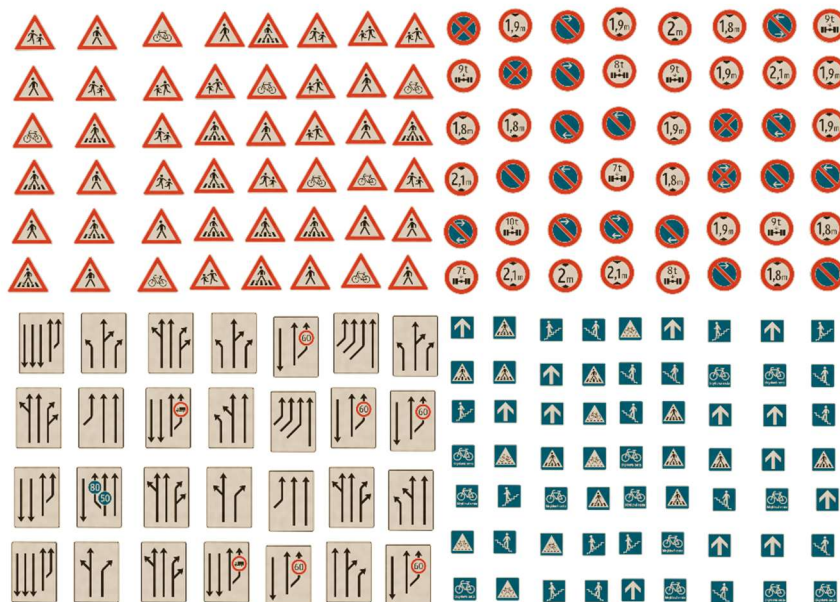


Figure 11: Example of the variety of the created traffic signs

- Rendering test – rendered using global lighting. After rendering, a simple post-processing step has been applied to simulate real-life camera and atmospheric effects, as well as their imperfections (glow, lens flare, light streaks, lens distortion, mist pass, etc.).



Figure 12: Test scene with postprocessing

6.2. Current AI Component Progress

Our initial phase has focused on validating key stages of this pipeline through several successful proofs-of-concept:

- **Diffusion Model Pipeline for Images:** We have established a working pipeline between Blender and ComfyUI, successfully using Stable Diffusion for image-to-image enhancement of our static renders. This validates the core photorealism workflow for still images.
- **Video Consistency Exploration:** We have conducted preliminary tests with the Cosmos framework on our HPC cluster. This work confirms our understanding of the temporal consistency challenge and provides successful proof of concept for our ControlNet-based video enhancement strategy.
- **Image alterations for Scene Augmentation:** We have explored prompt-based image alterations to introduce hazards, specifically using Qwen-Image-Edit [30] and Stable Diffusion to realistically render missing manhole covers in test scenes (see Figure 13), validating the final augmentation step of our proposed pipeline.



Figure 13: Example of editing an image to incorporate a hazard into the scene, a missing manhole cover. Generated using Qwen-Image-edit

- **AI-Driven Asset Creation:** We have experimented with Trellis, a framework based on 3D Gaussian Splatting, to generate 3D models from Google Street View imagery of Bratislava. This demonstrates a viable path for the first stage of our pipeline: populating Blender scenes with location-specific assets. Examples are illustrated in Figure 14.

