Project Report on Sketch to Scene

Yanzhe Kong

May 2025

Project Page: https://github.com/LTDuckie/sketch2scene
Demo Video: https://www.bilibili.com/video/BV1dQEeznEub/

1 Introduction

The "Sketch to Scene" project aims to transform hand - drawn sketches into 3D scenes. It involves multiple stages including sketch-to-image conversion, image refinement, image segmentation and 3D model generation. This project combines different technologies and libraries to achieve the goal, providing an innovative way to bring 2D sketches to life in a 3D space.

2 Background Research

2.1 Sketch to Image Synthesis

Sketch-to-image synthesis aims to transform abstract hand-drawn sketches into realistic colored images. The core challenge in sketch-to-image synthesis lies in recovering complete geometric structures from abstract, sparse sketches and generating high-fidelity colored images.

Shape Completion: Shape completion is fundamental for image synthesis, aiming to infer complete object structures from fragmented or abstract lines. SketchyGAN(Chen and Hays 2018) introduces Masked Residual Units (MRUs) and a two-stage strategy. They combine edge maps and sketch data augmentation to generate diverse images. Also, Interactive Sketch and Fill(Ghosh et al. 2019) adopts a shape generator to appearance generator pipeline. Experiments show that separating shape completion and appearance generation reduces error accumulation, achieving an F-Score of 78.21 on the Sketchy dataset. And so Multi-modal Completion and Interaction is introduced.

Diffusion: Diffusion models generate images via iterative denoising, excelling in handling the abstractness and ambiguity of sketches. DiffSketching(Wang et al. 2023) pioneers diffusion-based image synthesis by aligning sketch-image semantic features where Cross-domain Constraints get important.

Current methods struggle with small objects and complex interactions, for instance, buttons and occlusions.

2.2 Image to Image Generation

Image-to-Image Translation is to transform the content, style, or modality of an input image into a target domain. It is by now a very popular tool in industries. We focus on two typical area of image to image generation since we care more about whether the generated image is realistic.

Style Transfer: Style transfer aims to transfer the artistic style such as brushstrokes, color palettes and textures of a reference image to target content while preserving its semantic structure. Optimization-Based Style Transfer Proposed in neural style transfer using VGG networks (Gatys, Ecker, and Bethge 2016) extracts content features and style features is one way of the style transfer.

Condition Guided: Condition guided learning controls the generation process using auxiliary information. To reach flexible control on the generated image with multi-modal conditions, ControlNet(L. Zhang, Rao, and Agrawala 2023) enhanced diffusion models by adding conditional layers which support segmentation maps and texts.

These two could be combined as StyleStudio(Lei et al. 2024) realized Text-Driven Style Transfer recently. However, current models still lack the ability to generate splendid details.

2.3 3D Objects from an Image

Single-image object modeling and scene reconstruction involve recovering 3D geometric structures, semantic labels, and even instance segmentation from a single 2D image.

Volumetric Representation-Based Reconstruction: It uses voxel grids or implicit functions to model 3D geometry directly, which learn 2D-to-3D mappings leveraging CNNs or Transformers. Large Pose 3D Face Reconstruction(Jackson et al. 2017) proposes a Volumetric Regression Network (VRN) with stacked Hourglass Networks to fuse multi-scale features.

NeRF with Diffusion Priors for Fine-Grained Geometry: It Combine Neural Radiance Fields (NeRF) with diffusion models to leverage 2D generative priors for multi-view consistency and texture realism. Magic123(Jackson et al. 2017) adopts Instant-NGP NeRF and employs Deep Marching Tetrahedra (DMTet) for high-resolution mesh refinement to get the models.

Transfer Learning and Multi-Task Panoptic Reconstruction are also introduced recent years and it's been a trend that we construct models with Implicit Representations and Multi-Modal Fusion. TripoSR(Tochilkin et al. 2024) is an AI-modeling tool with these features.

3 Main Methodologies

The pipeline of this project consists of four stages and we use several open-source Github repositories and write a script to combine them together to reach the final effect. As a start, we have:

Sketch2scene Folder
I— flowty-realtime-lcm-canvas
I— Stable Diffusion WebUI
I— MIDI-3D
I— ...

