Journal of Digital Media & Culture Technology Vol.6, No.1, June (2026), pp.69-76
ISSN 2983-0664(p) 2983-0850(e) *http://dx.doi.org/10.29056/jdmct.2026.06.07*

Voice-to-Mesh: A Mixed Reality Pipeline for 3D Content Generation Using Text-to-3D Model

Yejin Kim[1†] , Minjun Kang[2†] , Suhyeon Kim[3], Younhyun Jung[4*]

Abstract

Recent advances in text-to-3D generation have enabled the creation of 3D assets from natural language prompts. However, Mixed Reality (MR) authoring workflows still rely on manually prepared 3D assets, making it difficult to generate and utilize new objects. In addition, generated mesh assets often vary in scale, object origin, orientation, and coordinate conventions, requiring additional normalization and alignment for direct placement in MR environments. This additional processing interrupts the continuity of the content creation workflow. To address these limitations, we present an end-to-end pipeline that generates 3D assets from multilingual voice commands and automatically normalizes the generated assets for use in MR environments. Speech-to-Text recognition and Large Language Model-based parsing convert natural language into structured commands. The proposed pipeline enables generated mesh assets to be immediately utilized, stored, and reused within the MR environments. Through a storytelling scenario, we demonstrate the feasibility of interactive MR content creation.

Keyword : Mixed Reality, Text-to-3D generation, Generative Content, Interactive MR

1. Introduction

Mixed Reality (MR) combines the physical world with virtual objects, providing an immersive and interactive environment. Unlike conventional 2D display-based environments, MR supports more intuitive interaction through modalities such as hand gestures, eye gaze, and spatial movement. Accordingly, MR devices have been increasingly adopted in a wide range of interactive and immersive content

1 School of Computing, Gachon University, Gyeonggi-do, Republic of Korea [Graduate Student]
e-mail: vvvsfhk@gachon.ac.kr
2 School of Computing, Gachon University, Gyeonggi-do, Republic of Korea [Undergraduate Student]
e-mail: alswns1119@gachon.ac.kr
3 School of Computing, Gachon University, Gyeonggi-do, Republic of Korea [Graduate Student]
e-mail: kih629@gachon.ac.kr
4 School of Computing, Gachon University, Gyeonggi-do, Republic of Korea [Professor]
e-mail: younhyun.jung@gachon.ac.kr (Corresponding author)

† These authors equally contributed to this manuscript

* This research was supported by the Regional Innovation System & Education (RISE) program through the Gyeonggi RISE Center, funded by the Ministry of Education (MOE) and Gyeonggi-do, Republic of Korea. (2025-RISE-09-A01), and by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (RS-2025-00554526).

Received(April 21, 2026), Review Result(1st: May 9, 2026), Accepted(June 13, 2026), Published(June 30, 2026)

experiences.

With recent advances in generative AI, content generation has expanded beyond images toward 3D representations that incorporate geometric structure and appearance. In particular, text-based 3D mesh generation enables the synthesis of high-quality 3D mesh assets with shape and texture information from natural language prompts. Existing text-to-3D approaches include diffusion-based models [1], NeRF-based models [2], and models provided through commercial APIs, such as Meshy [3] and LumaAI [4]. These API-based models make mesh generation more accessible by allowing users to generate 3D mesh assets from simple text prompts.

However, current methods of using 3D assets in MR environments remain limited. Most MR applications rely on pre-authored 3D assets, such as GLB or FBX files, that are prepared and imported into the application. This workflow makes it difficult to create new virtual objects within the MR space in response to user requests. When a new asset is needed, users must perform separate modeling work, convert the file, and re-import it into the MR project. Furthermore, mesh assets generated by text-to-3D models often vary in scale, origin, and coordinate systems, requiring additional post-processing such as normalization and coordinate alignment. As a result, users cannot immediately utilize the desired objects, and must repeatedly perform modeling, file conversion, and repositioning steps, thereby disrupting the continuity of the MR content authoring workflow.

To address these limitations, we propose an end-to-end pipeline that integrates Text-to-3D API based mesh generation, asset loading, scale normalization, and user interaction. When a user enters a voice command through the MR device, the system parses the command, automatically generates a 3D mesh, normalizes its scale, and presents it in front of the user. The generated object can be freely manipulated using hand gestures. The entire process was implemented and evaluated on HoloLens 2, demonstrating its ability to support efficient MR content creation scenarios, including MR-based storytelling.

