RESenv: A Realistic Earthquake Simulation Environment based on Unreal Engine

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Earthquake significantly impact human life and economic activities, necessitating efficient search and rescue operations to minimise loss and damage. As AI and robotics become widely applyed in these operations, the need for large, high-quality synthetic visual data has increased rapidly. However, current earthquake simulation methods, which primarily focus on the destruction of individual building frames, fall short of providing visually realistic simulations for multi-building urban scenarios. In response to the need, we propose a novel earthquake simulation environment based on the Chaos Physics System in Unreal Engine. This method aims to provide high-resolution, comprehensive visual and scenario simulation data for AI and robotic training in search and rescue operations. We employ actual seismic waveform data from an online database and process virtual building models for destructive simulations. Harnessing the capabilities of the game engine, we achieve realistic rendering, accurate physical collisions, and three-degree-of-freedom geological motion, surpassing traditional simulation frameworks in computational efficiency and user-friendliness. The proposed simulation environment offers a high level of detail and realism, serving as a rich, iterative data source for AI training in path planning and image recognition tasks related to earthquake rescue operations. We demonstrate the effectiveness of our approach through three specific AI-involved tasks, including similarity detection, path planning, and image segmentation.**

***Index Terms*—Earthquake Simulation, Virtual Environment, Artificial Intelligence, Robotic Training, Unreal Engine**

I. Introduction

Earthquakes, as recurrent natural disasters, profoundly impact human life and economic activities [1]. In an effort to enhance the efficiency of earthquake rescue operations and mitigate the loss of life and property, researchers are continuously exploring advanced search and rescue methods and technology [2]. Integrating artificial intelligence (AI) and robotics has been increasingly prominent in earthquake responses, encompassing tasks such as path planning, automatic obstacle avoidance, and image recognition [3]. Owing to the unpredictable and complex nature of earthquakes, developing methods for iterative training of these technologies across diverse scenarios is crucial for ensuring their robustness and reliability in related missions.

In recent years, virtual game engines have shown immense potential in disaster training and serious gaming applications, such as firestorm and flashflood simulations [4]. These engines considerably reduce operational complexity and shorten training cycles compared to traditional scientific simulator. Concurrently, realistic rendering techniques based on ray tracing have been proven effective in substituting real-world data in various visual recognition research areas [5]. However, there remains a dearth of research on constructing virtual environments for earthquake simulations using game engines.

We present RESenv, an environment for earthquake simulation utilising the Chaos physics engine within Unreal Engine 5 (UE5). By obtaining actual seismic waveform data from online databases and importing it into UE5, we simulate building destruction in the virtual environment. The primary objective of the approach is to provide high-resolution, comprehensive visual and scenario simulation data for search and rescue missions and robotics, serving as a synthetic data source for AI training in path planning and visual recognition applications.

The main contributions of this study are as follows:

1) We propose a method for realistic earthquake simulation based on UE5, which implements comprehensive destructive simulations of multi-building scenarios, using actual seismic data obtained online.
2) Three specific AI-involved tasks demonstrate the effectiveness of our approach, including similarity detection, path planning, and image segmentation, proving that our virtual seismic environment can effectively offer a high-quality dataset for AI training.
3) By leveraging the game engine's capabilities, we considerably reduce and streamline the complexity and work cycle of traditional methods. The intuitive interface eliminates domain-specific constraints, broadening user accessibility.

II. Related Work

This section reviews research related to earthquake simulation and AI training for rescue missions, which form the basis for our proposed UE-based earthquake simulation approach.

A. *Earthquake Simulation*

Earthquake simulation, a longstanding research focus in geophysics, geology, and engineering [6], has seen significant advancements due to recent progress in computer hardware. This has enabled more sophisticated modelling of earthquakes and consequent building damage using numerical simulation techniques [7], [8]. Stress simulations of individual buildings, initially aimed at analysing seismic stress-induced deformation and structural optimisation, have matured, and some researchers have employed finite element analysis (FEA) for assessing earthquake-induced building damage and exploring risk mitigation strategies [9], [10]. However, due to real-world buildings' structural complexity, material diversity, and computational constraints, most simulations only model primary load-bearing structures and facades, resulting in discrepancies between simulated and actual outcomes. Current earthquake platforms, constrained by the complexity of the physics engine limits and simulating only single or two degree-of-freedom (DOF) vibrations, fail to mimic the three DOF motions of actual earthquakes [11], [12].

Urban multi-building simulations, compared to individual building simulations, emerged much later. One of the most widely used frameworks is HAZUS, developed by the United States Federal Emergency Management Agency (FEMA) [13]. Based on standardised Geographic Information System (GIS) methodologies, HAZUS is employed for estimating the impact of earthquakes, post-earthquake fires, floods, and hurricanes, among other disasters. To overcome the limitations of HAZUS's single DOF model, Japanese researchers introduced the Integrated Earthquake Simulation (IES) framework, utilising multi-dimensional data fusion calculation methods [14]. Subsequently, Turkish researchers developed a regional building simulation method for Istanbul using MATLAB, based on the IES framework [15]. Similar to individual building simulations, multi-building simulations are also constrained by software limitations in terms of physical collisions and building structural complexity. Although a study by David et al. employed large-scale computing to simulate the motion of geological faults and measure building responses [16], the focus of these research predominantly lies in calculating the complexity of geological structures, with scant attention paid to the fidelity of building structures.

In our approach, we utilise the Chaos destruction system[1] and Nanite visualisation system[2] within the UE 5 game engine, achieving previously challenging fracture and fragmentation simulations for different materials and complex hybrid structures, accurate physical collisions, and three DOF geological motion. Our method demonstrates a significant improvement in computational efficiency and cost compared to conventional techniques, enabling real-time and accelerated calculations on consumer-grade computers. Benefiting from the user-friendliness of the game engine, we have established a pre-fabricated library of materials and structures readily available to users by creating pre-set programs, greatly simplifying operational difficulties compared to traditional simulation frameworks, and thus enabling researchers from various fields to use our approach with ease.