3.1 Stage 1: Sketch to Image

The first thing we need to do is to generate an image from the input sketch.

We choose the project flowty-realtime-lcm-canvas which uses Latent Consistency Models(Luo et al. 2023). LCM LoRA is a technique that accelerates diffusion models by fine-tuning LoRA layers to mimic the multi-step denoising process of a teacher model in fewer steps. With such powerful models flowty-realtime-lcm-canvas could turn sketches into images in real-time.

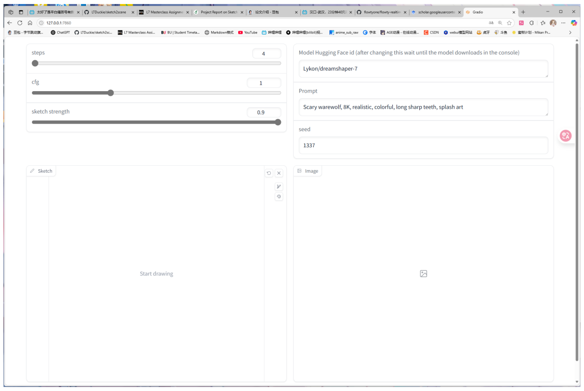
What we need to do is to set up the project under our parent folder Sketch2scene Folder. This project uses gradio==3.44.1 so in conda we build up a virtual environment for it and install necessary requirements.

```
conda create −n flowty python=3.12
```

And then we set up as the project readme says. To test if we have build it well, we should do

```
conda activate flowty
cd flowty−realtime−lcm−canvas
python ui.py
```

Open the web link in the powershell window, we could have the webui as shown in figure 1.



127.0.0.1:7860

steps 4

cfg 1

sketch strength 0.9

Model Hugging Face id (after changing this wait until the model downloads in the console)

Lykon/dreamshaper-7

Prompt

Scary warewolf, 8K, realistic, colorful, long sharp teeth, splash art

seed

1337

Figure 1: Enter Caption

3.2 Stage 2: Image Refinement

This step is to do image refinement so that our generated image in stage 1 could turn into a realistic style.

There are many image refinement tools currently, in this project, we could use the most typical one stable diffusion webui(Rombach et al. 2022).

Stable Diffusion's img2img converts an input image into a latent space, adds adjustable noise, and uses a UNet to predict noise patterns guided by text prompts. Through iterative denoising, it refines the latent representation, balancing fidelity to the original image with creative changes. The result is decoded into a new image, merging input structure with prompt-driven aesthetics.

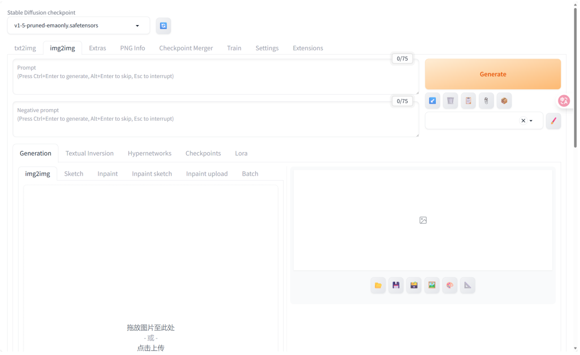
Also, set up the project under our parent folder Sketch2scene Folder. This project uses python==3.10 so in conda we build up another virtual environment for it.

```
conda create -n sdweb python=3.10
```

And then we set up as the project readme says. To test if we have build it well, we should do

```
conda activate sdweb
cd SD_webUI
python webui-user.py
```

Open the web link in the powershell window, we could have the webui as shown in figure 2.



Stable Diffusion checkpoint
v1-5-pruned-emaonly.safetensors

Figure 2: stable diffusion webui

3.3 Stage 3: Image Segmentation

Image Segmentation is to segment the image into different objects and output a corresponding image of segmentations so that Stage 4 could generate models in the scene separately.

Grounding DINO(Liu et al. 2024) is an open-set object detection model with text prompts that integrates the DINO (Transformer-based detection framework) with grounded pre-training.

