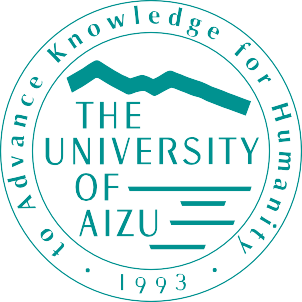
A thesis submitted in partial satisfaction of the requirements for the degree of Master of Computer Science and Engineering in the Graduate School of the University of Aizu

**Utilizing Unreal Engine 5 and Neural Radiance Fields for the Development of a High-Fidelity Robot Navigation Simulator:**

**An Approach for Accurate Environmental Replication and Enhanced Navigation Training**



### by

Yuminosuke Sato

#### March 2024

ⓒCopyright by Taro Aizu, March 2024

All Rights Reserved.

The thesis titled

#### Utilizing Unreal Engine 5 and Neural Radiance Fields for the Development of a High-Fidelity Robot Navigation Simulator:

#### An Approach for Accurate Environmental Replication and Enhanced Navigation Training

by

### Yuminosuke Sato

is reviewed and approved by:

*Hoge Hoge*

*Associate Professor*

**Chief referee**

*Professor*



*March 20*24

*Jiro Fukushima*

*Senior Associate Professor*

*Hoge Hoge*

Date:

Date:

Date:

# Contents

[Chapter 1 Introduction](#_bookmark6) 1

[Chapter 2 Body](#_bookmark7) 2

* 1. [Figure and Table](#_bookmark8) 2
  2. [Citation](#_bookmark15) 3
  3. [Abbreviations and Symbols](#_bookmark16) 4

[Chapter 3 Conclusion](#_bookmark19) 5

**List of Figures**

[Figure 2.1 Illustration of you writing the master thesis](#_bookmark10) 2

[Figure 2.2 Two types of PC](#_bookmark12) 3

# List of Tables

[Table 2.1 Example of table](#_bookmark14) 3

# List of Abbreviations

PC Personal Computer UoA University of Aizu WS Work Station

# List of Symbols

***a*** Vector

***A*** Matrix

R Set of real numbers

# Acknowledgment

**Abstract**

**Chapter 1**

**Introduction**

## How do you simulate robot navigation by transforming the real environment into a virtual environment? In the rapidly developing field of robotics, the simulation of real environments for robot navigation and operation is a crucial yet challenging endeavor. Creating 3D virtual environments that closely reflect reality is not only time-consuming and costly, but also has limitations in achieving a high level of accuracy. Moreover, the task is further complicated by the difficulty in replicating the dynamics of real environments and the insufficient integration of sensor data, which remains a persistent problem in the development of realistic simulation environments [1].

## Historically, expensive software and hardware, coupled with a need for specialized knowledge and high technical proficiency, have impeded the development of high-quality 3D models. The proper integration of sensor data from real-life situations has also placed constraints on the advancement of robot navigation and perception capabilities. Despite the significant risks and high costs associated with testing robots in real environments [1-3], it remains a crucial step in the development process. Nevertheless, simulations offer a promising alternative by providing a safe and cost-effective approach for robot testing and training, and even allowing for the simulation of robot operations that are unfeasible in real settings [1,4].

## Recognizing the potential of realistic simulation environments, this study proposes an innovative approach that utilizes Unreal Engine 5[5] to construct simulations based on accurate digital representations of real environments. The focus of this approach is to bridge existing gaps by providing high-fidelity, realistic digital simulations. Our method captures real environments from commodity mobile cameras and uses Luma AI[6] to derive NeRF[7] data. This data is utilized to form the foundation of virtual environments in Unreal Engine 5. This inventive approach streamlines robot navigation and operation simulations, while minimizing the related real-world risks. The result is a more secure and fortified framework for robot navigation and operation training and testing.

## This study represents a notable advancement in the field, introducing a method that carefully captures real environments with remarkable accuracy. Through the combination of Unreal Engine 5 and NeRF technology, we have developed a 3D simulation environment with unprecedented realism and functionality. This methodology enhances not only robot navigation and operation simulations but also opens up new opportunities in various domains, such as VR, AR, and the film industry, proving the multifaceted applications of this pioneering technology.

