Joint CSCE Construction Specialty & CRC Conference 2025
Conférence conjointe spécialisée en construction de la SCGC et CRC-2025

Montreal, Quebec
July 28-31, 2025 / *28-31 juillet 2025*

A Framework of Employing GPT and NeRF for Intelligent Robotic-Controlled Building Envelope Inspection

Ahmad Gholizadeh Lonbar[1], Yining Wen[1], Kaiwen Chen[1]* and Hongsheng He[2]

[1] Department of Civil, Construction and Environmental Engineering, University of Alabama, 245 7th Ave Suite 2004, Tuscaloosa, AL 35401, US
[2] Department of Computer Science, University of Alabama, 245 7th Ave Suite 2004, Tuscaloosa, AL 35401, US

ABSTRACT: This paper explores the integration of Neural Radiance Fields (NeRF) and Generative Pre-trained Transformers (GPT) to enhance robotic control systems. Traditional SLAM-based methods, while effective, often fail to build high-quality, cognitive maps of the physical world for complex control tasks, particularly in dynamic environments. To address these limitations, our research proposed a framework integrating the state-of-the-art perception and navigation techniques to enhance the human-robot collaboration in building inspection tasks, including 1) NeRF for creating detailed, photorealistic 3D representations of environments, and 2) GPT for translating natural language commands into precise robotic actions. This combination not only improves the resolution and fidelity of environmental models but also simplifies the human-robot interaction, making advanced robotic systems more accessible and intuitive for operators without specialized training. Through a series of pilot experiments, we demonstrate that this approach's feasibility and advances in enhancing the efficiency of robotic task execution in scenarios requiring detailed spatial awareness and dynamic task adaptation. The results indicate a promising direction for the future of autonomous systems, where high fidelity environmental understanding and ease of human interaction are paramount.

1. INTRODUCTION

Robot control is the key aspect of robotics and automation, focusing on the interaction between humans and machines to perform specific tasks. Robots are widely used in industrial automation, healthcare, and domestic applications, for example, drones or unmanned aerial vehicles (UAVs), are increasingly employed in surveillance, delivery, agriculture, emergency response, and construction. The control systems for both robots involve real-time decision-making, navigation, path planning, and actuation to achieve their objectives efficiently and reliably, often in dynamic or uncertain environments. Traditionally, these systems rely on predefined algorithms and sensor-based feedback loops to adapt to changes in their operational context.

Robotic control systems rely heavily on vision-based methods to enable autonomous navigation and interaction within complex environments. In traditional methods, cameras and sensors are used to generate

3D representations of the scene that are used to facilitate robust localization and motion planning algorithms (Wen et al.,2023). Traditional vision-based methods in robotic control have proven effective in many scenarios by offering robust localization and motion algorithms. However, they exhibit limitations when it comes to delivering comprehensive, high-resolution maps to both robots and human controllers. This shortcoming diminishes the operator's ability to fully comprehend and interact with the environment, a critical factor especially in dynamic settings such as construction sites. Even though these methods are capable of providing robust solutions for localization and basic motion planning, they often lack the detail necessary to implement more complex decision-making processes, especially when the situation is dynamic (H. Zhao et al., 2024a). Moreover, these systems typically rely on professional controllers to initiate and manage manipulation processes. This dependence on specialized skills restricts the user-friendliness of the systems and limits their adoption in sectors like construction, where operators often lack technical training.

One of the most promising emerging techniques is NeRF, which are capable of rendering continuous volumetric scenes of high fidelity, which is crucial for advanced robot control (Šlapak et al., 2024). NeRF addressed these challenges by synthesizing photorealistic images from sparse input data through a deep learning framework that models both volumetric density and scene appearance. The improvement does not only enhance the resolution and vividness of environmental representations, it also provides a cognitive map enabling advanced functionalities such as precise object manipulation as well as interactive tasks requiring detailed spatial awareness (Tagliabue & How, 2024).

Natural Language Processing (NLP) has emerged as a powerful tool to enhance robot control by enabling intuitive human-machine interaction through natural language commands. NLP methods facilitate the translation of spoken or written instructions into actionable commands for robots, reducing the complexity of traditional programming interfaces. With advances in deep learning and language models, NLP-driven control systems can interpret ambiguous or context-dependent language, adapt to different user intents, and provide robust responses. Applications include voice-controlled drones for search and rescue operations, robot assistants in healthcare that respond to spoken patient needs, and smart manufacturing systems where operators communicate instructions verbally to robotic arms. This integration of NLP not only improves usability but also broadens the accessibility of robotics technology for non-expert users.