This module is involved in the referred repositories in Stage 4. So they only need to be set up once.

3.4 Stage 4: Generate Models

With the initial refined image and the image of segmentation, we could now step into modeling stage to generate a scene.

We choose the project MIDI-3D(Huang et al. 2024). MIDI is a model for generating 3D scenes from a single image using multi-instance diffusion. It decomposes the input image into instance-level 2D representations, encodes each into a 3D latent space, and uses a diffusion process to generate 3D geometry, materials, and lighting for each instance, conditioned on camera parameters and scene context.

Set up as the project readme says. To test if we have build it well, we should do

```
cd MIDI-3D
python gradio_demo.py
```

Open the web link in the powershell window, we could have the webui as shown in figure 3.

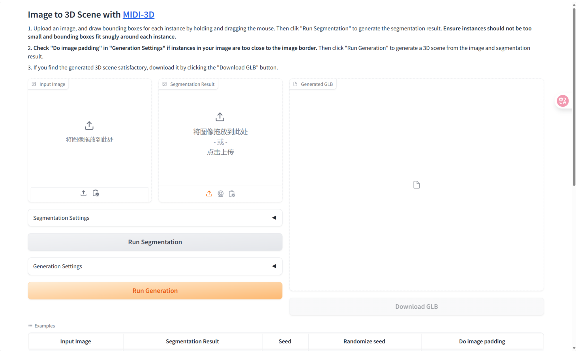


Image to 3D Scene with MIDI-3D

1. Upload an image, and draw bounding boxes for each instance by holding and dragging the mouse. Then clik "Run Segmentation" to generate the segmentation result. **Ensure instances should not be too small and bounding boxes fit snugly around each instance.**

2. **Check "Do image padding" in "Generation Settings" if instances in your image are too close to the image border.** Then click "Run Generation" to generate a 3D scene from the image and segmentation result.

3. If you find the generated 3D scene satisfactory, download it by clicking the "Download GLB" button.

Figure 3: MIDI-3D

Users could draw squares to assist the segmentation. Therefore, this process could be done manually.

3.5 Combine and Build in One

At the moment the pipeline is clear, let's create a notebook file in the Sketch2scene Folder.

Get User Input, including the sketch image and the text prompts.

```
sketch_path = ”sketch_sample_2.png”
Prompt = input('Input sketch tokens: ')
```

run the flowerty and get user input:

```
process = subprocess.Popen(
    'conda-run',
    '—n-flowty_env',
    'python-flowty/ui.py',
    text=True,
    shell=True)
```

use API of flowty:

```
r = sketch_to_image_client.predict(
            Prompt,
            sketch_path,
            4,          # 'steps'
            1,          # 'cfg'
            0.9,        # 'sketch strength'
            −1,         # 'seed'
            fn_index=0)
```

After each stage, the path of the generated image would be saved and the result would always be represented. Once the user is not satisfied with the result, it would be easy to change any of the parameters or prompts. Here is an example:

```
with open(”output.txt”, ”r”) as f:
    sketch_to_image_result = f.read().strip()
Image.open(sketch_to_image_result)
```

The process following would be similar. See sketch2scene.ipynb for details.

4 Experiments

The experiments of this project would correspondingly be divided into 4 stages. We will have five sketches samples, put them into stages and finally get the modeled scene.

4.1 Stage 0: prepare the sketches

In this stage, we draw five sketches as figure 4, 5, 6, 7 and 8. They are all sketched with Microsoft Paint except for the last one which is downloaded from the internet.

It is recommended that users' sketches have the nearly equivalent width and height so that it would be easier for Stage 1. Otherwise, self-padding of sketches are suggested.

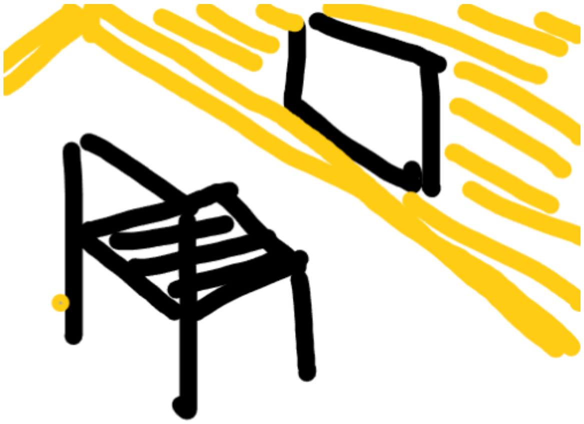


Figure 4: sketch 1

Prompts for figure 4 are: one chair, white floor, photo on the wall.

