PROJECT DOCUMENTATION

Project Title : Convert Photo or Text to a Simple 3D Model

Author :

Krishnagopal Jay

Kgjay2004@gmail.com

02.05.2025

Objective :

This prototype demonstrates the ability to transform a simple **text prompt**—such as *"a small toy car"*—into a **3D mesh model** using open-source machine learning techniques. The system output downloads an .obj file and provides a real-time 3D visualization. This project showcased capabilities in AI-driven 3D modeling and independent project execution.

Technology :

- Programming Language : Python 3.11
- Environment Management : virtualenv
- Core Libraries :
    - Torch
    - Shap-e
    - Trimesh
    - Pyrender
    - Matplotlib
    - Pyyaml

Workflow :

- **Input :**
  - A short text description(e.g., “a small toy car”)
- **Processing :**
  - Load pretrained shap-e models(text-to-latent diffusion + decoder).
  - The prompt is passed to shap-e to generate latent 3D representations.
  - Latents are decoded into 3D models with color and shape.
- **Output :**
  - A downloadable 3D model in .obj format.
  - We visualized the 3D model with the Blender and also with the Trimesh library.

Theoretical Overview :

This project uses OpenAI’s shap-e, a diffusion-based model that generates 3D shapes directly from text prompts. The process involves two key steps:

- **Text-to-Latent Diffusion**: A pretrained model interprets the prompt and generates a compressed 3D representation (latent code).
- **Latent-to-Mesh Decoding**: The latent is decoded into a colored 3D mesh, which is saved as an .obj file.

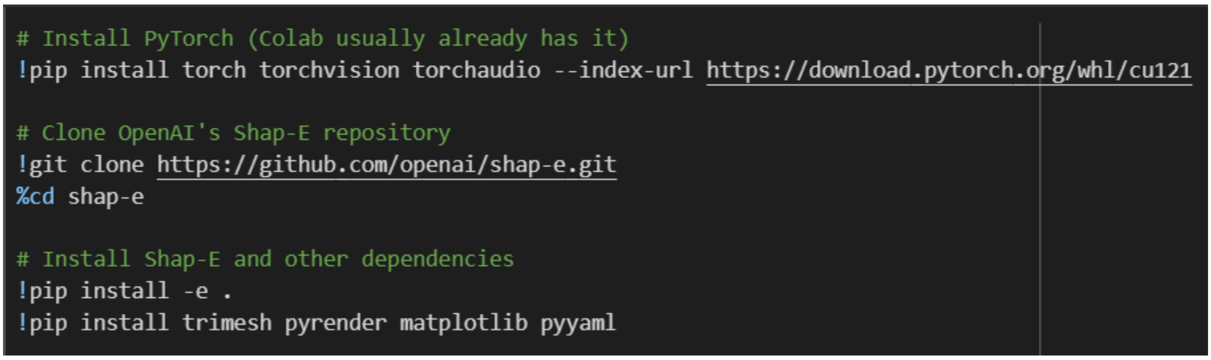
Unlike voxel-based methods, Shap-e enables efficient and high-quality mesh generation suitable for prototyping and 3D content creation.

To visualize and manage the generated models, the project uses **trimesh**, a lightweight Python library that supports loading, displaying, and exporting 3D files.

Key Features :

- Accepts natural language prompts.
- Uses pretrained models for zero-shot 3D shape generation.
- Exports in industry-standard 3D formats.
- Real-time 3D plot rendering for quick feedback.

Requirements :



```
# Install PyTorch (Colab usually already has it)
!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121

# Clone OpenAI's Shap-E repository
!git clone https://github.com/openai/shap-e.git
%cd shap-e

# Install Shap-E and other dependencies
!pip install -e .
!pip install trimesh pyrender matplotlib pyyaml
```

Future Enhancements :

- Web UI with Streamlit for interactive use.
- Integration with text expansion models to improve prompt-to-mesh quality.

Conclusion :

This project effectively uses OpenAI's cutting-edge Shap-e model to turn text into expressive 3D model. The end-to-end prototype demonstrates a modern approach to generative 3D modeling and highlights skills in AI tooling, 3D graphics, and independent execution.