The integration of NeRF and GPT introduces significant advancements over traditional vision-based methods. NeRF enhances spatial awareness and task precision by generating high-resolution, photorealistic 3D representations, enabling both human operators and robots to navigate and manipulate objects more effectively. Concurrently, GPT simplifies human-robot interaction by converting natural language commands into structured robotic instructions, making these systems more intuitive and accessible to non-expert users while reducing reliance on professional controllers. This research explores the combined application of NeRF and GPT to overcome key limitations in robotic control, particularly in construction environments. By leveraging NeRF for detailed environmental modeling and GPT for intuitive command processing, our approach enhances operational efficiency, safety, and adaptability. This integration not only addresses technical constraints but also broadens the usability of robotic systems across various industrial sectors, making advanced automation more practical and user-friendly.

2. LITERATURE REVIEW

Advancements in Neural Radiance Fields (NeRF) have reshaped 3D modeling, overcoming traditional limitations such as high costs, storage inefficiencies, and limited realism, particularly in robotics and industrial applications (Šlapak et al., 2024; H. Zhao et al., 2024a). By offering efficient and realistic modeling of complex scenes, NeRF has become essential for precision-critical applications, including video compression, depth estimation for obstacle avoidance, and robotic manipulation (Šlapak et al., 2024; H. Zhao et al., 2024a). In industrial settings, NeRF techniques substantially reduce data volumes without compromising quality, demonstrating adaptability across various resolutions (Šlapak et al., 2024; H. Zhao

et al., 2024a). Similarly, in the robotics field, dynamic NeRF models generate accurate depth maps that enhance collision avoidance systems, crucial for safe and autonomous navigation (H. Zhao et al., 2024a).

The innovation of distributed NeRF models in collaborative robotics represents a significant technological advancement. These models allow multiple robots to share learned model weights instead of raw sensory data, significantly reducing communication loads and enhancing system robustness, particularly in bandwidth-limited environments or situations with visibility constraints (H. Zhao et al., 2024a; Tagliabue & How, 2024). This collaborative approach ensures geometric consistency across sparse input views, improving perception accuracy in complex settings (Tagliabue & How, 2024). Moreover, integrating NeRF with Control Barrier Functions (CBFs) has led to the development of predictive safety filters that enhance robotic controllers by visualizing potential collision paths, thereby supporting more dynamic and less conservative navigation strategies (Adamkiewicz et al., 2022; Tagliabue & How, 2024). The application of NeRF in training robust visuomotor policies, such as through techniques like Tube-NeRF, uses data augmentation and robust predictive controls to handle uncertainties in navigation and manipulation. This method effectively bridges the simulation-to-reality gap, adapting to environmental variations and sensor inaccuracies, and enhancing the practical deployment of robots in real-world scenarios (Adamkiewicz et al., 2022). Studies have investigated various applications demonstrating how NeRF can aid robots in understanding and navigating complex 3D environments. Recent studies leverage NeRF to enhance robot perception in complex 3D environments. For instance, Zhao et al. (2024) propose a distributed NeRF framework for collaborative multi-robot perception, where robots share compact NeRF weights to efficiently reconstruct shared scenes, achieving robust performance in sparse-view settings with minimal bandwidth (H. Zhao et al., 2024b). Similarly, Kerr et al. (2023) introduce LERF, embedding CLIP-based language fields into NeRFs to enable real-time 3D language queries, enhancing semantic understanding for robotic navigation and interaction (Kerr et al., 2023). Neff et al. (2021) introduce DONeRF, a method to accelerate NeRF for real-time rendering in applications like games and virtual reality. Addressing NeRF's high computational cost, which requires dozens of petaFLOPS due to excessive network evaluations, DONeRF employs a dual-network architecture. A depth oracle network predicts optimal sample locations along view rays using a single evaluation, reducing the number of samples needed by focusing on scene surfaces. A shading network then computes radiance for these samples. The approach achieves up to 48x faster inference than NeRF while maintaining or improving image quality, enabling interactive frame rates (15–20 fps at 800x800) on a single GPU without additional memory overhead (Neff et al., 2021).

Advancements in large language models (LLMs), such as GPT and its multimodal variants, have significantly enhanced robot control by enabling intuitive natural language interactions and task planning (Chen et al., 2024; Mao et al., 2023; Wake et al., 2024). These models have been applied in diverse areas, including motion planning for autonomous vehicles, where systems like GPT-Driver reformulate trajectory planning as a language modeling task, achieving high precision and adaptability to novel scenarios while enhancing transparency and interpretability through chain-of-thought reasoning (Mao et al., 2023). Efficient drone control has also been revolutionized, as demonstrated by TypeFly, which employs a specialized language called MiniSpec to reduce response latency and execution time, enabling streamlined operations and real-time adaptability (Chen et al., 2024). Multimodal task planning systems, such as those based on GPT-4V, integrate visual and language inputs to allow robots to learn from human demonstrations, generating detailed task plans and extracting critical affordance information like grasp types and motion waypoints. Despite their potential, occasional hallucinations in generated plans underscore the importance of human supervision in these setups (Wake et al., 2024). Multi-robot coordination has also been transformed with systems like MultiBotGPT, which leverages GPT-3.5 to execute complex, multi-robot tasks such as navigation and target search, achieving superior success rates compared to previous NLP models like BERT (W. Zhao et al., 2024). The shift toward edge-based deployments has addressed critical concerns about privacy, latency, and connectivity. Edge systems, such as those employing GPT-4-Turbo and quantized LLaMA 2, enable real-time decision-making in environments where cloud connectivity is unavailable, such as disaster response and rescue operations (Javaid et al., 2024; Sikorski et al., 2024).