2. Methodology

2.1 Mesh Generation in XR

Tong et al. [5] proposed MS2Mesh-XR, which generates 3D meshes by combining hand-drawn sketches and voice prompts. The system creates a high-resolution reference image from the user input and reconstructs it into a 3D mesh. Although this supports rapid and user-friendly mesh generation, it requires users to provide a sketch. Our approach enables mesh generation using only natural language commands. Furthermore, our pipeline allows the generated objects to be manipulated in MR

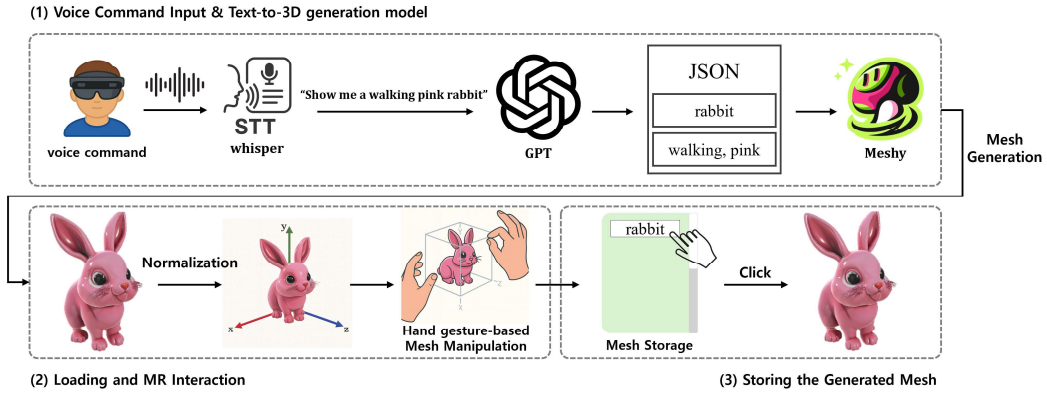
environments.

2.2 Content Creation in XR

Kim et al. [6] introduced a Codeless Content Creator System that allows users to configure MR content by selecting and placing previously uploaded 3D assets without programming knowledge. This system is effective for enabling users to compose MR scenes using existing models. However, users can only use 3D assets that are stored on the server. In contrast, Our system provides a generative MR pipeline that creates new 3D mesh assets from natural language input and supports their placement and interaction within the MR environment.

3. Implementation

We designed an end-to-end pipeline that enables Text-to-3D asset technology to be used within an MR environment. The proposed system consists of three main components: (1) voice-based command input and Text-to-3D asset generation module, (2) mesh loading and interaction within the MR environment, and (3) storage and reuse the generated meshes. The overall architecture of the proposed Text-to-3D mesh generation MR pipeline is illustrated in [Fig. 1]. The following sections describe the structure and functionality of each component in detail.



[Fig. 1] Overview of the Proposed Text-to-3D mesh generation MR Pipeline

3.1 Voice Command Input and Text-to-3D Generation Model

The Mixed Reality Toolkit (MRTK) [7] for HoloLens 2 primarily supports English-based voice

commands, making it difficult to accurately recognize the speech from non-English users. In particular, frequent errors can occur when users speak a voice command in various languages such as Korean, Chinese, or Japanese. It is also challenging to interpret commands that reflect complex sentence structures or contextual information. These limitations reduce the accessibility of MR content creation.

To address this, we designed a multilingual voice command processing pipeline that combines a Speech-to-Text (STT) model and a Large Language Model (LLM). When a user inputs a voice command, the Whisper API [8] records it and converts the multilingual sentence into text using STT model. The converted text is then passed to the GPT-4o-mini API [9], which is guided by a designed prompt. The GPT model analyzes the sentence semantically and interprets the user's intent in a structured form.

The GPT model automatically parses the command into JSON format required by the Meshy API. The Meshy API requires explicit input field such as target object to be generated and style-related descriptions, including appearance, texture, and color. For accurate object generation, these attributes need to be provided in a structured format. Therefore, the GPT model extracts the target object and style-related cues from the user's natural language command and reconstructs them into JSON fields.