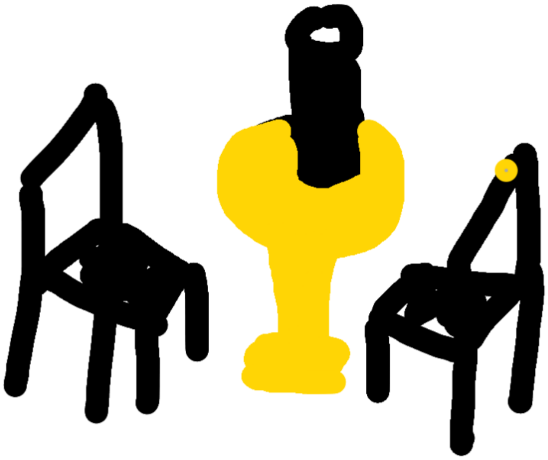


Figure 5: sketch 2

Prompts for figure 5 are: two chairs, one table, one basket.

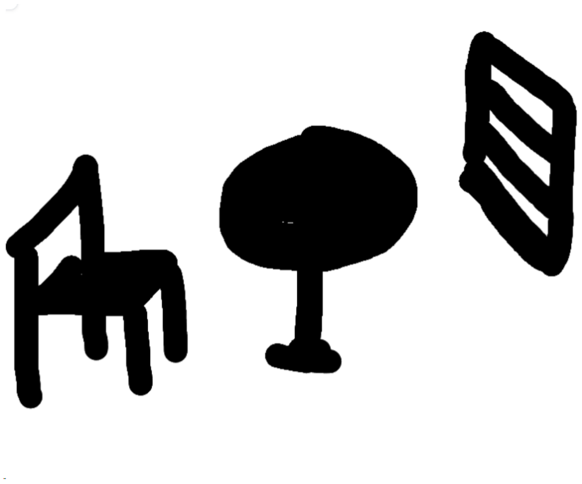


Figure 6: sketch 3

Prompts for figure 6 are: one chair, one table, one shelf.

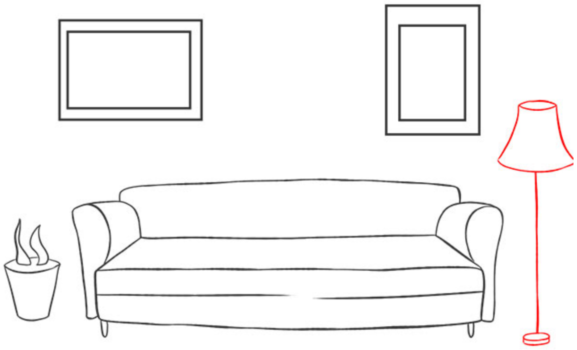


Figure 7: sketch 4

Prompts for figure 7 are: one sofa, photos on the wall, red lamp, one plant.

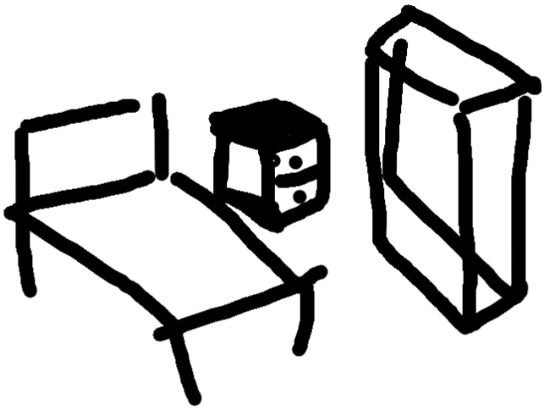


Figure 8: sketch 5

Prompts for figure 8 are: one cabinet, one bed, one closet.