Furthermore, integrating LLMs with unmanned aerial vehicles (UAVs) has opened avenues for dynamic, autonomous control. For instance, LLM-enhanced UAVs can process real-time environmental data, refine decision-making processes, and execute complex tasks such as inspection and surveillance using natural language commands (Choudhury et al., 2024). Despite these advancements, challenges persist, including execution latency due to token generation, the need for robust human feedback loops to refine task planning, and ensuring model reliability under variable conditions (Chen et al., 2024; Wake et al., 2024). Future directions include enhancing multimodal frameworks to better integrate visual and language data, optimizing LLMs for efficient edge deployment, and designing user-centric frameworks to ensure safe, reliable, and scalable robot control systems. These developments aim to expand the applicability of LLMs in diverse domains, from autonomous vehicles to industrial inspections and emergency response (Javaid et al., 2024; Sikorski et al., 2024; Vemprala et al., 2024; Wake et al., 2023). To address the gap between natural language descriptions of object goals and spatial geometric maps, Huang et al. (2023) proposed VLMaps, which directly fuse visual-language features with 3D reconstruction of the physical world.

3. METHODOLOGY

The proposed workflow integrates NeRF and a language model (GPT) for autonomous drone operation in 3D scene analysis. It begins with initializing the drone, which then captures RGB images from a section of a building. Using COLMAP and NeRF, the system estimates camera positions and orientations, followed by 3D registration and point cloud generation. Next, components such as windows are detected, scaled, and their distances calculated. GPT processes this data to issue navigation commands to the drone. The drone then captures close-range data, returns to its orbit path, and proceeds to collect data for the next building section.

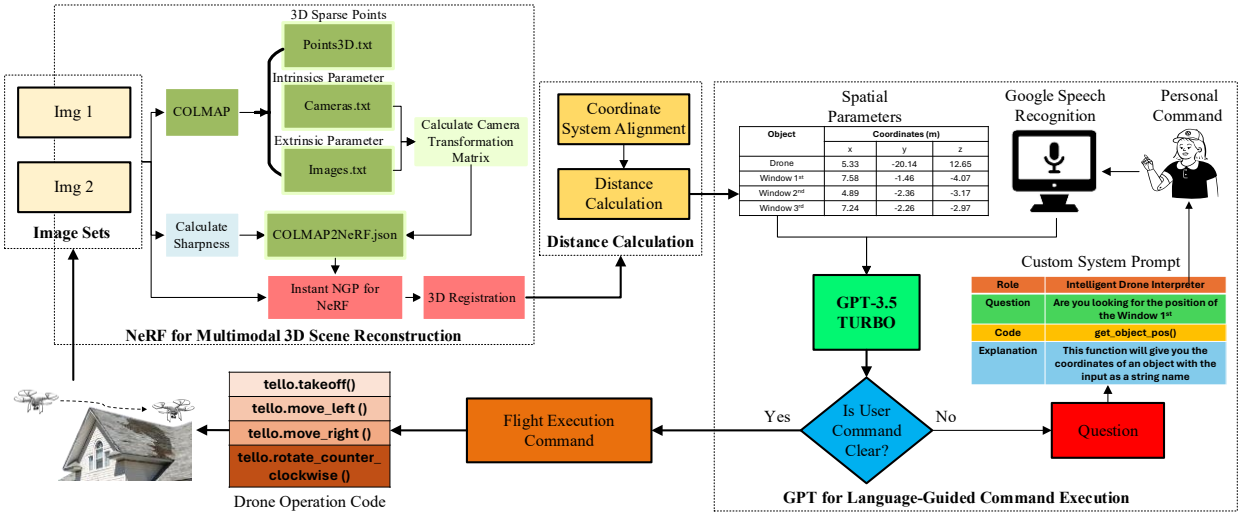

Fig. 1. Framework of Robotic Drone-Assisted Inspection Integrated with NeRF and GPT

3.1 Hardware and Software Setup

The first step in the process is to set up the hardware and software within this drone robotic system. We use programmable drone model to enable customized operation of the drone through scripts. Software Development Kit (SDK) was used to enable the drone's take off, landing, and moving to target objects based on specified coordinates, among other interactive actions. These commands are encapsulated in the command functions file, which contains a series of functions to control the drone's movement using the SDK commands.