[1]https://docs.unrealengine.com/5.1/en-US/destruction-overview/

[2]https://docs.unrealengine.com/5.1/en-US/nanite-virtualized-geometry-in-unreal-engine/

B. *AI Training for Rescue Missions*

AI applications in search and rescue operations, as well as the robotics domain, are becoming increasingly widespread. Numerous researchers are dedicated to employing deep learning and reinforcement learning techniques for complex terrain path planning, image recognition, and other related tasks [17], [18]. For instance, the study by LinLin et al. utilised the SBMPC algorithm to investigate path planning problems for search and rescue robots [19]. Xuexi et al. explored indoor search and rescue using Simultaneous Localisation and Mapping (SLAM) and Light Detection and Ranging (LiDAR) methods [20]. Farzad's research introduced the application of deep reinforcement learning (DRL) methods in search and rescue robot tasks [21]. These studies underscore the potential of artificial intelligence in enhancing the effectiveness and efficiency of search and rescue missions.

The success of AI approaches largely depends on the availability of ample and high-quality data as training inputs, which can accurately represent the complexities and dynamics of environments affected by earthquakes [22]. However, in earthquake rescue scenarios, collecting and obtaining real-world data poses significant challenges. To address data limitations, researchers have developed various virtual environments, such as the RoboCup Rescue Simulation Environment [23], USARSim [24], and the BCB environment developed by Laurea University of Applied Sciences [25], for creating training data for deep learning and reinforcement learning algorithms in search and rescue operations. Nonetheless, these frameworks currently lack the level of texture rendering realism and detail richness required for training AI models that rely on image recognition and depth data inputs. This also results in substantial discrepancies between the volume and complexity of simulated scenarios and actual search and rescue missions.

Our proposed simulation environment fills this void by aiming to provide a highly realistic and detailed virtual earthquake damage environment using a ray-tracing system and authentic scanned textures. The environment allows for generating high-quality training data that can be directly used for AI algorithm visual recognition and depth data scanning. Weather phenomena, lighting conditions, and post-earthquake dust will be effectively simulated.

III. METHOD

RESenv executes the earthquake simulation in a three-stage process: **data preparation**, **data binding**, and **simulation**. Figure 1 describes the flowchart of the method.

A. *Data Preparing*

Virtual Building Processing Due to the flexibility in model importation within UE, virtual building models based on Polygon Mesh can be acquired from various sources. For instance, manual creation using modelling software like Blender, computation from GIS data in CityEngine software [26], or generation via AI methods [27]. However, to ensure that the building models can be effectively simulated, the models first need to be pre-processed before importing them

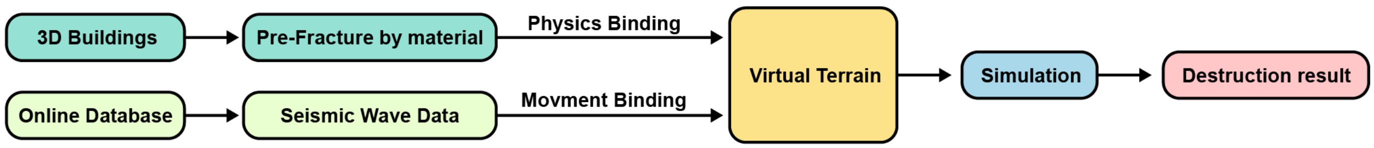

Fig. 1. Flowchart of RESenv for earthquake simulation. There are three steps: scenario preparation, data binding, and simulation. During the data preparation phase, 3D building models are imported into UE, which are then pre-fractured by material groups. Actual seismic wave data was acquired from the IRIS online database and imported via a graphical UI system. In the data binding phase, the 3D buildings are bound to the virtual terrain by PCA through an automated analysis program. The seismic wave data is converted into terrain displacements for binding. During the simulation phase, RESenv is run at a specified frame rate, 40 FPS for desktop computer simulation, 90 FPS for VR training, and 240 FPS for high frame rate sensor training. As the simulation begins, the seismic wave data displaces the terrain, which in turn causes the pre-fractured 3D buildings to be destroyed. RESenv is interactable throughout the simulation.

into our method through UE. Initially, the size units of the models need to be standardised. Typically, Polygon Mesh-based models do not have a unified scale unit like NURBS Surface models; the models need to be pre-scale to align to the cross-platform unified units. In RESenv, the default length unit in UE5 is adopted (centimetre) . Secondly, building models need to be segmented according to distinct materials, such as concrete, bricks, and wooden structures, to set up pre-fracture settings[3] separately after importing into UE.

In contrast to traditional earthquake simulation platforms that only simulate the physical collision between blocks and joints, our approach based on the Chaos Physics System in UE5 can achieve effects similar to real building destruction. Figure 2 presents examples of pre-fracture settings for three different wall materials. As seen in the figure, different segmentation strategies and levels can be applied according to distinct materials. The fractured geometry collection can be set with different break-force thresholds to represent the strength of the materials.

Seismic Wave Data Acquisition The seismic waveforms used by earthquake simulation can be divided into two categories. 1) Waveforms recorded from actual earthquakes that have already occurred. These waveforms can be obtained from publicly available datasets online. An earthquake event is often recorded by multiple seismometers located at different geographical locations; by cross-comparing and applying algorithms for noise reduction, the absolute motion of the Earth's surface can be authentically reproduced in simulation platforms. 2) Waveforms synthesised through algorithms [28], [29]. In seismic resistance testing of buildings, researchers have developed various methods to synthesise earthquake waveforms in order to assess the impact of different levels and types of earthquakes on building structures. This enables the simulation platform to carry out unlimited iterations of earthquake tests in any conditions. Our method primarily aims to simulate the damage sustained by urban buildings in actual earthquakes to provide realistic datasets for AI visual-based training; therefore, the method initially implements the simulation of global earthquake waveforms obtained from the IRIS online database[4]. The acquired seismic waveform data records three DOF of geological movement, named "BH1" (east-west direction), "BH2" (north-south direction), and "BHZ" (vertical direction). We have implemented a user-friendly user interface (UI) and Python-based automatic format conversion program in UE (Figure 3), enabling users to directly obtain seismic waveforms by clicking on the geographical location and event occurrence time on the global map without the need for complex data searches and imports. Once the user selects the required data, the waveforms are automatically converted into