## 

**Chapter 2**

**Related work**

## 3D Modeling Techniques for Real-world Environments

This study introduces several innovative approaches to 3D modeling techniques for real-world environments. Firstly, a novel method using Geo-CNN technology is presented, enabling the modeling of local geometric structures of 3D point clouds. This technique captures the geometric relationships between points and their neighboring points, offering efficient operations that can be seamlessly integrated into many 3D point cloud analysis applications [8]. Additionally, an efficient Simultaneous Localization and Mapping (SLAM) approach utilizing data from 3D laser scanners is proposed. This method allows for fine-tuning alignments during online mapping based on local mapping and hierarchical optimization backend [9]. Furthermore, the 3D SLAM problem in complex outdoor and indoor environments based on millimeter-wave radio communication signals is explored, with a new method proposed using a deep learning-based mapping algorithm to construct 3D point cloud maps of the environment [10]. This study introduces a new geometric nonlinear probabilistic estimator algorithm, enabling accurate emulation of the true SLAM problem's nonlinear dynamics. This filter provides measurements of angular velocity, translational velocity, landmarks, and Inertial Measurement Units, ensuring satisfactory results.

## Visual Navigation through Realistic Simulators

Modeling real indoor scenes, including datasets like Matterport3D[11], Gibson[12], Replica[13], and Habitat-Matterport3D[14], involves extensive work. Unlike these datasets, primarily created using dedicated scan setups, scenes representing NeRF are trained using minimal video data from readily available mobile cameras.

**Visual Navigation in Simulation**: Several simulation suites have been proposed for embodied visual navigation tasks, combining 3D simulators like Habitat with various 3D scene datasets mentioned earlier[15, 16], iGibson[17], AI2/ROBO-THOR[18]. These simulators are used for learning visual navigation policies[19], solving object-based navigation[20], and incorporating language commands[21]. These approaches mainly consider dynamically simple platforms (wheeled robots) and operate purely in simulation.

## Unreal Engine in Robotics

Unreal Engine 5 has been employed in various robotic simulations. It proposes a generalized framework for highly realistic simulations of many robots and drones in natural environments. This framework, using Unreal Engine 4, generates optical and depth sensor outputs from any position and orientation within the environment[22]. By combining the Unreal Engine and Air-Sim system, scenes resembling real marine environments are created, constructing a visual dynamic simulation platform[23]. A comprehensive simulation system has been developed using \*\*Mission Oriented Operating Suite (MOOS) and Unreal Engine 4 (UE4)[24]. A highly realistic virtual reality environment is proposed for robotics simulation and synthetic data generation. Based on Unreal Engine 4, this environment aims for robotic agents to explore ultra-realistic indoor scenes and interact with objects in a visually realistic manner within that simulated world[25].

In robotics research, there are various platforms to choose from for physical simulation environments, each with its unique advantages and constraints.

**ROS & Gazebo**, released in 2007, adopts a socket-based client-server data transfer structure, which is a primary reason for its lack of reproducibility. Parallel execution isn't provided by default, requiring additional work. Moreover, package installation and usage are more complex compared to other environments, with high dependencies, making it less popular in recent RL research.

On the other hand, CoppeliaSim offers an integrated development environment and supports robot simulation. However, to ensure reproducibility, it requires the use of synchronous communication mode, which might lead to time losses. Additionally, optimizations are needed concerning communication speed and scalability.

Pybullet is cost-effective, lightweight, and user-friendly, but its graphic quality is low, making it unsuitable for visual-based sim-to-real research. MuJoCo provides a high-quality simulation engine and graphics, but long-term use requires a commercial license.

Recently released RaiSim & Surreal offer features necessary for modern RL research, allowing users to build parallel systems without much effort. Unity ML also has high-quality graphic rendering beneficial for visual-based sim-to-real research, but being a vast and complex system, it takes time to understand.

Lastly, NVIDIA IsaacGym offers simulations utilizing GPU acceleration, bypassing the bottleneck of data communication between CPU and GPU, resulting in a dramatic performance improvement. This innovative technology eliminates the need to invest time and money in building large-scale CPU cluster systems[26].

Unreal Engine possesses many superior features as a physical simulation environment compared to other platforms. Firstly, its physical simulation accuracy is exceptionally high, especially showcasing its prowess in the field of game development. This is complemented by a powerful Graphical User Interface (GUI) that provides real-time high-quality visual feedback. This GUI is renowned for delivering high-quality 3D graphics and real-time rendering.