To receive natural language commands, convert them into drone operation functions, and process the AI assistant calculation, a local PC serves as a workstation in the workflow. The natural voice recognition function is designed based on the python module, speech recognition. The translation from voice commands to operation functions is based on prompt engineering of GPT-3.5-turbo.

3.2 NeRF for Multimodal 3D Scene Reconstruction

NeRF processing involves two key steps: Colmap for estimating camera parameters and NeRF for 3D scene registration. The NeRF-based reconstruction of a cognitive 3D scene supports the fusion and display of drone-captured imagery data and ML-segmented objects. A Neural Radiance Field (NeRF) is defined as a method that utilizes deep learning to represent a scene as continuous 5D coordinates, namely spatial coordinates (x, y, z) and viewing directions (Θ, ϕ), to construct high-fidelity 3D models from sets of 2D images (Gholizadeh Lonbar & Chen, 2025). These 5D coordinates will be continuously optimized to minimize the error between the synthesized and actual views using a fully connected deep neural network (Gholizadeh Lonbar & Chen, 2025).

COLMAP, an open-source software for SfM and Multi-View Stereo (MVS), is designed to recover camera projections and observes 3D points from a set of images. Its process includes three main stages: feature detection and extraction, exhaustive feature matching and geometric verification, and the incremental reconstruction of structure and motion. This involves identifying significant features within the images, attaching descriptors, and matching these features across all images to register them incrementally and triangulate new points. The software outputs both intrinsic and extrinsic camera parameters. The radial camera model typically used allows for the determination of focal length and the translation vector, along with adjustments for pixel uniformity without considering skewness or distortion. The intrinsic camera matrix K and the extrinsic matrix defined by rotation R and translation parameters are then outlined, enabling the construction of the camera projection matrix M as shown in Eqs 1-3 (Chang et al., 2022.):

$$K = \begin{bmatrix} f_x & 0 & TX \\ 0 & f_y & TY \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$\mathrm{R} = \begin{bmatrix} 1-2s(q_y^2+q_z^2) & 2s(q_xq_y-q_zq_w) & 2s(q_xq_z+q_yq_w) \\ 2s(q_xq_y+q_zq_w) & 1-2s(q_y^2+q_z^2) & 2s(q_xq_z-q_yq_r) \\ 2s(q_xq_z-q_yq_w) & 2s(q_xq_z+q_yq_w) & 1-2s(q_y^2+q_z^2) \end{bmatrix} \tag{2}$$

$$\mathrm{M} = \mathrm{K}[\mathrm{RT}] \tag{3}$$

These parameters are ultimately formatted into text files that can be converted and used as input for NeRF models in a JSON format. The process yields several critical outputs including the x, y, and z coordinates along with quaternion components for rotation, and angular parameters θ and δ. Incorporating these outputs into the intrinsic camera matrix K not only captures focal length and pixel uniformity but also angles of rotation and translation, significantly enhancing the detail and realism of the 3D models generated. The R matrix, derived from quaternion parameters, represents the camera's orientation, transforming points between the camera and the world coordinate systems. The M matrix, or camera projection matrix, combines the intrinsic camera matrix K, which includes constants like focal length and optical center with the extrinsic parameters of rotation R and translation T. This matrix M=K[RT] effectively maps 3D world coordinates to 2D image coordinates, integrating the camera's physical properties and its position in space to accurately project images. (Chang et al., 2021.).

The complicated learning-based NeRF processing has become widely accessible through several platforms, offering unique features for researchers and developers. NVIDIA's Instant-NGP, which used in this paper, provides an optimized implementation of NeRF, allowing fast training and rendering even on consumer GPUs, making it one of the most efficient options available (Chang et al., 2021.). Nerfstudio, an open-source platform, simplifies NeRF use by providing intuitive tools for building, training, and visualizing

NeRF models, and it also offers plugins for integration with popular 3D software such as Blender and Unreal Engine, making it user-friendly for both professionals and a broader audience (Chang et al., 2021.). For those preferring flexibility, PyTorch NeRF implementations offer a customizable way to train NeRF models using popular deep learning frameworks like PyTorch (Chang et al., 2021.). These platforms together provide a robust ecosystem for NeRF applications across various operation systems.