The Meshy API generates a 3D mesh in GLB format based on the received JSON parameters. When the generation is completed, the server automatically downloads the generated asset and passes it to the MR application, where it can be used as a virtual object in the MR space. Through this end-to-end pipeline, users can generate and place desired 3D assets using natural voice commands with reduced language constraints.

3.2 Loading and MR Interaction

The Meshy API tends to generate meshes with varying scales and proportions depending on the input command and the model's internal optimization process. Such inconsistency can reduce spatial consistency in the MR environment. When excessively large or small objects are loaded, they may also decrease visual stability and ease of manipulation within the user's MR field of view. Therefore, a normalization process is required to handle generated meshes consistently within a common spatial coordinate system.

To address this, we designed an automatic normalization pipeline based on glTFast [10]-based mesh loading and a container structure. The generated mesh is loaded into Unity through glTFast and placed inside the normalization container. This container computes the actual spatial extent of the entire mesh by integrating the bounding boxes of its sub-meshes. Based on the minimum and maximum bounds of

the object, the system automatically performs scale normalization and center alignment regardless of the original mesh size. The normalized container is then placed at a certain distance in front of the camera, providing consistent visualization of Text-to-3D assets generated at different scales.

The normalized mesh container is integrated with the MRTK3-based interaction system of HoloLens 2. The container supports hand tracking and gesture recognition functions provided by MRTK3, allowing users to interact with the generated object through actions such as grabbing, moving, rotating, and resizing. This design ensures that various generated 3D assets are placed according to a consistent spatial reference in the MR environment, while allowing users to adjust them as intended.

3.3 Storing the Generated Mesh

Since the Text-to-3D model generates a new 3D mesh for each request, reusing the same object requires repeated model calls. This causes unnecessary API requests and delays, which can interrupt the MR content authoring workflow.

We implemented a mesh management system that automatically saves generated meshes and enables their immediate reuse through a scroll view UI. The system stores the GLB file generated by the Meshy API in local storage and creates a button for each mesh in the scroll view UI panel. The label of each button is automatically assigned based on the target extracted by GPT from the voice command, allowing users to intuitively browse and select previously generated objects by scrolling through the list.

When the user selects a specific button, the corresponding GLB file is immediately loaded and placed in the MR space. In particular, when the same mesh needs to be placed multiple times, users can instantiate multiple copies simply by selecting the same button repeatedly, without repeated API calls. As a result, the proposed system can minimize workflow interruptions caused by object generation delays in complex scene authoring tasks, such as MR-based storytelling.

4.Utilization Examples

4.1 Storytelling - The Adventures of the Rabbit and the Duck

To demonstrate the applicability of the proposed system to content authoring, we conducted two storytelling-based content generation experiments.

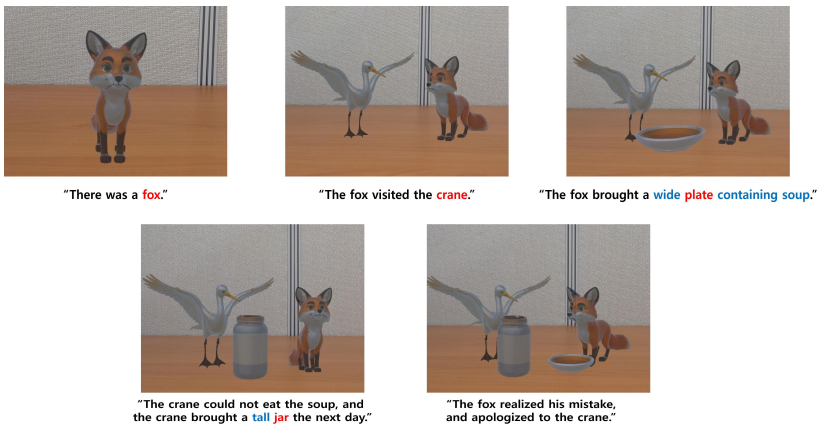
In the first experiment, we used the storytelling scenario “The Adventure of the Rabbit and the Duck,” and the input sentences and corresponding generated mesh results are summarized in [Fig. 2].

[Fig. 2] "The Adventure of the Rabbit and the Duck": input sentences and the corresponding generated mesh results (red: target, blue: style).

