Demo: Near Real-time ChatGPT-AR

Yuchen Ding, Pengyuan Zhou
University of Science and Technology of China

ABSTRACT

Augmented reality (AR) applications based on conventional approaches lack the adaptability to cater to different scene requirements and address users' personalized demands effectively. This demon presents ChatGPT-AR, a ChatGPT-powered near real-time voice-to-AR mobile application system, that can create different 3D-augmented spaces using voice commands. Further, ChatGPT-AR enables near real-time contextual editions in AR fashion.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**.

KEYWORDS

Augmented reality, Human-computer interaction

ACM Reference Format:
Yuchen Ding, Pengyuan Zhou. 2023. Demo: Near Real-time ChatGPT-AR. In *The 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys '23), June 18–22, 2023, Helsinki, Finland.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3581791.3597296

1 INTRODUCTION

Adapting different types of AR content to various user demands and environmental conditions often requires significant human assistance and production costs. In addition, the hardware limitations often hinder the user's experience, participation, and freedom [1]. To address these challenges, we propose ChatGPT-AR, a mobile voice interactive AR platform. ChatGPT [1] is a generative pre-trained transformer model for chat and conversation-oriented language tasks. Its superior performance has attracted wide attentions to explore its potential in numerous fields including AR [5]. ChatGPT-AR allows users to generate, modify, and interact with AR models through voice commands, providing a convenient, low-maintenance, and extensible solution for future AR applications.

To assess usability and effectiveness, we recruited five participants of various ages and genders to test ChatGPT-AR in different environments and scenarios, including bedrooms, offices, living rooms, and outdoor areas. Our study focused on the user-centered design approach to evaluate our system's functionality, usefulness, and impact. We have identified critical challenges and opportunities that can be addressed to achieve seamless context-based operations by integrating ChatGPT and AR. Specifically, the major contributions of ChatGPT-AR include: (i) enabling the continuity of context-based operations, (ii) simplifying and streamlining the operation logic for user ease and incorporating voice feedback capabilities to facilitate command revisions, and (iii) reducing the computational burden on mobile devices and ensuring the efficient realization of voice-to-AR operations.

2 SYSTEM OVERVIEW

ChatGPT-AR captures scene information through a monocular RGB camera and utilizes the Whisper[2] pre-trained model for speech-to-text conversion. The system uses the ChatGPT API to obtain corresponding AR scripts and processes them through the ARKit[3] engine, enabling efficient prompt parsing and contextual semantic understanding without the need for a depth camera. We have developed a visual application that enables users to utilize voice commands for AR creation in various scenarios.

2.1 Voice-to-Text Interface

Whisper is a general-purpose speech recognition model, which is a Transformer sequence-to-sequence model trained on various speech processing tasks. Our goal is to utilize this pre-trained model for efficient and fast speech-to-text conversion while maintaining better privacy and security by storing the model locally and avoiding the need to upload audio streams to the cloud.

2.2 Prompt Engineering Module

The voice-converted text needs to be converted into a task-oriented prompt through prompt engineering [4]. This includes selecting the format and wording of prompts, deciding the information to provide in prompts, and identifying the most effective prompts to achieve the desired results. In our system, we have defined ChatGPT as a code generator that exclusively produces anonymous JavaScript functions. Furthermore, we have established specific details crucial to its operation, including function whitelisting and generation rules. Remarkably, the selection of prompt engineering module is scalable and can be personalized with domain-specific training according to different use cases.

2.3 ChatGPT Interface

Initially, ChatGPT is employed to extract the object name from the raw text. For example, if the user says, "Place a football on the table in front of me", ChatGPT detects "football" as the model name to generate. For this purpose, we have created a function utilizing ChatGPT to return JavaScript codes within an anonymous function containing detailed instructions for producing the model element and adjusting its position, among other features. The design of anonymous functions offers a malleable approach to code organization, with benefits including simplified and refactored code.