3.3 Distance Calculation

After estimating the camera position and generating 3D model, which captures the 3D coordinates of essential structural elements like windows targeted for close-range inspections, the methodology progresses by identifying these specific elements for a thorough evaluation. Initially, a point cloud, which is extracted from the NeRF model, undergoes scaling to align with available Ground Control Points (GCPs). This scaling is critical as it assures that the point cloud reflects true-to-life dimensions and accurately represents the physical environment. Once the scaling process is complete, the exact geographical positions of both the target window and the drone are established as shown in Fig. 2. This precise positioning is crucial for the subsequent steps where the distance between the drone and the window needs to be calculated. A MATLAB script is employed to process this spatial data, effectively determining the exact separation between the drone and its target. This calculation plays a pivotal role in the inspection process as it informs the subsequent command sequence.

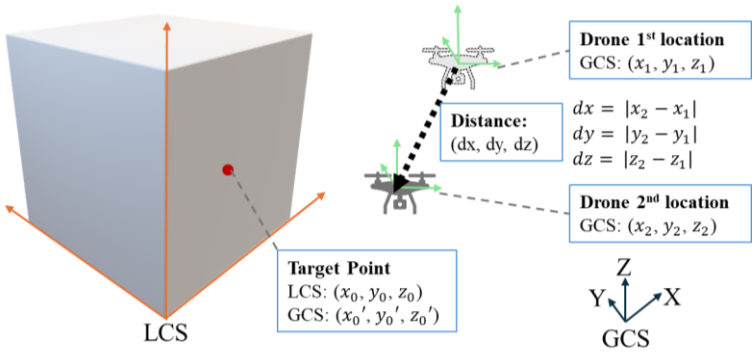

Fig. 2. Coordinate transformation and Distance calculation

3.4 GPT for Language-Guided Command Execution

In this step, building inspectors issue voice commands to guide the drone based on their inspection needs. These voice instructions are first converted into text, forming the initial user input based on the Google Speech Recognition (GSR). GSR is a powerful automatic speech recognition system converting spoken language into written text using deep learning models trained on vast amounts of speech data. It supports real-time audio transcription in over 100 languages with high accuracy.

The drone's and the segmented targets' real-time positional data, such as the coordinates, the vectorized distances, and the pitch degrees, are continuously computed using the perceived scene and 3D localization techniques. These parameters serve as contextual feedback for guiding the inspection process. A well-prepared custom system prompt gives the GPT model a controlled API-like interface, which helps ensure the output contents are only the commands as designed, in predictable and parsable format, and ready for use in downstream code. We define the role of GPT model as an intelligent drone interpreter, plan the structure of the responses in three sections (Question, Code, and Explanation), and format a set of allowed functions for generating drone control commands (such as, take of, land, and get_object_pose). This system message sets the behavior and capabilities of GPT before the user begins interacting.

With the combined prompts including the text transcript form inspector's voice command, the real-time positional contextual feedback, and the custom system prompt fed into GPT model, it outputs a structured response in three sections as designed in the dictionary, including Question, Code, and Explanation (Wen & Chen, 2024). Questions ask clarification if needed, codes give the operation commands in the robot SDK functions format, and explanation justify the commands for user.

4. PILOT STUDY

In this section, we presented the results from our pilot study. The pilot study aimed to explore the above-mentioned methodology and equip the Robomaster TT with natural language processing and autonomous navigation using output from the dense captioning model. Initially, we discuss the outcomes of using COLMAP and NeRF for 3D registration. Following this, we detail the calculated distances, which are used to control the robot's movements effectively. This method ensures precise and reliable robot operation based on accurate spatial data.

4.1 Drone Robot Setup

The first step is to set up the hardware and software of the DJI RoboMaster TT (Tello Talent) drone system, designed for STEAM education. The drone (98 x 92.5 x 41 mm, 87 g) supports a 30 m flight height and 100 m range via Wi-Fi, with an expansion kit (12.5 g) including an open-source controller (49.5 x 32 x 15.2 mm, 9 g), a dot-matrix display and distance-sensing module (35.3 x 31.5 x 8.6 mm, ~3.5 g), and an extension board (1.3 g). It features a 5 MP camera (720p at 30 fps), sensors (vision positioning, infrared, accelerometer, gyroscope, barometer), and supports Scratch, Python, and Arduino programming via Wi-Fi/USB. The ESP32-D2WD-powered controller offers I2C, UART, SPI, GPIO, PWM, and ADC interfaces, with 2.4/5.8 GHz Wi-Fi, Bluetooth 2.1+EDR, and USB 2.0 connectivity, powered by a 1,100 mAh battery for ~13 minutes of flight. The system is controlled via a PC with a 13th Gen Intel Core i9-13900KF at 3.00 GHz, 32.0 GB RAM, and an Intel UHD Graphics 770 GPU (8.0 GB dedicated,15.9 GB shared memory), running a 64-bit system without pen/touch input, ensuring efficient drone programming and control.

