SceneForge AI: MCP, A Multimodal Agent for Generative 3D Art and Scripting in Blender

(SceneForge AI: 블렌더에서 생성형 3D 아트와 스크립팅을 수행하는 멀티모달 MCP 에이전트)

이세훈, 윤동의, 박준형, 이주엽

**초록**

본 프로젝트는 자연어 명령을 통해 Blender를 제어해 3D 모델링을 할 수 있는 AI 기반 멀티모달 에이전트를 제안한다. 사용자의 입력은 서버를 통해 Python 코드로 변환되어 Blender 내부 애드온에서 자동 실행되며, 직관적인 3D 모델링 환경을 제공한다. 시스템은 LLM 기반 명령 생성, 에러 대응, 멀티모달 입력 기능을 포함하며, Claude 기반 MCP 오픈소스 구조를 벤치마킹하고 GPT, Gradio, Hugging Face 등 다양한 도구를 활용하여 구현되었다.

**1. 소개 (Introduction)**

Blender는 강력한 기능을 갖춘 오픈소스 3D 툴이지만, 복잡한 UI와 높은 학습 곡선으로 인해 비전문가에게는 진입 장벽이 높다. 기존 자동화 시스템은 주로 단순 매크로나 기본 오브젝트 배치에 그쳤다.

본 프로젝트는 이러한 한계를 해결하기 위해 자연어 명령으로 Blender를 제어하는 3D 시스템 MCP를 설계하였다. Claude 기반 오픈소스를 확장하고, 최신 LLM, RAG, Text-to-Image 및 Text-to-Mesh 모델을 통합함으로써, 사용자는 고급 기능을 몰라도 직관적인 언어만으로 객체 생성, 배치, 텍스처 적용이 가능하다.

**2. 시스템 구조 및 동작 방식**

MCP 시스템은 크게 네 가지 핵심 구성 요소로 이루어진다:

1. **자연어 명령 입력**: 사용자는 "나무 책상을 방 한가운데 놓아줘"와 같이 직관적인 명령을 입력한다.
2. **LLM + RAG 기반 파이썬 코드 생성기**: 명령을 해석한 후, Blender 파이썬 API 문서를 기반으로 RAG를 통해 코드 템플릿을 검색하고 LLM이 최종 스크립트를 구성한다.
3. **Stable Diffusion 기반 생성기**: 명령에 필요한 텍스처(.png)와 3D 객체(.obj)는 각각 Text-to-Image, Text-to-Mesh 모델을 통해 실시간 생성된다.
4. **실행 및 시각화**: 최종적으로 생성된 파이썬 코드는 Blender 내부 애드온에서 자동 실행되며, 객체는 씬(scene)에 배치되고 텍스처가 적용된다.

사용자는 후속 명령을 입력해 객체 수정, 위치 이동, 텍스처 교체 등을 반복할 수 있다.

**3. 기술적 기여**

기존 Claude MCP 구조는 구버전 API 호출과 문서 부족으로 인해 오류가 잦고 코드 정확도에도 한계가 있었다. 이를 개선하기 위해 본 시스템은 Blender 최신 문서 1,761페이지, 9,429개 항목을 임베딩하고, all-MiniLM-L6-v2와 FAISS 인덱싱 기반의 의미 기반 문서 검색을 구현하였다. 이로써 생성되는 코드는 최신 API 흐름에 부합하며, 고급 함수와 파라미터 조정이 반영된 정교한 구조로 개선되었다.

또한 MCP는 기존 시스템의 단순 색상 매핑 및 primitive 객체 배치 수준을 넘어, Stable Diffusion 기반 텍스처 생성과 Text-to-Mesh를 활용한 객체 생성을 통해 시각적 품질과 표현력을 크게 향상시켰다.  
[그림 1]은 본 논문의 모델을 이용하여 텍스처를 적용하고 복잡한 형태의 Mesh를 생성한 것을 보여준다.

스크린샷, 디자인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다. 스크린샷, 우드, 예술, 야외이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

[그림1]

**4. 결론 (Conclusion)**

본 논문은 자연어 기반 3D 제어 시스템 MCP의 설계와 구현을 소개하였다. MCP는 LLM, 생성형 AI, RAG 기술을 결합하여 복잡한 Blender 작업을 자연어만으로 수행할 수 있도록 하며, 기존 시스템의 한계를 효과적으로 개선하였다. Stable Diffusion과 Text-to-Mesh를 통해 실제 텍스처와 3D 객체를 자동 생성함으로써 시각적 표현력과 모델링 자유도가 향상되었고, 의미 기반 문서 검색으로 최신 API에 맞는 정교한 코드 생성도 가능해졌다.

또한 [그림 2]는 MCP가 기존 모델에 비해 사용자 만족도, 응답 속도, 정확도, 텍스처 해상도 등 다양한 항목에서 고르게 우수한 성능을 보였음을 보여준다. 이를 통해 MCP는 정성적 편의성과 정량적 성능 모두에서 유의미한 향상을 입증하였다.

이로써 MCP는 단순 명령 해석을 넘어 실시간 의도 반영이 가능한 고도화된 3D 생성 시스템으로 발전하였으며, 향후에는 음성 명령과 실시간 피드백을 지원하는 대화형 플랫폼으로의 확장이 기대된다.

도표, 텍스트, 그래프, 라인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다. 도표, 디자인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

[그림 2]

**참고문헌 (References)**

1. Hugging Face. “Diffusers Library.” <https://github.com/CompVis/stable-diffusion>
2. Hugging Face. “diffusers GitHub Repository.” <https://github.com/huggingface/diffusers/tree/main>
3. Blender Foundation. “Blender Python API.” <https://docs.blender.org/api>
4. Zhengyi. “LLaMA-Mesh.” <https://huggingface.co/Zhengyi/LLaMA-Mesh>
5. Hugging Face. “Stable Diffusion v1.5.” <https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5>
6. Ahuja, Sid. “Blender-MCP.” <https://github.com/ahujasid/blender-mcp>
7. Blender Foundation. “Blender Python API (Current).” <https://docs.blender.org/api/current/index.html>
8. CompVis. “Stable Diffusion.” <https://github.com/CompVis/stable-diffusion>
9. Daddy N Kids Makers. “블렌더 자동화 실험 기록.” <https://daddynkidsmakers.blogspot.com>