Each sentence is converted into text by Whisper, parsed by GPT, and then passed to the Meshy API to generate a 3D asset of the corresponding object. Since objects generated from earlier sentences can be reused in subsequent scenes, already generated objects are excluded from the target field, and newly introduced objects in each sentence are used as targets for mesh generation. For example, in the first sentence of [Fig. 2], "There was a mysterious door," the target is parsed as "door," and the style is parsed as "mysterious," resulting in the generation of a door mesh. When the second sentence, "There was a pink rabbit walking in front of the door," is provided, the door object is not selected again because it has already been generated. Instead, GPT focuses on the newly introduced object, the rabbit, and parses the target as "rabbit" and the style as "pink, walking." In cases where no new object appears, as in the fifth sentence, the target is returned as null to prevent unnecessary mesh generation.

4.2 Storytelling - The Fox and The Crane

As shown in [Fig. 3], the second storytelling experiment demonstrates that the proposed system can extract newly introduced objects, such as a fox, a crane, a plate, and a jar, while avoiding unnecessary mesh generation when no new object appears. For example, when a sentence includes an adjective phrase modifying a noun, such as "a plate containing soup" in the third sentence, the prompt is designed to extract the core object, "plate," as the target and the additional description, "containing soup," as style information. In addition, it is confirmed through the fifth sentence that the system returns null when no new object appears, even if the storytelling context continues.



[Fig. 3] "The Fox and the Crane": input sentences and the corresponding generated mesh results (red: target, blue: style).

5. Conclusion

In this paper, we present an end-to-end pipeline for integrating text-based 3D mesh generation into MR content authoring. The proposed system combines voice command-based mesh generation, automatic normalization, MR interaction, and mesh management functions that support storing, managing, and reusing generated assets.

The storytelling-based experiments demonstrate that the proposed pipeline can generate 3D mesh assets from user voice commands and visually reconstruct them within the MR space. However, the generation time of the Text-to-3D model varies depending on the complexity of the object geometry and texture. While simple meshes require approximately 3 to 4 minutes, assets with high-resolution textures can take up to 10 minutes to generate. This indicates that the current system is not yet suitable for strict real-time generation.

Future work will focus on improving mesh generation speed, enhancing texture detail, and integrating post-processing techniques to provide a more seamless MR content creation environment.

References

[1] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "DreamFusion: Text-to-3D using 2D diffusion," unpublished.

[2] Z. Li et al., "Neuralangelo: High-fidelity neural surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, Canada, Jun. 18-22, 2023, pp. 8456-8465, doi: 10.1109/CVPR52729.2023.00817.

[3] Meshy.ai, "Meshy: Text-to-3D and image-to-3D generation platform," *meshy.ai*, https://www.meshy.ai/discover (accessed May 28, 2026).

[4] Luma Labs, "Luma AI: NeRF-based 3D capture & generation platform," *lumalabs.ai*, https://lumalabs.ai/ (accessed May 28, 2026).

[5] Y. Tong, Y. Qiu, R. Li, S. Qiu, and P. A. Heng, "MS2Mesh-XR: Multi-modal sketch-to-mesh generation in XR environments," in *IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, Lisbon, Portugal, Jan. 27-29, 2025, pp. 272-276, doi: 10.1109/AIxVR63409.2025.00052.

[6] S. Kim, D. Lee, J. Park, M. Song, and Y. Jung, "Codeless content creator system: Anyone can make their own mixed reality content without relying on software developer tools," in *SIGGRAPH Asia Posters*, Daegu, Republic of Korea, Dec. 6-9, 2022, pp. 1-2, doi: 10.1145/3550082.3564194.

[7] Microsoft, "Mixed reality toolkit (MRTK) for Unity," *learn.microsoft.com*, https://learn.microsoft.com/ko-kr/windows/mixed-reality/mrtk-unity/mrtk3-overview/ (accessed May 28, 2026).

[8] OpenAI, "Speech to text," *platform.openai.com*, https://platform.openai.com/docs/guides/speech-to-text (accessed May 28, 2026).

[9] OpenAI, "Text generation," *developers.openai.com*, https://developers.openai.com/api/docs/guides/text (accessed May 28, 2026).

[10] A. Atteneder, "glTFast," *github.com*, https://github.com/atteneder/glTFast (accessed May 28, 2026).