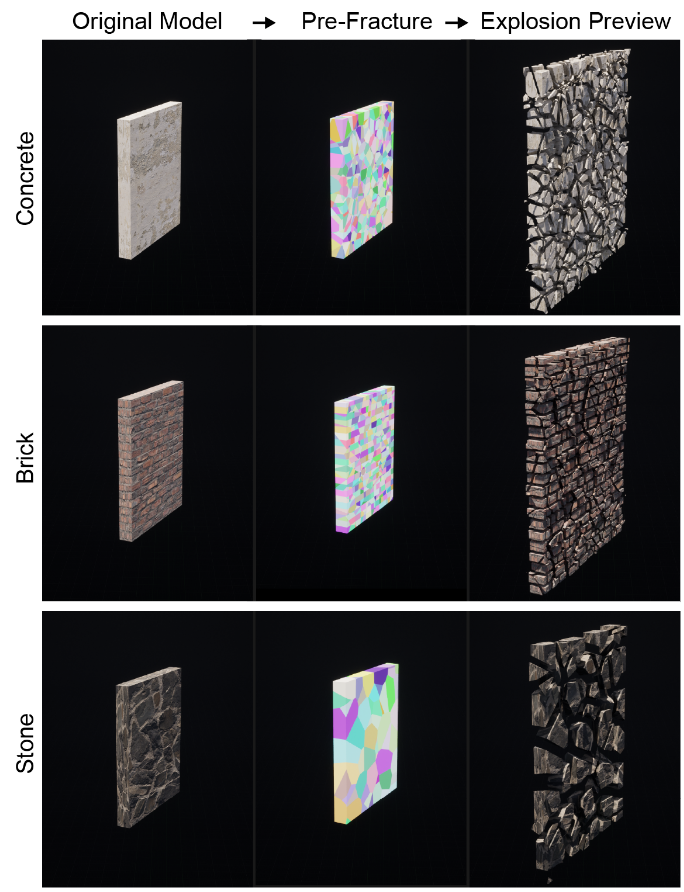

Fig. 2. Three examples of pre-fracturing of walls in different materials. Once the building has been imported into UE, the 3D model needs to be pre-fractured in order to be damaged accordingly in the earthquake simulation. The pre-fracture is set according to the material properties of the building. The three instances show the set-up process, segmentation methods, and explosion view for concrete, brick wall, and stone structures, respectively.

[3]https://docs.unrealengine.com/5.1/en-US/destruction-overview/
[4]http://ds.iris.edu/ds/nodes/dmc/data/

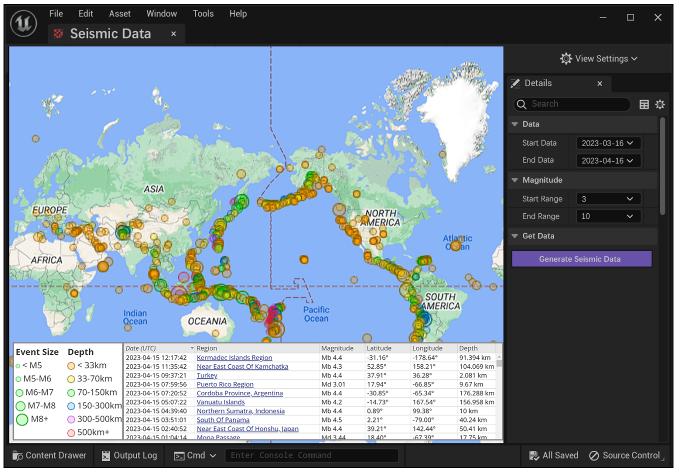

Fig. 3. The RESenv user interface for acquiring IRIS online seismic data. The user interface contains an interactive world map that can be clicked on to select the seismic data to be acquired. The column on the right side allows to define time ranges and earthquake levels as a filter.

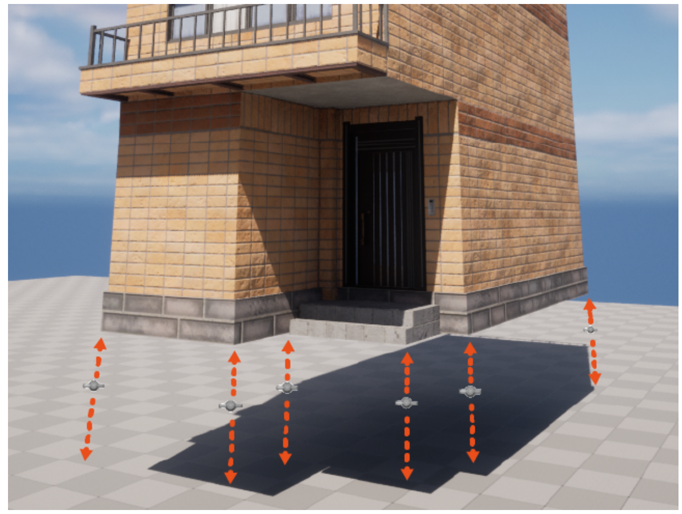

Fig. 4. Example of a 3D building automatically bound to a virtual terrain by the Physics Constraint Actor (PCA) placement program.

a "DataTable" file supported by UE.

B. Data Binding with Virtual Terrain

Similar to reality, our method simulates the anchor force exerted on buildings by ground movement. Consequently, the RESenv requires a virtual terrain in UE to support the buildings and bind the movements generated by the earthquake data.

The foundation of the building model is bound to the virtual terrain using the "Physics Constraint Actor" (PCA)[5] in the UE physics system. This binding establishes a set of physical anchor force constraints between the building and the terrain; the building's linear movement threshold and twisting threshold in the Cartesian coordinate system can be set by the building's materials and volumes. Similar to real-world events, during minor earthquakes, the building's foundation and walls will experience slight displacements without detaching from the ground. In a more significant earthquake, the foundation and walls will be fractured and collapse according to the building structure. The RESenv implemented a C++ program to automatically analyse the building foundation's shape to place PCA binding points. This program also estimates the building's volume and preliminarily sets the threshold values of various constraint forces at each binding point for quick configuration. Figure 4 demonstrates the distribution of binding points when a building is automatically bound to the ground in our approach.