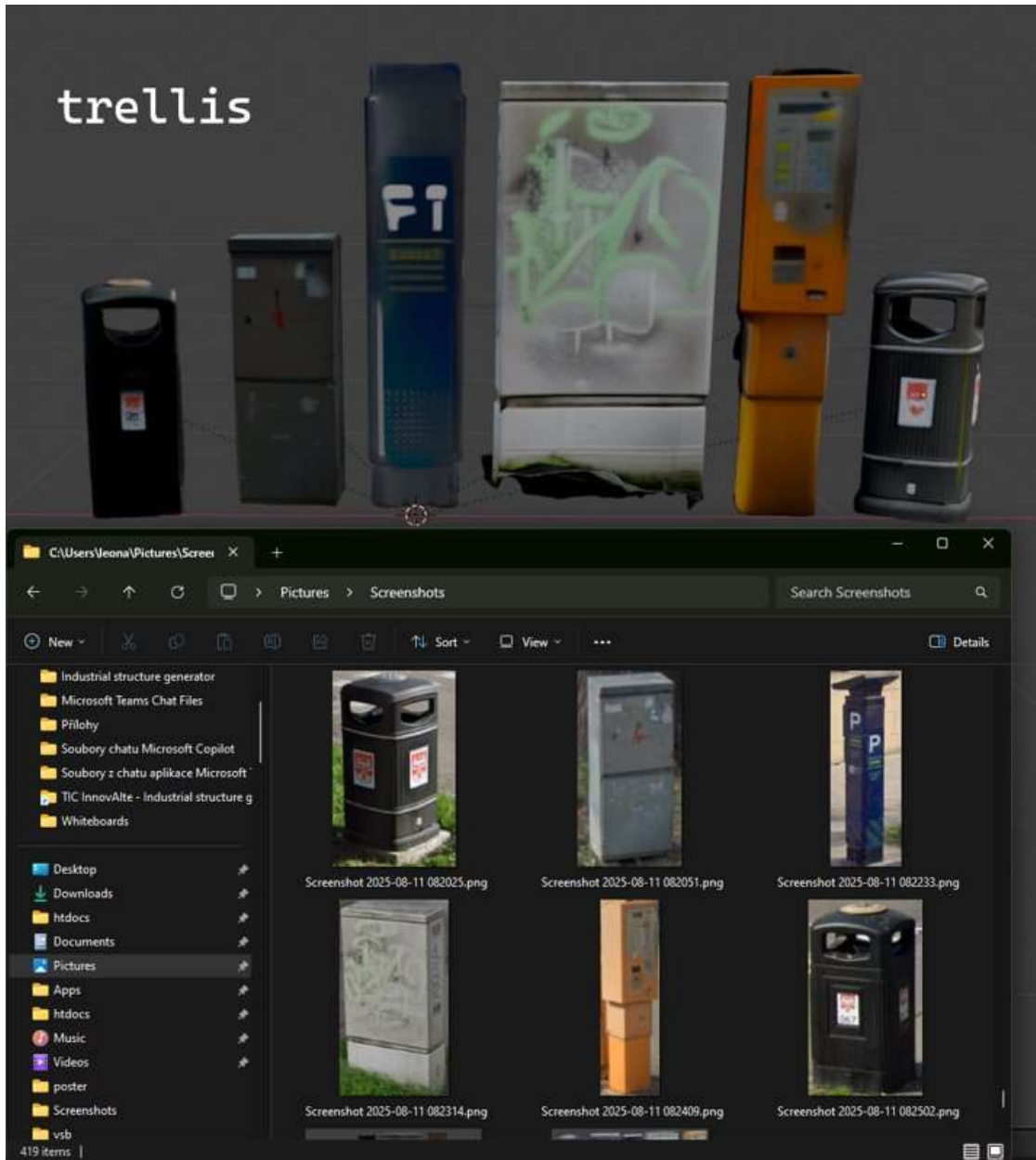


Figure 14: Generated 3D objects using Trellis

6.3. Identified Challenges and Gaps

Our initial work has highlighted several key challenges that will be the focus of our upcoming development:

- **Blender-based procedural synthetic data generation:** While Geometry Nodes and the Cycles renderer offer powerful foundations for parametric and photorealistic scene

creation, bridging the reality gap requires hybrid workflows that integrate physics-based simulation, real sensor modeling, and controlled domain randomization. Most of these challenges we aim to resolve by applying the proposed AI workflows.