[1] https://openai.com/blog/chatgpt

[2] https://github.com/ggerganov/whisper.cpp

[3] https://developer.apple.com/augmented-reality/arkit/

MobiSys '23, June 18–22, 2023, Helsinki, Finland

ACM ISBN 979-8-4007-0110-8/23/06.
https://doi.org/10.1145/3581791.3597296

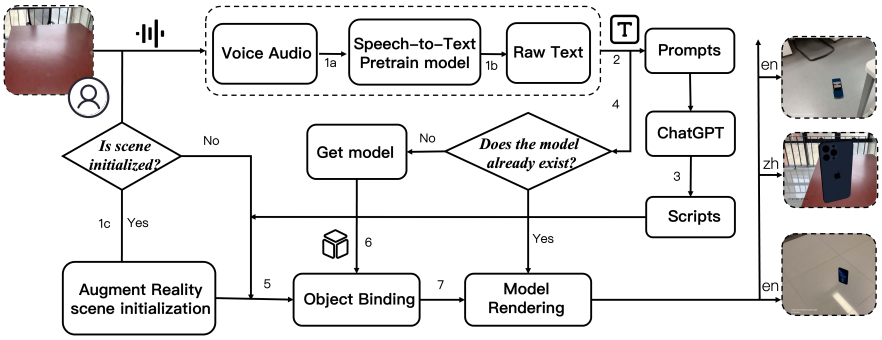


Figure 1: System architecture. ChatGPT-AR inputs the user's voice audio stream into the Whisper model (step 1a) to obtain the unprocessed text (1b) while initializing the AR scene (1c). Then, the raw text is processed via prompt engineering to return prompts (2). Subsequently, the prompts are matched to the corresponding scripts while calling ChatGPT (3). Step 4 Finally, ChatGPT-AR acquires a model for object binding (6) using the scripts (5) and presents the rendered model (7).

2.4 Contextual AR Edition

Our system utilizes the ARKit backend to conduct a local search for the target object model. If the model is not stored locally, it will be immediately searched and downloaded from the cloud. The model is then bound to the created entity, and an anonymous function is used to set its position, angle, and object behavior. The ChatGPT-AR system provides a broad-ranging experience by being mobile-device-compatible (e.g., for iPhone and iPad), allowing users to use the system in diverse environments. Moreover, it facilitates context-aware AR edition through ChatGPT, thus enabling context memory. This feature empowers users to make secondary modifications to the created AR entity, such as relocating it from the desktop to the bottom surface or maneuvering it up and down along the z-axis.

2.5 End to End Performance

For interactive systems, the delay is an important end-to-end performance metric that reflects the system's responsiveness to user commands. ChatGPT-AR system's delay mainly contains three parts: (i) speech-to-text conversion, (ii) cloud model retrieval, and (iii) waiting for ChatGPT response. In our tests, the delay of (i) is generally less than 4s; (ii) depends mainly on network speed and model size, e.g., 2 seconds to download the iPhone model with a 10 Mbps bandwidth; and (iii) ranges from 2s to 5s.

3 DEMONSTRATION

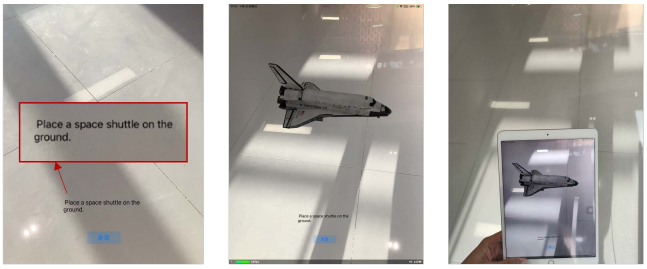
We have currently implemented ChatGPT-AR in iOS 14.0+ using ARKit, a framework developed by Apple for building augmented reality applications. The test environment is an iPad Air with iOS 16.3 and Xcode 14.0. All actions by the user are performed on a mobile device. Implementation of ChatGPT-AR on Android can employ the same architecture using ARCore.

This live demo[4] includes the following steps:

Step 1: The user opens the app and waits for the AR environment to complete initialization.

Step 2: The user clicks the recording button, speaks the command, and then clicks the stop button. ChatGPT-AR then uses Whisper for voice-to-text conversion and sends the result after prompt engineering to ChatGPT, waiting for the system response.

Step 3: After the system completes parsing the prompts and scripts, ChatGPT-AR renders the corresponding AR model and repeats step 2.



(a) (b) (c)

Figure 2: An example view of ChatGPT-AR in the real world. For the demonstration, we used an iPad to display (a) voice-to-text, (b) a model of a space shuttle, and (c) a real-world perspective.

4 FUTURE WORKS

We plan to create 3D object models from scratch or directly edit attribute values on existing models using methods like diffusion models [3] or NeRF [2]. The major challenge is 3D generation in real-time on consumer-grade terminals. More user studies could also be conducted to assess the system's effectiveness in real-world scenarios and identify critical bottlenecks.

REFERENCES

[1] Lik-Hang Lee and et al. 2021. All One Needs to Know about Metaverse: A Complete Survey on Technological Singularity, Virtual Ecosystem, and Research Agenda. *ArXiv* abs/2110.05352 (2021).
[2] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.
[3] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *ArXiv* abs/2209.14988 (2022).
[4] Denny Zhou and et al. 2022. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. *ArXiv* abs/2205.10625 (2022).
[5] Pengyuan Zhou. 2023. Unleashing chatgpt on the metaverse: Savior or destroyer? *arXiv preprint arXiv:2303.13856* (2023).

[4] https://youtu.be/IqTxKnfd--8