To enable the virtual terrain to move in space like the earth's surface during an earthquake, we bind the previously acquired earthquake waveform data "BH1", "BH2", and "BHZ" to the "X", "Y", and "Z" axes of the terrain's movement using a C++ program automatically. The original waveform frequency of earthquakes in the IRIS database is 40 Hz (40 recorded samples per second). The RESenv implements three different frame rates in UE: 40 FPS, 90 FPS, and 240 FPS, corresponding to the desktop computer simulation, VR training, and high frame rate sensor data synthesis, respectively. The data for 90 and 240 FPS is pre-generated using the *wavelet interpolation* algorithm.

C. Simulation

Upon completion of the data binding, our method can be executed in UE in Simulate In Editor (SIE) mode[6]. It is worth noting that, unlike traditional simulation approaches, our method inherits features from Unreal Engine, allowing all virtual assets and fractured models to be interactive during runtime. Applications such as VR search and rescue training and robotic dynamic obstacle avoidance will transition from static scene training to dynamic training with time-varying properties. Figure 5 displays an example of a complete simulation process. In this instance, a concrete frame and a brick wall are bound to a flat virtual ground. The seismic data is sourced from the magnitude 7.4 earthquake in Oaxaca, Mexico, on June 23, 2020, recorded by the seismograph station coded as TEIG. The simulation lasts 360 seconds and runs at a frame rate of 40 FPS.

IV. Experiment

In order to verify the efficacy of our approach, two earthquake simulation experiments with three tasks were designed with the aim of providing synthetic visual data for AI training. 1) Two historical earthquake events and two laboratory experiments were reproduced and simulated with damage to buildings with four different materials. The similarity between real and simulated damaged buildings was assessed using a pre-trained Vision Transformer (ViT) model. 2) Using GIS

[5]https://docs.unrealengine.com/5.1/en-US/physics-constraints-in-unreal-engine/

[6]https://docs.unrealengine.com/5.1/en-US/in-editor-testing-play-and-simulate-in-unreal-engine/

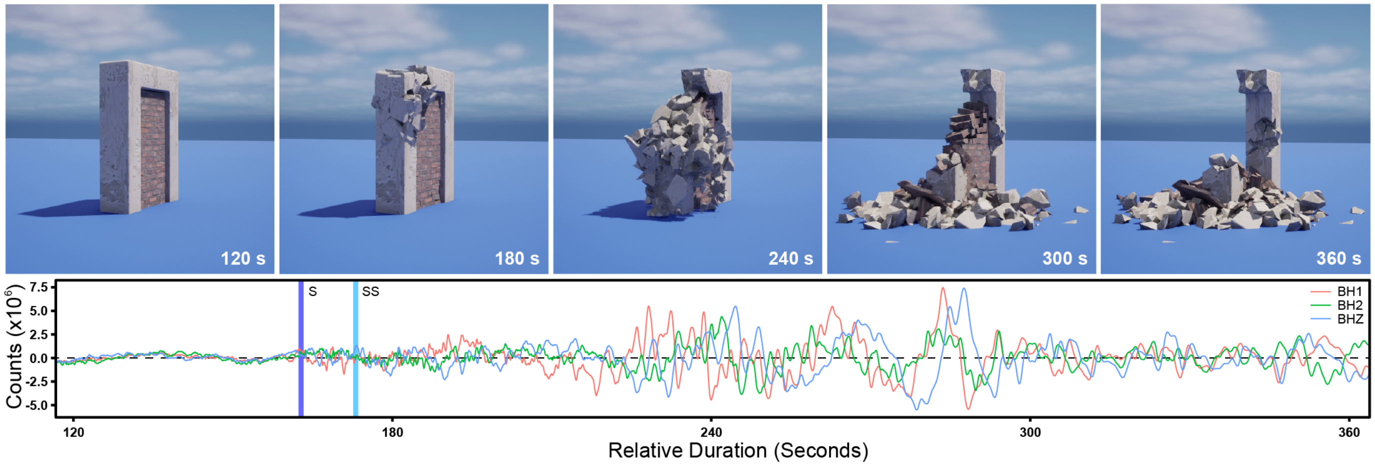

Fig. 5. An example showing the process of earthquake simulation by RESenv. In this example, a wall with a concrete and brick structure is pre-fractured and bound to a virtual terrain. The data taken from the 7.4 magnitude earthquake in Oaxaca, Mexico, on 23 June 2020, recorded by the seismic station code-named TEIG. The entire simulation lasted 360 seconds. Five frames were extracted to demonstrate the changes in distraction to the wall during the simulation.

data, we recreated a Japanese neighbourhood and then conducted a 20 random endpoints robot path planning test in the simulated post-earthquake area based on synthetic visual data. The completion rates of the robot's path and the success rates of visual recognition en route are counted.

A. *Realistic Simulation of Buildings*

Four real-world earthquake-induced building damage scenarios are selected and re-created the destruction in a simulated environment using RESenv to compare the accuracy of our approach. Figure 6 presents the original references and simulation results of the four scenarios. Each scenario simulates a building structure based on different materials. Two were simulated on a laboratory shake table [30], [31], and two were natural earthquakes [32], [33].

We obtained the building data and seismic wave data of the two laboratory-simulated scenarios via email correspondence. The seismic wave data of the two real-world scenarios were directly obtained from the IRIS database through the RESenv UI system, while the two buildings were reconstructed from multiple viewpoints using multiple online references. All four buildings were constructed in Blender, and surface textures were obtained from the Quixel Megascans material library[7]. The simulation was carried out in UE 5.1.1 on a Razor laptop with an RTX-3070 GPU, an AMD Ryzen 6900HX CPU, and 16GB of RAM as the minimal requirement. The initial UE scenario was set to the default configuration.