4.2 Camera position and NeRF 3D registration

In this section, we present the 3D registration of the building using 16 images (Fig. 3). This process was designed to replicate a realistic scenario involving the scanning of a small part of a building for 3D reconstruction. This approach helps in understanding the practical applications and challenges of implementing 3D modeling techniques in real-world settings.

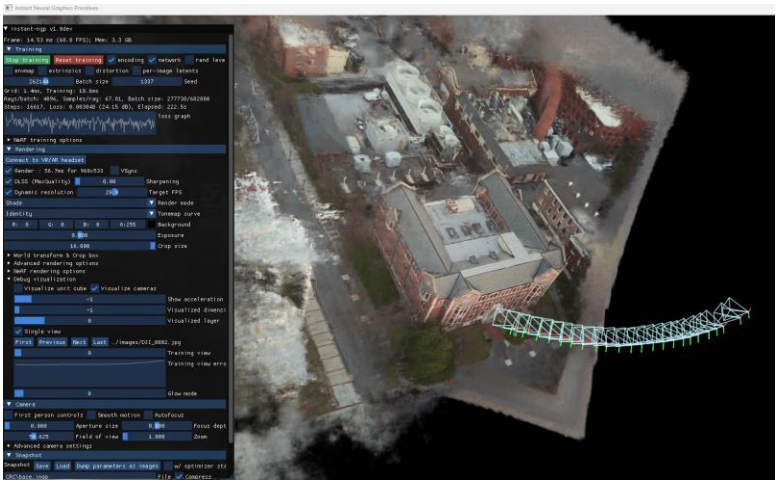

Fig. 3. NeRF Result for 3D registration using Instant-NGP

4.3 Distance Calculation based on NeRF Result

Following this step, the 3D point cloud was utilized to calculate the distance between the drone's location and the building façade (Fig. 4).

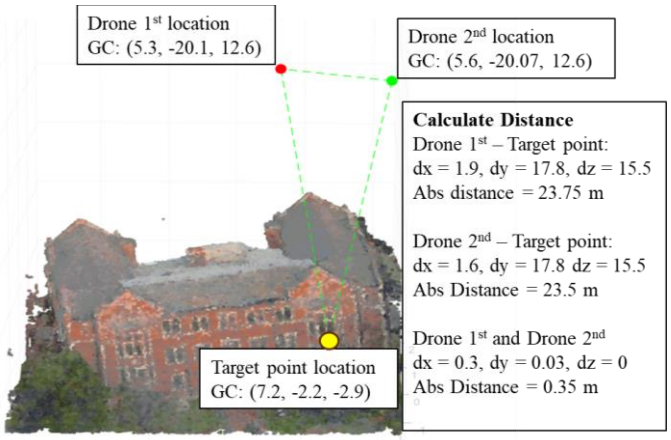

Fig. 4. Distance calculation using 3D point cloud and Robot Control

4.4 Robot Control Based on NeRF and GPT

In the initial phase of the pilot study, we integrated GPT-based control with the drone. This allowed us to relay commands to the drone in natural language, thereby streamlining the interaction process. The GPT model's impressive language understanding capabilities facilitated this communication, interpreting the user's commands and effectively translating them into corresponding code functions for drone execution (Table 1).

Table 1: Code functions for drone execution

Input: Natural Language	Description	Function	Output: Drone Operation SDK Command
"Take off"	Establish a connection with the drone and initiate the SDK mode. the takeoff function commands the drone to start flying.	initialize	tello.connect tello.takeoff
"Fly closer to the first window on the second floor"	Find the target window parameter from real-time spatial feedback. Move drone to specified coordinates at a speed of 30 cm/s.	get_object_pos	tello.move_left tello.move_right tello.rotate_counter_clockwise tello.move_forward
"Take a picture"	capture a frame or image from its current viewpoint, which is then used in subsequent processing.	capture_frame	tello.get_queued_frame tello.frame

During the pilot study, we tested the efficacy of our system in real-world scenarios. For demonstration purposes, we have highlighted an illustrative workflow (see Fig. 5) where the drone operator interacts with the GPT model to execute specific tasks. The process initiates with a command issued by the drone operator: "Can you lift off and find a window?" This query is then processed by our GPT model, which utilizes its trained natural language processing abilities to understand the specifics of the command and prompts a clarification question: "There are 18 targets are objected as window. Which window should the drone approach?" Upon receiving further clarification from the operator ("The target window is first from the right on the 2nd floor."), the GPT model proceeds to integrate the distances calculated from NeRF to prompt a decision: "The drone should move 6.8m right, 9m backward, and descend 2m." This instruction was then translated into a corresponding sequence of code. It generates the appropriate commands for the drone:

takeoff(), followed by final_pos_xyzt() with the relevant coordinates of the desired plant. Simultaneously, the GPT model provides an explanation for the generated code. This workflow demonstrates the system's capability to understand natural language commands, seek clarifications, and generate corresponding drone navigation commands, making drone control a more intuitive and user-friendly process.