4.2 Stage 1: Sketch to Image

Figure 9 demonstrates the result after Stage One. (Correspondingly Sketch 1 to 5 from left to right)

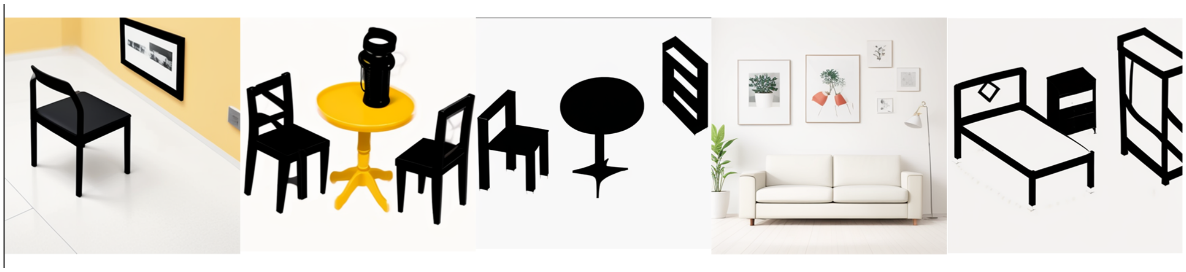


Figure 9: After Stage 1, results

4.3 Stage 2: Image Refinement

Figure 10 demonstrates the result after Stage Two. (Correspondingly Sketch 1 to 5 from left to right)

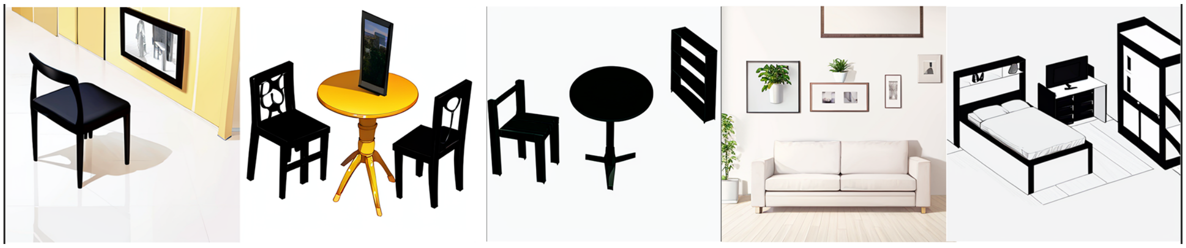


Figure 10: After Stage 2, results

4.4 Stage 3: Image Segmentation

Without any hint of selections of objects, the project could still generate the image segmented automatically(run in code). However, it is suggested that users manually(run in web) select objects so that the outcome could be more accurate.

4.4.1 Run in our code:

Figure 11 demonstrates the result after Auto Stage Three. (Correspondingly Sketch 1 to 5 from left to right)

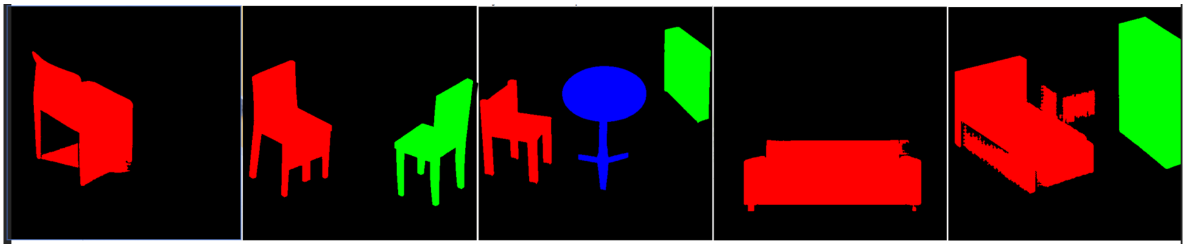


Figure 11: After Stage 3(automatically), results

4.4.2 Run in web with selection:

Figure 12 demonstrates the result after Manual Stage Three. (Correspondingly Sketch 1 to 5 from left to right)