Moreover, Unreal Engine supports a wide range of sensors and actuators, allowing researchers to simulate various sensors and actuators they require. This is highly beneficial for many applications and use cases in robotics research, especially in various robotics sub-domains.

Additionally, Unreal Engine boasts a vast community and extensive documentation, providing the support researchers need. This allows researchers to build parallel systems without much effort and utilize advanced simulation capabilities.

However, when compared to simulators specialized for specific robotics research needs, some advanced features might be lacking. Yet, with its powerful physics engine and high-quality graphic rendering capabilities, Unreal Engine has advantageous characteristics for visual-based sim-to-real research. Therefore, with its high physical simulation accuracy, extensive support for sensors and actuators, and a robust community and support, Unreal Engine stands out as a superior choice over other physical simulation environments.

## Neural Radiance Fields in robotics

Modeling real indoor scenes, including datasets like Matterport3D[11], Gibson[12], Replica[13], and Habitat-Matterport3D[14], involves extensive work. Unlike these datasets, primarily created using dedicated scan setups, scenes representing NeRF are trained using minimal video data from readily available mobile cameras.

**Visual Navigation in Simulation**: Several simulation suites have been proposed for embodied visual navigation tasks, combining 3D simulators like Habitat with various 3D scene datasets mentioned earlier[15, 16], iGibson[17], AI2/ROBO-THOR[18]. These simulators are used for learning visual navigation policies[19], solving object-based navigation[20], and incorporating language commands[21]. These approaches mainly consider dynamically simple platforms (wheeled robots) and operate purely in simulation.

1. 1. CITATION

\end{minipage}

\hfill

\begin{minipage}{0.48\hsize}

\centering

\includegraphics[scale=0.4]{./Figure/kaden\_laptop.png}

\subcaption{Laptop PC}

\label{fig:laptop}

\end{minipage}

\caption{Two types of PC}

\label{fig:pc}

\end{figure}





* + 1. Desktop PC

　　　　　　　(b)Laptop PC

Figure 2.2: Two types of PC

If you insert some tables, you should use table environment. For example, you put the following code (Listing [2.3)](#_bookmark13) and you can see Table [2.1.](#_bookmark14)

Listing 2.3: Example of Table 1

\begin{table}

\centering

\caption{Example of table}

\label{table:1}

\begin{tabular}{c|cc|c}

Name & Price & Number & Subtotal \\

\hline

Apple & 130 & 3 & 390 \\

Banana & 60 & 8 & 480 \\

Orange & 100 & 5 & 500 \\

\hline

\multicolumn{3}{r|}{Total amount} & 1370

\end{tabular}

\end{table}

Table 2.1: Example of table

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Price | Number | Subtotal |
| Apple | 130 | 3 | 390 |
| Banana | 60 | 8 | 480 |
| Orange | 100 | 5 | 500 |
| Total amount | | | 1370 |

## Citation

In this section, we give some examples of citation. If you cite something from books or papers, you must append references in your paper. Using .bib file is convenient to manage

CHAPTER 2. BODY

references because .bib file format of papers has already been made by web service such as google scholar and IEEE Xplore Digital Library.

For example, if you cite a book named “Citation example from a book”, you put TEX com- mand \cite{book1} and get the following [[1].](#_bookmark20) Similarly, you cite 78 page of a paper named “Citation example from a paper”, you put \cite[p. 78]{paper1} and get [[2,](#_bookmark21) p. 78].

## Abbreviations and Symbols

In this section, we introduce an convenient package named acro for abbreviations and symbols. If you show lists of either abbreviations, symbols or both, you should use this package. Listings [2.4](#_bookmark17) and [2.5](#_bookmark18) are example codes of an abbreviation and a symbol respectively. short field of abbreviations is the short form of a word, and long field is the long form. However, short field of symbols should be set a symbol, and long field should be written description

of the symbol.

Listing 2.4: Example of a definition for an abbreviation

\DeclareAcronym{pc}{ short = PC ,

long = Personal Computer , class = abbrev

}

Listing 2.5: Example of a definition for a symbol

\DeclareAcronym{A}{ short = $\bm{A}$ , long = Matrix , sort = A ,

class = nomencl

}

We can define other words by similar codes, and should make up the definitions into a

file (e.g., ./Chapter/Acronym.tex). If you put \ac{ws} and \acs{A}, you get Work Station [(WS)](#_bookmark0) and [***A***](#_bookmark3) respectively. Another example of output is the following. In this example, we called \ac{ws} again, so we obtain a different output than before. On the other hand,

\acs{A} outputs same result because \acs{} command always outputs the short format of

／

abbreviations and symbols.