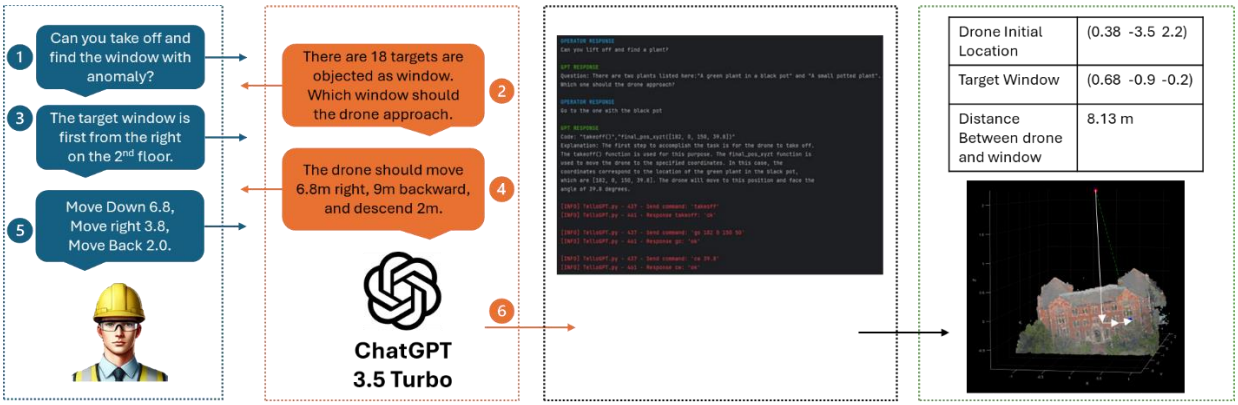

Fig. 5. Autonomous Drone Navigation: Processing Operator Commands with a GPT Model

5. CONCLUSION AND DISCUSSION

In summary, this research has developed a novel method for building inspections that harnesses the power of drone technology integrated with advanced artificial intelligence capabilities. Our methodology, combining the natural language processing of OpenAI's GPT-3.5Turbo and NeRF for targets in adaptive scenario mapping, represents a significant step forward in the field of building inspections. The utilization of GPT has made the control of drones more user-friendly by enabling them to understand and respond to natural language commands. This lowers the technical barrier for operators, making the technology more accessible and easier to adopt. The integration of NeRF facilitates the creation of detailed object dictionaries from images. These dictionaries, when translated into spatial coordinates, allow the drone to better comprehend and navigate its environment, significantly enhancing the precision and accuracy of inspections. The ability to easily localize areas of concern in a building, as provided by our method, is a major advancement in improving the efficiency and effectiveness of inspections. Not only does it reduce the time and resources required, but it also enables quicker responses to potential structural issues, thereby enhancing building safety and longevity. In conclusion, our research offers a more accessible, precise, and user-friendly approach to building inspections. It stands as a significant contribution towards the development of smarter, AI-driven solutions for the construction and maintenance industry. With further refinement and real-world testing, the potential of our methodology to revolutionize building inspections is vast and promising. Future work will explore the real-time reconstruction of the 3D built environment to create a visualized working platform for interactive and informative decision-making throughout the robot-based building inspection process.

ACKNOWLEDGEMENT

This work was supported by the U.S. NSF Grant 2431468, “An AI-Assisted Digital Twin Platform for Advanced Energy Auditing and Retrofitting Analysis in Low-Income Homes at Multiple Scales” and the U.S. NSF Grant 2531557, “Digital Twin and AI-Infused Drones for Energy Retrofitting in Residential Envelopes.”

REFERENCES

Adamkiewicz, M., Chen, T., Caccavale, A., Gardner, R., Culbertson, P., Bohg, J., & Schwager, M. (2022). Vision-Only Robot Navigation in a Neural Radiance World. *IEEE Robotics and Automation Letters*, 7(2), 4606–4613. https://doi.org/10.1109/LRA.2022.3150497

Chang, K., Lin, J., & Zhu, X. (n.d.). *An End-to-End Neural Architecture for View Synthesis: Comparing NeRF + ColMap v. NeRF --*.

Chen, G., Yu, X., Ling, N., & Zhong, L. (2024). *TypeFly: Flying Drones with Large Language Model* (arXiv:2312.14950). arXiv. https://doi.org/10.48550/arXiv.2312.14950

Choudhury, N. R., Wen, Y., & Chen, K. (2024). Natural Language Navigation for Robotic Systems: Integrating GPT and Dense Captioning Models with Object Detection in Autonomous Inspections. *Construction Research Congress 2024*, 972–980. https://doi.org/10.1061/9780784485262.099

From Words to Flight: Integrating OpenAI ChatGPT with PX4/Gazebo for Natural Language-Based Drone Control. (2023). *Proceedings of 2023 the 13th International Workshop on Computer Science and Engineering*. 2023 the 13th International Workshop on Computer Science and Engineering. https://doi.org/10.18178/wcse.2023.06.031

Gholizadeh Lonbar, A., & Chen, K. (2025). NeRF-Enhanced Digital Twin for Building Anomaly Inspection Using Unmanned Aerial Systems (UASs). *In CIB Conferences*, *1*(1), 242.