To ascertain the similarity between the simulated results and the actual structural damage, specifically in the context of robotic visual recognition tasks, we utilised a verified ViT deep learning model, specifically designed for feature similarity assessment. This model underwent pre-training with the ImageNet-21K dataset and its efficacy was substantiated in research conducted by Omini et al. [34] Images representing real-world damaged buildings and those derived from our simulations were independently input in the ViT model to compute their similarity. The final computational outcomes are presented in Table I.

The results demonstrate that the simulations for all four structures attained a considerable degree of resemblance to real-world scenarios. Our proposed earthquake simulation technique exhibits a robust capability to accurately replicate the damage patterns induced by actual seismic events in buildings. The observed discrepancies in the outcomes could be attributed to variations in the pre-fracture parameters of the 3D building materials, as compared to those of the reference structures. Consequently, these disparities give rise to differences in the morphology and movement trajectories of the fragmented masses within the simulation.

TABLE I
SIMILARITY OF THE FOUR SCENARIOS SIMULATED WITH RESENV.

Scenario	***Brick***	***Wood***	**Stone**	**Concrete**
Similarity	94.80%	90.07%	92.25%	89.34%

B. *Scenario Simulation and Robotic Training*

To evaluate the effectiveness of RESenv in conducting earthquake simulations within urban settings featuring clusters of buildings for robotic training, two distinct tasks are designed. Initially, the GIS data from a Japanese neighbourhood were obtained via OpenStreetMap and subsequently converted into a 3D scene utilising CityEngine. The scene was then imported into UE. All buildings within the scene were automatically anchored to the terrain using the program in RESenv, while the terrain was linked to earthquake data. A robot model, sourced from RoverRobotics®, was positioned in the scene and equipped with simulated RGB and depth camera sensors (Figure 7.a, d, e). The robot was assigned two tasks: 1) Utilising a DRL model based on SLAM, as proposed by Shuhuan et al. [35], the robot was instructed to randomly select

[7]https://quixel.com/megascans

Fig. 6. Earthquake simulation using RESenv for four actual scenarios. Two laboratory experiments (columns 1-2) and two actual buildings (columns 3-4) were chosen. Row 1: The original forms of four buildings before the earthquake events. Row 2: Destruction of buildings following the earthquake events. Row 3: The 3D models are recreated based on the actual buildings and are given textures and pre-fracture settings in RESenv. Row 4: Destruction results of four buildings after RESenv earthquake simulation.

20 coordinates as endpoints for path planning and obstacle avoidance testing within the simulated environment (Figure 7.b). The ratio of the completed length of each path to its total length is recorded. This test aimed to verify whether our simulation framework could provide effective earthquake scenarios for AI path-planning methods with demonstrated efficacy. 2) Concurrently with Task 1, the Segment-Anything Model (SAM)(model: ViT-H) [36] was selected as the state-of-the-art for generalised image segmentation model to collect data from RGB sensors for object segmentation (Figure 7.f). Image segmentation and its edge detection serve as the foundation for training AI models and path-planning tasks. For each frame from the camera, we compared the alignment of edges between the built-in UE segmentation (ground truth), SAM-processed ground truth, and SAM segmentation (Appendix A fig.8. Row 2-3). By using the Canny algorithm with edge dilation for pixel-level alignment deviation, the segmentation score was calculated as the proportion of overlapping edge pixels to the total edge pixels in the UE segmentation (Appendix A fig.8. Row 4-6). The final success rate for each path is derived from the average of all frames.

Upon completing the aforementioned tasks, the results (4 typical in Tabel II, full targets in Appendix B Tabel III) revealed that in Task 1, pertaining to path planning, 80% of path-planning trials achieved a 100% completion rate. When the input images have a resolution of 1550×1162 with a dilation kernel of 50 pixels, SAM achieved an overall 95% accuracy in detecting edges when compared to SAM-processed ground truth. These results prove our simulated post-earthquake scenario can furnish an effective image segmentation data source for visual recognition, thereby facilitating the training of various visual AI models.