[WS,](#_bookmark0) Personal Computer [(PC)](#_bookmark1) and University of Aizu [(UoA)](#_bookmark2) are abbreviations whereas [***A***](#_bookmark3), [***a***](#_bookmark4) and [R](#_bookmark5) are part of the symbols.

**Chapter 3**

**Conclusion**

**References**

1. H. Yuan and R. C. Veltkamp, "PreSim: A 3D Photo-Realistic Environment Simulator for Visual AI," in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2501-2508.
2. P. Salvini, D. Paez-Granados, and A. Billard, "Safety concerns emerging from robots navigating in crowded pedestrian areas," International Journal of Social Robotics, vol. 14, no. 2, pp. 441-462, 2022.
3. S. G. Tzafestas, Introduction to Mobile Robot Control. Elsevier, 2013.
4. D. Fernandez-Chaves, et al., "Robot@ VirtualHome, an ecosystem of virtual environments and tools for realistic indoor robotic simulation," Expert Systems with Applications, vol. 208, pp. 117970, 2022.
5. Unreal Engine 5. [Online]. Available: <https://www.unrealengine.com/>. [Accessed: September 15, 2023].
6. Luma Unreal Engine Plugin. [Online].Available:https://docs.lumalabs.ai/9DdnisfQaLN1sn.[Accessed: September 15, 2023]
7. Z. Wang, S. Wu, W. Xie, M. Chen and V. Adrian Prisacariu, "NeRF–: Neural radiance fields without known camera parameters", arXiv:2102.07064, 2021.
8. S. Lan, et al., "Modeling local geometric structure of 3D point clouds using geo-cnn," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
9. D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3D lidar-based online mapping," in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018.
10. Z. Mou and F. Gao, "Millimeter Wave Wireless Communication Assisted Three-Dimensional Simultaneous Localization and Mapping," arXiv preprint arXiv:2303.02617, Mar. 2023.
11. A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3d: Learning from rgb-d data in indoor environments,” arXiv preprint arXiv:1709.06158, 2017.
12. F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, “Gibson env: Real-world perception for embodied agents,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9068–9079, 2018.
13. J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, et al., “The replica dataset: A digital replica of indoor spaces,” arXiv preprint arXiv:1906.05797, 2019.
14. S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, et al., “Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai,” arXiv preprint arXiv:2109.08238, 2021.
15. M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al., “Habitat: A platform for embodied ai research,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9339–9347, 2019.
16. A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, et al., “Habitat 2.0: Training home assistants to rearrange their habitat,” Advances in Neural Information Processing Systems, vol. 34, pp. 251–266, 2021.
17. B. Shen, F. Xia, C. Li, R. Martín-Martín, L. Fan, G. Wang, C. Pérez-D’Arpino, S. Buch, S. Srivastava, L. Tchapmi, et al., “igibson 1.0: a simulation environment for interactive tasks in large realistic scenes,” in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7520–7527, IEEE, 2021.
18. E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, “Ai2-thor: An interactive 3d environment for visual ai,” arXiv preprint arXiv:1712.05474, 2017.
19. P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al., “On evaluation of embodied navigation agents,” arXiv preprint arXiv:1807.06757, 2018.
20. A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, “Simple but effective: Clip embeddings for embodied ai,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14829–14838, 2022.
21. P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3674–3683, 2018.
22. S. Casao, et al., "A Framework for Fast Prototyping of Photo-realistic Environments with Multiple Pedestrians," arXiv preprint arXiv:2304.07059, 2023.
23. M. Wang, et al., "LQR Control and Optimization for Trajectory Tracking of Biomimetic Robotic Fish Based on Unreal Engine," Biomimetics, vol. 8, no. 2, art. no. 236, 2023.
24. P. Martinez-Gonzalez, et al., "Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation," Virtual Reality, vol. 24, pp. 271-288, 2020.
25. J. Collins, S. Chand, A. Vanderkop and D. Howard, "A Review of Physics Simulators for Robotic Applications," in IEEE Access, vol. 9, pp. 51416-51431, 2021