- **Robust Temporal Consistency:** While NVIDIA Cosmos provided valuable insights, achieving perfectly stable and artifact-free video sequences at scale remains a primary challenge requiring further research into combining appropriate input combinations of guiding features from Blender (edges, segmentations, Depth maps).
- **Ground-Truth Integrity at Scale:** Ensuring that AI enhancements do not corrupt ground-truth data is a critical concern. We need to develop robust validation methods and refine our use of ControlNet to guarantee data integrity.
- **Engineering the Blender-HPC Bridge:** Developing the robust, user-friendly Blender add-on that can seamlessly package data and manage jobs on the HPC cluster is a significant software engineering task.
- **Fine-Grained Controllability:** Balancing the creative capabilities of generative models with the need for precise, predictable output requires extensive prompt engineering and refinement of control mechanisms.

7 Next Steps and Future Work

Planned improvements include procedural modeling (more parameters, new assets), integration of AI augmentation modules, benchmarking synthetic datasets against real-world data, and validation with AI models for traffic safety and smart buildings.

The focus will be on creating more 3D assets to cover specific environmental details related to the area of traffic and buildings. In the area of buildings with human avatars, cooperation with Comenius University will be leveraged, since they have long-term experience with modeling and simulating avatars. Following that, further integration and interconnection of procedural/parametric approaches in Blender will be implemented to quickly generate a large variety of modified versions of selected use cases. An important aspect in that will also be the use of Blender's physical engine, mainly to provide a plausible simulation of e.g., car accidents.

Our specific focus for the AI augmentation modules will be to engineer an automated Blender-to-HPC pipeline that operationalizes our successful proofs of concept. This pipeline will use controlled diffusion models to systematically inject photorealism and complex environmental variations into the base scenes for both images and video. This is the critical step required to bridge the reality gap and ensure our synthetic datasets provide maximum value for training and validating the final AI models.

A key upcoming phase will be the combination of real and synthetic data, utilizing our prepared platform tools once sufficient data has been gathered. This will enable comprehensive model training using Vision Transformers (ViT) and other advanced AI tools, following the latest trends and incorporating them into our ongoing research activities.

8 Conclusions

Within this report, we outline our strategy, which encompasses various tools and methods for generating synthetic data that can enhance the research and development of AI algorithms in the domains of traffic safety as well as security, and safety in smart buildings.

This report also tries to answer how data can contribute to the reach and fulfilment of one of the InnovAite project objectives, which is: **“4. Enhanced Traffic Safety: Reduce traffic accidents by at least 10% in selected urban areas using AI-based traffic safety systems within three years.”**

The next report, regarding tools and methods for synthetic data generation, is planned for M24 of the project, marked as D5.2: Frameworks for Synthetic Data Generation. It will describe the final solution for data creation and the generated data itself and will be the continuation of the effort marked here.

9 Annexes

List of annexes:

Name of Annex	Description of content of Annex	Confidentiality	Location	Link
Street 3D Assets for Blender	A collection of 3D models that can be used to populate virtual streets	Public	Zenodo	https://doi.org/10.5281/zenodo.17754828