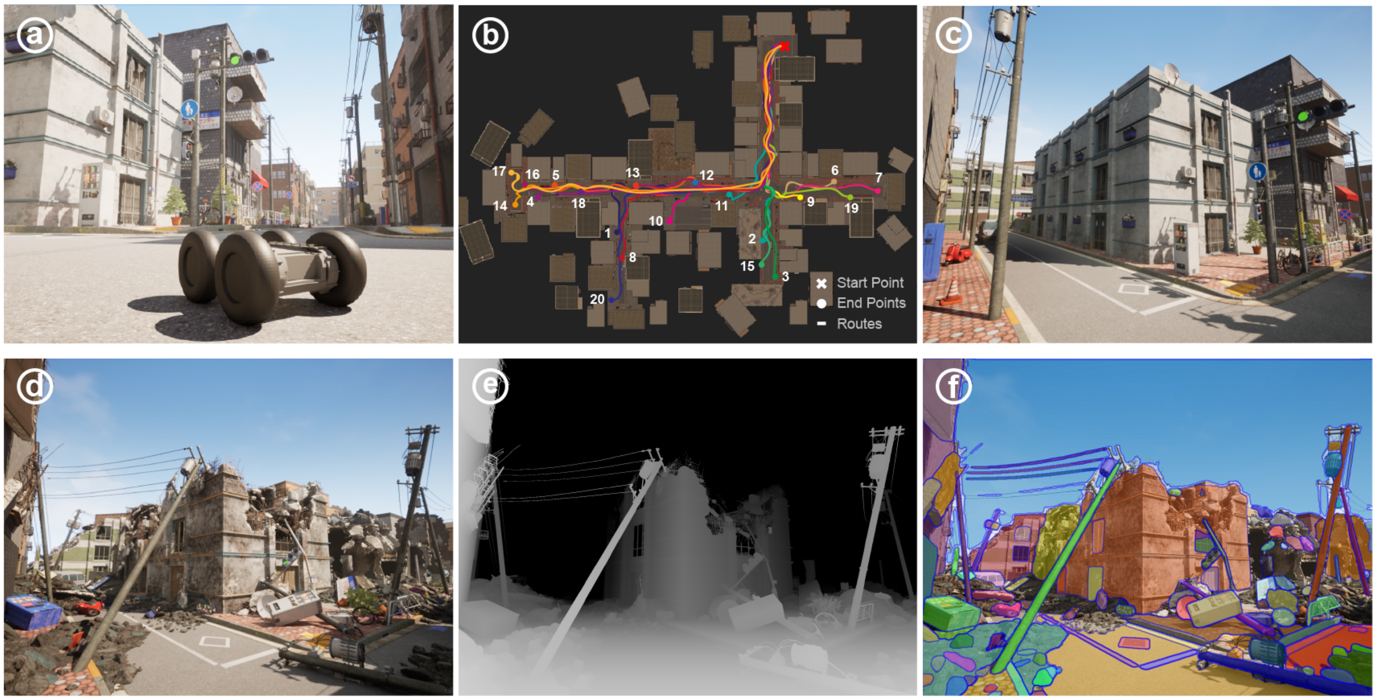

Fig. 7. Multi-building scenario earthquake simulation experiments in RESenv. Task 1 is to perform a robot path planning and obstacle avoidance test using the pre-trained DRL SLAM to verify the effectiveness of the earthquake simulation scenario for robot training. Task 2 is to use SAM to perform image segmentation detection on synthetic data from a RGB camera while the robot is travelling. Ultimately, the segmentation success rate will be counted. a: buildings generated in CityEngine using GIS data. A rover robot is placed in the scenario to perform path-planning tasks based on SLAM DRL. b: A record of 20 randomly selected endpoints for the path planning task. c: simulated RGB camera view of the original scenario in UE. d: RGB camera view of the scenario after an earthquake simulation. e: simulated depth camera view for SLAM DRL algorithm data input. f: RGB camera view with SAM image segmentation.

TABLE II
SUCCESS RATES OF DRL SLAM PATH PLANNING AND ACCURACY OF IMAGE SEGMENTATION EDGES FOR 4 TYPICAL OUT OF 20 TARGETS

Tgt. Pt.	**Path Plan.**	**Edges Acc.**			
		UE Segment		UE + SAM	
		Ker. 25	**Ker. 50**	**Ker. 25**	**Ker. 50**
1	Complete	81.2%	91.9%	89.9%	96.0%
8	Complete	76.9%	87.3%	96.4%	99.1%
14	96.55%	78.8%	90.1%	87.9%	95.2%
20	85.80%	74.2%	86.2%	90.4%	95.5%

Key Findings and Unforeseen Challenges The findings indicate that our proposed earthquake simulation approach effectively generates realistic urban destruction scenarios for robot training. The high completion rates in Task 1 suggest that our simulation environment is capable of providing challenging yet achievable path planning and obstacle avoidance test scenarios for AI algorithms.

Moreover, in Task 2, we observed lower completion rates for some paths. Upon further inspection, we attributed this not to the complexity of the urban scenario itself, but to the *glare from the virtual sun* interfering with the simulated RGB camera used by the robot (Appendix A fig.8. Path-14, Path-20), leading to visual recognition difficulties. This situation has been overlooked in studies using ideal laboratory conditions and similar simulation platforms. This highlights the importance of considering the complexity and multifactorial nature of real-world environments when designing and testing AI algorithms for disaster response and recovery, rather than focusing solely on object simulation.

V. DISCUSSION AND FUTURE WORK

Discussion Our study introduces an innovative earthquake simulation environment, designed to generate realistic urban scenarios for VR and robot training in the context of disaster response and recovery. Using computer vision techniques such as ViT, DRL SLAM, SAM, and our proposed earthquake simulation method, we have demonstrated the effectiveness of our approach by completing three distinct tasks: similarity, path-finding success rate, and segmentation edge accuracy. Our results show the environment is feasible for the deployment of downstream tasks.

Limitation There are several limitations to our study that should be addressed in future work. First, while our method simulates seismic damage to buildings, it does not account for the complexities involved in simulating the flexing of foundations and land. Second, the building material parameters used in our simulations are based on ideal parameters and iterative settings, which may not accurately reflect various building structures in different geographical regions. Finally, the performance of the AI algorithms in our simulated scenarios may be affected by various factors not considered in our study, such as lighting conditions and the presence of smoke or dust.

Future Works In the future, we plan to address the limitations by integrating more diverse building models and materials, potentially using perceptual similarity metrics, to enhance the realism of RESenv. We will also explore the adaptability of our approach to other disaster types, for instance, floods, hurricanes, and tsunamis, to expand its applicability in a broader range of emergencies. More strategically, in addition to simulations focused on disaster damage, realistic environmental multi-factor complexity, and stochasticity will be focused on, such as sudden weather changes, with the aim of creating a non-ideal extreme environment to increase the robustness of AI models trained in RESenv.

REFERENCES

[1] I.-L. Joseph, "The effect of natural disaster on economic growth: Evidence from a major earthquake in haiti," *World Development*, vol. 159, p. 106053, 2022.

[2] A. V. Nazarova and M. Zhai, "The application of multi-agent robotic systems for earthquake rescue," *Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms*, pp. 133–146, 2020.

[3] E. Magid, A. Pashkin, N. Simakov, B. Abbyasov, J. Suthakorn, M. Svinin, and F. Matsuno, "Artificial intelligence based framework for robotic search and rescue operations conducted jointly by international teams," in *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings" ER (ZR) 2019, Kursk, Russia, 17-20 April 2019*. Springer, 2019, pp. 15–26.

[4] E. M. Bourhim and A. Cherkaoui, "Selection of optimal game engine by using ahp approach for virtual reality fire safety training," in *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1*. Springer, 2020, pp. 955–966.

[5] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann *et al.*, "Kubric: A scalable dataset generator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3749–3761.

[6] S. S. Matin and B. Pradhan, "Challenges and limitations of earthquake-induced building damage mapping techniques using remote sensing images-a systematic review," *Geocarto International*, vol. 37, no. 21, pp. 6186–6212, 2022.