Huang, C., Mees, O., Zeng, A., & Burgard, W. (2023). *Visual Language Maps for Robot Navigation* (arXiv:2210.05714). arXiv. https://doi.org/10.48550/arXiv.2210.05714

Javaid, S., Fahim, H., He, B., & Saeed, N. (2024). Large Language Models for UAVs: Current State and Pathways to the Future. *IEEE Open Journal of Vehicular Technology*, *5*, 1166–1192. https://doi.org/10.1109/OJVT.2024.3446799

Kerr, J., Kim, C. M., Goldberg, K., Kanazawa, A., & Tancik, M. (2023). *LERF: Language Embedded Radiance Fields* (arXiv:2303.09553). arXiv. https://doi.org/10.48550/arXiv.2303.09553

Mao, J., Qian, Y., Ye, J., Zhao, H., & Wang, Y. (2023). *GPT-Driver: Learning to Drive with GPT* (arXiv:2310.01415). arXiv. https://doi.org/10.48550/arXiv.2310.01415

Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Mueller, J. H., Chaitanya, C. R. A., Kaplanyan, A., & Steinberger, M. (2021). DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum*, *40*(4), 45–59. https://doi.org/10.1111/cgf.14340

Sikorski, P., Schrader, L., Yu, K., Billadeau, L., Meenakshi, J., Mutharasan, N., Esposito, F., AliAkbarpour, H., & Babaiasl, M. (2024). *Deployment of Large Language Models to Control Mobile Robots at the Edge* (arXiv:2405.17670). arXiv. https://doi.org/10.48550/arXiv.2405.17670

Šlapak, E., Pardo, E., Dopiriak, M., Maksymyuk, T., & Gazda, J. (2024). Neural radiance fields in the industrial and robotics domain: Applications, research opportunities and use cases. *Robotics and Computer-Integrated Manufacturing*, *90*, 102810. https://doi.org/10.1016/j.rcim.2024.102810

Tagliabue, A., & How, J. P. (2024). Tube-NeRF: Efficient Imitation Learning of Visuomotor Policies From MPC via Tube-Guided Data Augmentation and NeRFs. *IEEE Robotics and Automation Letters*, *9*(6), 5544–5551. https://doi.org/10.1109/LRA.2024.3386053

Vemprala, S. H., Bonatti, R., Bucker, A., & Kapoor, A. (2024). ChatGPT for Robotics: Design Principles and Model Abilities. *IEEE Access*, *12*, 55682–55696. https://doi.org/10.1109/ACCESS.2024.3387941

Wake, N., Kanehira, A., Sasabuchi, K., Takamatsu, J., & Ikeuchi, K. (2023). ChatGPT Empowered Long-Step Robot Control in Various Environments: A Case Application. *IEEE Access*, *11*, 95060–95078. https://doi.org/10.1109/ACCESS.2023.3310935

Wake, N., Kanehira, A., Sasabuchi, K., Takamatsu, J., & Ikeuchi, K. (2024). GPT-4V(ision) for Robotics: Multimodal Task Planning From Human Demonstration. *IEEE Robotics and Automation Letters*, *9*(11), 10567–10574. https://doi.org/10.1109/LRA.2024.3477090

Wen, Y., Chen, K., & Choudhury, N. R. (n.d.). An Intelligent Robotic Sensing System for Indoor Building System Inspection. In *Computing in Civil Engineering 2023* (pp. 690–698). Retrieved February 1, 2024, from https://ascelibrary.org/doi/abs/10.1061/9780784485224.083

Zhao, H., Ivanovic, B., & Mehr, N. (2024a). *Distributed NeRF Learning for Collaborative Multi-Robot Perception* (arXiv:2409.20289). arXiv. https://doi.org/10.48550/arXiv.2409.20289

Zhao, H., Ivanovic, B., & Mehr, N. (2024b). *Distributed NeRF Learning for Collaborative Multi-Robot Perception* (arXiv:2409.20289). arXiv. https://doi.org/10.48550/arXiv.2409.20289

Zhao, W., Li, L., Zhan, H., Wang, Y., & Fu, Y. (2024). Applying Large Language Model to a Control System for Multi-Robot Task Assignment. *Drones*, *8*(12), 728. https://doi.org/10.3390/drones8120728