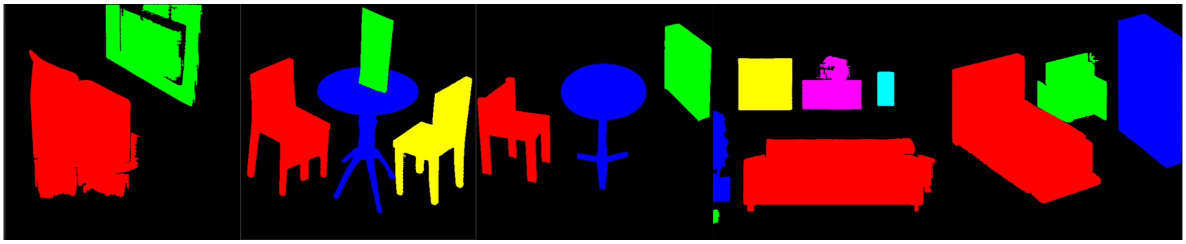


Figure 12: After Stage 3(manually), results

4.5 Stage 4: Generate Models

In stage three we may get two types of segmented images, which lead to different modelings. Figure 13 shows the result after automatic process, while figure 14 displays the other.

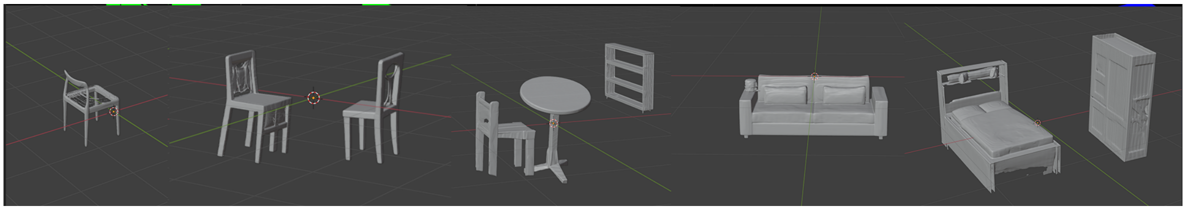


Figure 13: After Stage 4(automatically), results

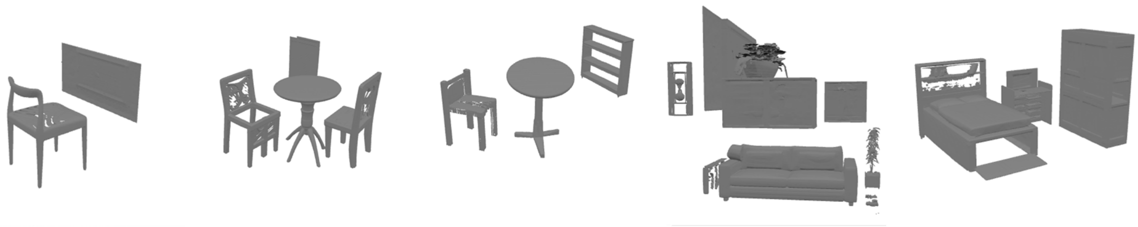


Figure 14: After Stage 4(manually), results

5 Critical Self - Evaluation

5.1 Experiment Results Analysis

Through experiments with five sets of sketches, the technical processes of each stage basically achieved the expected goals:

Sketch-to-Image (Stage 1): The flowty-realtime-lcm-canvas generates images in real time, performing well in simple scenes (e.g., single objects, as shown in Figure 4 and Figure 6), quickly restoring sketch structures and adding basic colors. However, in complex scenes with multiple interacting objects (e.g., Figure 5 and Figure 7), issues like disproportionate object sizes and blurred details emerge. For example, the relative dimensions of the chair and table in Figure 5 are obviously skewed. **Image Refinement (Stage 2):** Stable Diffusion WebUI effectively enhances image realism, particularly in material textures (e.g., wooden tabletops, fabric sofas) and lighting effects. However, it may generate content deviating from user expectations for details not clearly marked in sketches (e.g., wall decorations, plant shapes), requiring repeated adjustments via text prompts. **Image Segmentation (Stage 3):** Automatic segmentation (Grounding DINO) works stably in scenes with clear object boundaries (e.g., independent furniture, Figure 11) but struggles with overlapping objects (e.g., the sofa and photo wall in Figure 7), often causing segmentation errors. Manual segmentation via user bounding boxes improves accuracy (Figure 12) but reduces operational efficiency. **3D Model Generation (Stage 4):** MIDI-3D produces structurally complete 3D models for single-object scenes (e.g., the chair in Figure 4). However, multi-object scenes (e.g., the bedroom setup in Figure 8) suffer from unreasonable scene layouts, chaotic object occlusions, and poor material-lighting consistency.