[7] Z. Xu, Y. Wu, X. Lu, and X. Jin, "Photo-realistic visualization of seismic dynamic responses of urban building clusters based on oblique aerial photography," *Advanced engineering informatics*, vol. 43, p. 101025, 2020.

[8] B. E. Shaw, B. Fry, A. Nicol, A. Howell, and M. Gerstenberger, "An earthquake simulator for new zealand," *Bulletin of the Seismological Society of America*, vol. 112, no. 2, pp. 763–778, 2022.

[9] Z. Xu, Z. Zhang, X. Lu, X. Zeng, and H. Guan, "Post-earthquake fire simulation considering overall seismic damage of sprinkler systems based on bim and fema p-58," *Automation in Construction*, vol. 90, pp. 9–22, 2018.

[10] F. McKenna, "Opensees: a framework for earthquake engineering simulation," *Computing in Science & Engineering*, vol. 13, no. 4, pp. 58–66, 2011.

[11] P. McEntee, "Simpson strong-tie research: The tye gilb lab," Jun 2022. [Online]. Available: https://seblog.strongtie.com/2012/05/simpson-strong-tie-research-the-tye-gilb-lab/

[12] T. Oleson, "Faking quakes at full scale: Giant shake tables simulate earthquakes to make buildings safer," Jun 2022. [Online]. Available: https://www.earthmagazine.org/article/faking-quakes-full-scale-giant-shake-tables-simulate-earthquakes-make-buildings-safer/

[13] P. J. Schneider and B. A. Schauer, "Hazus—its development and its future," *Natural Hazards Review*, vol. 7, no. 2, pp. 40–44, 2006.

[14] M. Hori, *Introduction to computational earthquake engineering*. World Scientific, 2011.

[15] A. Sahin, R. Sisman, A. Askan, and M. Hori, "Development of integrated earthquake simulation system for istanbul," *Earth, Planets and Space*, vol. 68, no. 1, pp. 1–21, 2016.

[16] D. McCallen, F. Petrone, M. Miah, A. Pitarka, A. Rodgers, and N. Abrahamson, "Eqsim—a multidisciplinary framework for fault-to-structure earthquake simulations on exascale computers, part ii: Regional simulations of building response," *Earthquake Spectra*, vol. 37, no. 2, pp. 736–761, 2021.

[17] M. M. Costa and M. F. Silva, "A survey on path planning algorithms for mobile robots," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2019, pp. 1–7.

[18] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *Ieee Access*, vol. 8, pp. 191 617–191 643, 2020.

[19] L.-L. Wang and L.-X. Pan, "Research on sbmpc algorithm for path planning of rescue and detection robot," *Discrete Dynamics in Nature and Society*, vol. 2020, pp. 1–11, 2020.

[20] X. Zhang, J. Lai, D. Xu, H. Li, and M. Fu, "2d lidar-based slam and path planning for indoor rescue using mobile robots," *Journal of Advanced Transportation*, vol. 2020, pp. 1–14, 2020.

[21] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[22] B. Bischke, P. Helber, S. Brugman, E. Basar, Z. Zhao, M. A. Larson, and K. Pogorelov, "The multimedia satellite task at mediaeval 2019." in *MediaEval*, 2019.

[23] C. Skinner and S. Ramchurn, "The robocup rescue simulation platform," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 2010, pp. 1647–1648.

[24] G. Polverari, D. Calisi, A. Farinelli, and D. Nardi, "Development of an autonomous rescue robot within the usarsim 3d virtual environment," in *RoboCup 2006: Robot Soccer World Cup X 10*. Springer, 2007, pp. 491–498.

[25] C. Grunwald, A. A. Khalil, B. Schaufelberger, E. M. Ricciardi, C. Pellecchia, E. De Iuliis, and W. Riedel, "Reliability of collapse simulation–comparing finite and applied element method at different levels," *Engineering Structures*, vol. 176, pp. 265–278, 2018.

[26] I. M. Badwi, H. M. Ellaithy, and H. E. Youssef, "3d-gis parametric modelling for virtual urban simulation using cityengine," *Annals of GIS*, vol. 28, no. 3, pp. 325–341, 2022.

[27] S. Chaillou, "Ai & architecture," 2019.

[28] B. Moseley, A. Markham, and T. Nissen-Meyer, "Fast approximate simulation of seismic waves with deep learning," *arXiv preprint arXiv:1807.06873*, 2018.

[29] B. Moseley, T. Nissen-Meyer, and A. Markham, "Deep learning for fast simulation of seismic waves in complex media," *Solid Earth*, vol. 11, no. 4, pp. 1527–1549, 2020.

[30] M. Giaretton, J. M. Ingham, and D. Dizhur, "Shaking table testing of two storey as-built and retrofitted scaled urm buildings," in *Structures*, vol. 41. Elsevier, 2022, pp. 260–269.

[31] "Seismic testing of seismic intensity 6." [Online]. Available: https://www.youtube.com/watch?v=ddr5KT1nX7M

[32] R. Michaelson, "Earthquake in Turkey and Syria kills thousands and devastates cities," *The Guardian*, Feb. 2023. [Online]. Available: https://www.theguardian.com/world/2023/feb/06/turkey-earthquake-2023-gaziantep

[33] V. J. Team, "Turkey earthquake: Before and after pictures show extent of destruction," *BBC News*, Feb. 2023. [Online]. Available: https://www.bbc.com/news/world-europe-64544998

[34] H. Omori and K. Hanyu, "Predict inter-photo visual similarity via pre-trained computer vision models," in *Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence*, 2022, pp. 1–9.

[35] S. Wen, Y. Zhao, X. Yuan, Z. Wang, D. Zhang, and L. Manfredi, "Path planning for active slam based on deep reinforcement learning under unknown environments," *Intelligent Service Robotics*, vol. 13, pp. 263–272, 2020.

[36] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

Appendix A.

Image segmentation comparison and score calculation for four frames of the path planning task.