Table 1: Overview of annexes

References

- [1] AI & Deep Learning Camera Analytics " caman. CAMAN. (2025, March 18). <https://caman.ai/>.
- [2] Delussu, R., Putzu, L., & Fumera, G. (2024). Synthetic Data for Video Surveillance Applications of Computer Vision: A Review. International Journal of Computer Vision.
- [3] Martin, A. (2024). Synthetic Data, Synthetic Media, and Surveillance. Surveillance & Society, 22(1).
- [4] Zhu, L., Wang, L., Raj, A., Gedeon, T., & Chen, C. (2024). Advancing Video Anomaly Detection: A Concise Review. NeurIPS 2024 Poster / OpenReview.
- [5] Verma, N. (2025, February 7). Comprehensive guide to parametric modeling. Media. <https://ikarus3d.com/media/3d-blog/comprehensive-guide-to-parametric-modeling>.
- [6] Wikimedia Foundation. (2025, April 17). Procedural modeling. Wikipedia. https://en.wikipedia.org/wiki/Procedural_modeling.
- [7] Parametric to procedural. GSAPP SMORGASBORD. (n.d.). <https://smorgasbord.cdp.arch.columbia.edu/modules/4-grasshopper-intro/49-procedural-rules>.
- [8] Procedural City Generator: 3D city maker: ArcGIS CityEngine. Procedural City Generator | 3D City Maker. (n.d.). <https://www.esri.com/en-us/arcgis/products/arcgis-cityengine/overview>.
- [9] Buy autodesk maya 2026 software: 3D animation software. Autodesk. (2025, November 25). <https://www.autodesk.com/products/maya/overview>.
- [10] Houdini: Procedural content creation tools for film/TV, Gamedev and more. SideFX. (n.d.). <https://www.sidefx.com/products/houdini/>.
- [11] Maxon. (n.d.). Cinema 4D: 3D Animation & Modeling Software. <https://www.maxon.net/en/cinema-4d>.
- [12] The most powerful real-time 3D creation tool - unreal engine. (n.d.-b). <https://www.unrealengine.com/en-US>.
- [13] Unity real-time development platform: 3D, 2D, VR & AR engine. Unity. (n.d.). <https://unity.com/>.
- [14] Team, C. (n.d.). Carla. CARLA Simulator. <https://carla.org/>.
- [15] Blender - the free and open source 3D creation software — Blender.org. (n.d.). <https://www.blender.org/>.

- [16] van Gumster, J., & Lampel, J. (2022). Procedural Modeling with Blender's Geometry Nodes: A workshop on taking advantage of the Geometry Nodes feature in Blender for procedural modeling. In ACM SIGGRAPH 2022 Labs (pp. 1-2).
- [17] Give back to Blender. Introduction - Blender 5.0 Manual. (2025, November 25). <https://docs.blender.org/manual/en/latest/render/eevee/introduction.html>
- [18] Xiang, Jianfeng, et al. "Structured 3d latents for scalable and versatile 3d generation." Proceedings of the Computer Vision and Pattern Recognition Conference. 2025.
- [19] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.
- [20] Ramesh, Aditya, et al. "Hierarchical text-conditional image generation with clip latents." arXiv preprint arXiv:2204.06125 1.2 (2022): 3.
- [21] Saharia, Chitwan, et al. "Photorealistic text-to-image diffusion models with deep language understanding." Advances in neural information processing systems 35 (2022): 36479-36494.
- [22] Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models." Proceedings of the IEEE/CVF international conference on computer vision. 2023.
- [23] Khachatryan, Levon, et al. "Text2video-zero: Text-to-image diffusion models are zero-shot video generators." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.
- [24] Agarwal, Niket, et al. "Cosmos world foundation model platform for physical ai." arXiv preprint arXiv:2501.03575 (2025).
- [25] Guo, Yuwei, et al. "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning." arXiv preprint arXiv:2307.04725 (2023).
- [26] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." Proceedings of the IEEE international conference on computer vision. 2017.
- [27] Alias-Free Generative Adversarial Networks (StyleGAN3) — Karras et al. (2021). arXiv:2106.12423; GitHub: NVlabs/stylegan3.
- [28] Kerbl, Bernhard, et al. "3D Gaussian splatting for real-time radiance field rendering." ACM Trans. Graph. 42.4 (2023): 139-1.
- [29] Generate video, images, 3D, audio with ai. ComfyUI. (n.d.). <https://www.comfy.org/>.
- [30] Wu, Chenfei, et al. "Qwen-image technical report." arXiv preprint arXiv:2508.02324 (2025).

Innovaite
SLOVAKIA



www.innovaite.sk



Financované
Európskou úniou
NextGenerationEU

PLÁN [OBNOVY]



ÚRAD PODPRESEDU VLÁDY
SLOVENSKEJ REPUBLIKY
PRE PLÁN OBNOVY
A ZNALOSTNÚ EKONOMIKU


VAIA
VÝSKUMNÁ
A INOVAČNÁ
AUTORITA