5.2 Strengths

Innovative Technical Integration: Combining open-source tools like LCM, Stable Diffusion, Grounding DINO, and MIDI-3D to build a full pipeline from 2D sketches to 3D scenes, offering low-cost 3D content generation for non-professionals. **Interactive Flexibility:** Each stage supports parameter adjustment and result reuse, allowing users to optimize specific unsatisfactory steps (e.g., Stage 1's image style, Stage 4's model morphology) independently, reducing trial-and-error costs. **Real-Time Performance:** Stage 1 uses Latent Consistency Models for real-time sketch-to-image generation, significantly enhancing the user experience.

5.3 Weakness

Limited Complex Scene Handling: Current models struggle with small objects (e.g., buttons, decorations) and multi-object interactions (e.g., occlusions, proportional coordination), often producing geometric distortions or semantic inconsistencies. **High Manual Intervention Cost:** Stage 3's manual segmentation and Stage 4's scene layout adjustments rely on user input, with low automation, making it hard to scale for large-scale scene generation. **Cross-Stage Error Accumulation:** Flaws in earlier stages (e.g., blurry images in Stage 1) can propagate to later stages (e.g., reduced model accuracy in Stage 4), causing significant deviations between final results and original sketches. **Time-Consuming:** During Stage 4, the user have to wait for a long time for the scene to be completed.

5.4 Future Outlook

Model Optimization: Introduce multimodal fusion (e.g., joint text-image-3D training) to enhance semantic understanding of complex scenes. Leverage hierarchical generation capabilities of Diffusion models to improve small-object and detail accuracy. **Automation Upgrades:** Develop end-to-end adaptive workflows to minimize manual steps (e.g., automatic object interaction detection, intelligent scene layout optimization), and explore prompt-free or few-prompt generation modes. **Application Expansion:** Integrate the pipeline into architecture design, game development, etc., supporting custom material libraries and lighting presets to boost the practicality and artistic expression of 3D scenes.

References

Chen, Wengling and James Hays (2018). "Sketchygan: Towards diverse and realistic sketch to image synthesis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9416–9425.

Gatys, Leon A, Alexander S Ecker, and Matthias Bethge (2016). "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423.

Ghosh, Arnab et al. (2019). "Interactive sketch & fill: Multiclass sketch-to-image translation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1171–1180.

Huang, Zehuan et al. (2024). "MIDI: Multi-Instance Diffusion for Single Image to 3D Scene Generation". In: *arXiv preprint arXiv:2412.03558*.

Jackson, Aaron S et al. (2017). "Large pose 3D face reconstruction from a single image via direct volumetric CNN regression". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1031–1039.

Lei, Mingkun et al. (2024). "StyleStudio: Text-Driven Style Transfer with Selective Control of Style Elements". In: *arXiv preprint arXiv:2412.08503*.

Liu, Shilong et al. (2024). "Grounding dino: Marrying dino with grounded pre-training for open-set object detection". In: *European Conference on Computer Vision*. Springer, pp. 38–55.

Luo, Simian et al. (2023). "Latent consistency models: Synthesizing high-resolution images with few-step inference". In: *arXiv preprint arXiv:2310.04378*.

Rombach, Robin et al. (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695.

Tochilkin, Dmitry et al. (2024). "Triposr: Fast 3d object reconstruction from a single image". In: *arXiv preprint arXiv:2403.02151*.

Wang, Qiang et al. (2023). "Diffsketching: Sketch control image synthesis with diffusion models". In: *arXiv preprint arXiv:2305.18812*.

Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala (2023). "Adding conditional control to text-to-image diffusion models". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847.