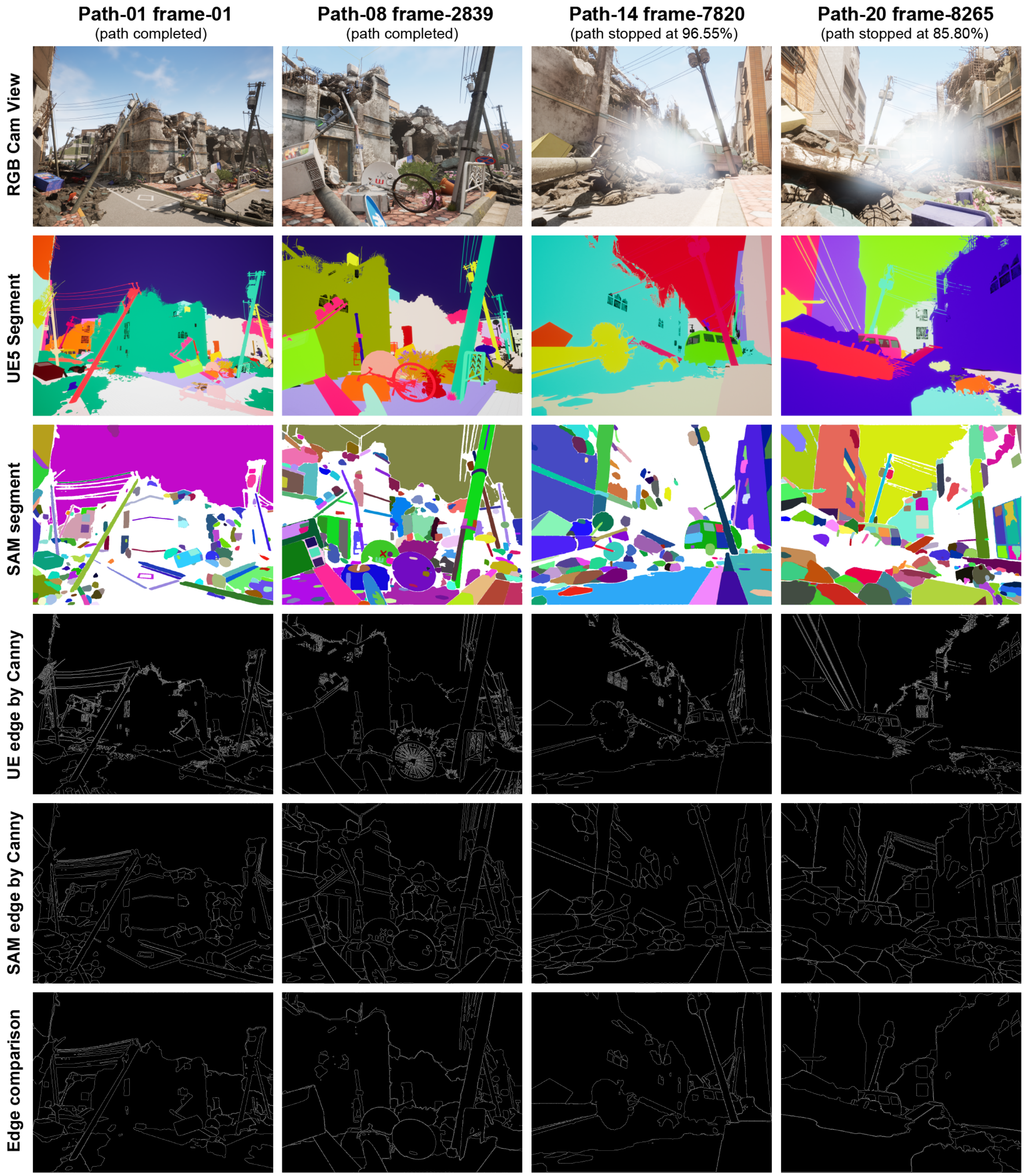

Fig. 8. Comparison of image segmentation and score calculation for 4 typical paths. **Row 1:** Synthetic RGB camera shot. **Row 2:** Object image segmentation using UE5 built-in camera post-processing material (ground truth). **Row 3:** Image segmentation of the RGB shot image using the SAM model. **Row 4:** UE segmentation edge map calculated using the Canny algorithm with edge dilation. **Row 5:** SAM segmentation edge map calculated using the Canny algorithm with edge dilation. **Row 6:** The segmentation score was calculated as the proportion of overlapping edge pixels to the total edge pixels in the UE segmentation.

Appendix B.

TABLE III
SUCCESS RATES OF DRL SLAM PATH PLANNING AND
ACCURACY OF IMAGE SEGMENTATION EDGES FOR 20 TARGETS

Tgt. Pt.	**Path Plan.**	**Edges Acc.**			
		UE Segment		UE + SAM	
		Ker. 25	**Ker. 50**	**Ker. 25**	**Ker. 50**
*1	Complete	81.2%	91.9%	89.9%	96.0%
2	Complete	76.3%	86.7%	88.6%	93.5%
3	Complete	79.2%	89.8%	87.9%	94.2%
4	Complete	75.6%	85.9%	86.8%	92.6%
5	Complete	82.5%	92.7%	90.8%	97.5%
6	Complete	71.8%	82.4%	85.0%	91.3%
7	85.4%	70.4%	81.1%	84.7%	90.8%
*8	Complete	76.9%	87.3%	96.4%	99.1%
9	Complete	79.9%	90.1%	89.5%	96.7%
10	Complete	81.0%	91.4%	90.6%	96.3%
11	Complete	80.2%	91.6%	90.4%	96.4%
12	Complete	81.6%	91.5%	89.9%	96.5%
13	Complete	80.7%	92.2%	90.2%	96.6%
*14	96.55%	78.8%	90.1%	87.9%	95.2%
15	Complete	81.3%	91.8%	90.0%	96.8%
16	Complete	80.9%	91.7%	90.5%	96.1%
17	92.21%	72.4%	85.0%	89.3%	94.9%
18	Complete	80.6%	91.4%	89.8%	96.7%
19	Complete	80.8%	91.9%	90.7%	96.3%
*20	85.80%	74.2%	86.2%	90.4%	95.5%