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.

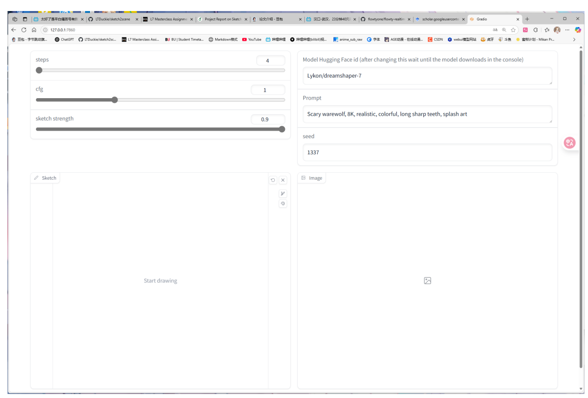

Figure 1: Enter Caption

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.

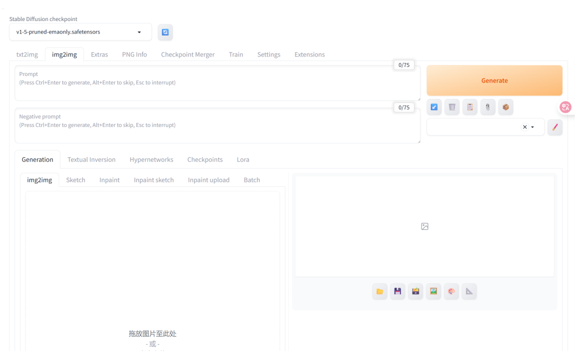

Figure 2: stable diffusion webui

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.

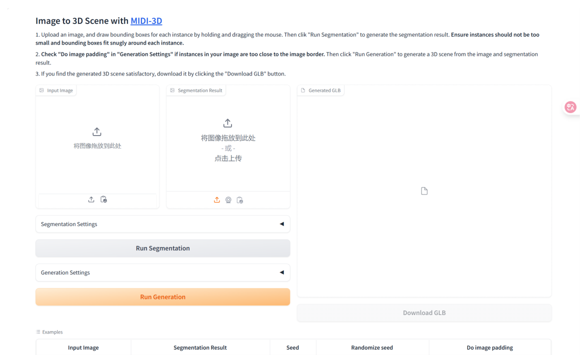

Figure 3: MIDI-3D

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.

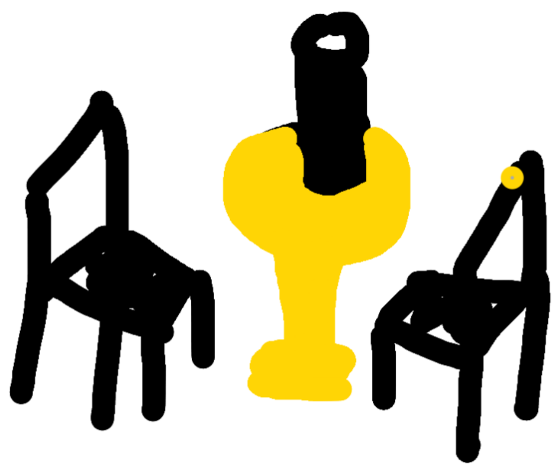

Figure 5: sketch 2

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

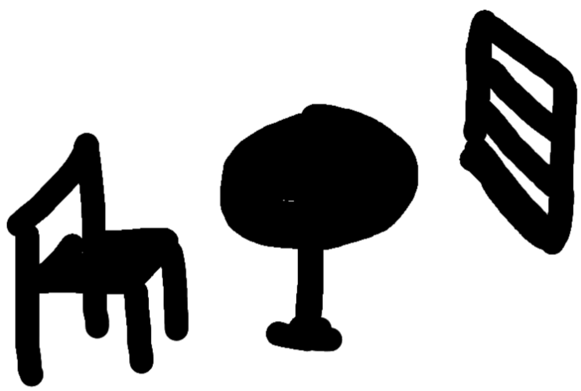

Figure 6: sketch 3

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

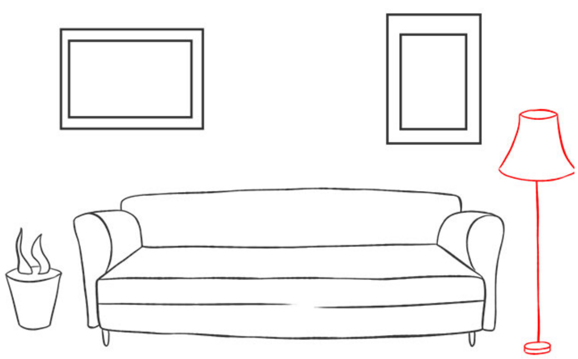

Figure 7: sketch 4

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

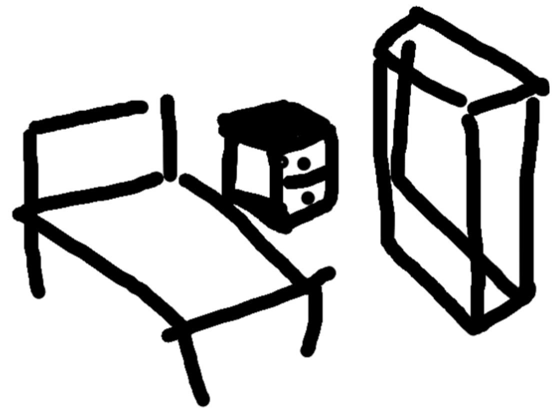

Figure 8: sketch 5

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)

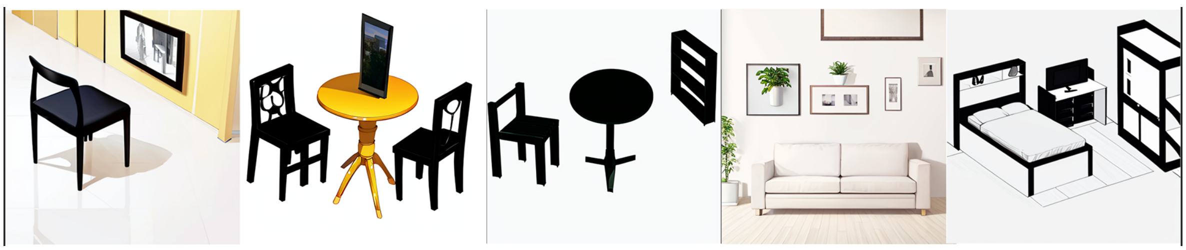

Figure 10: After Stage 2, results

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)

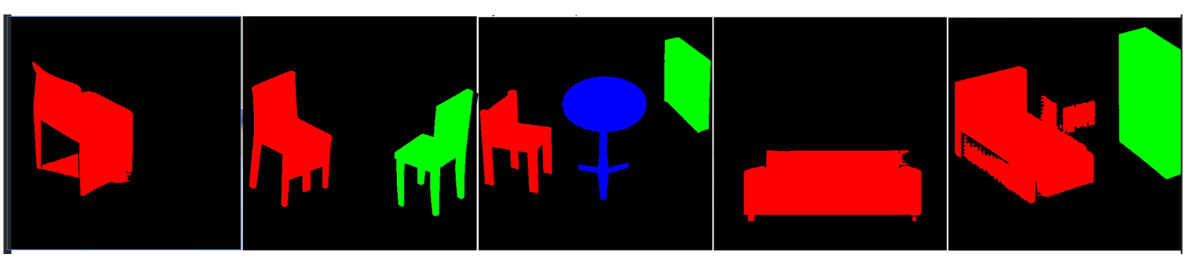

Figure 11: After Stage 3(automatically), results

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)

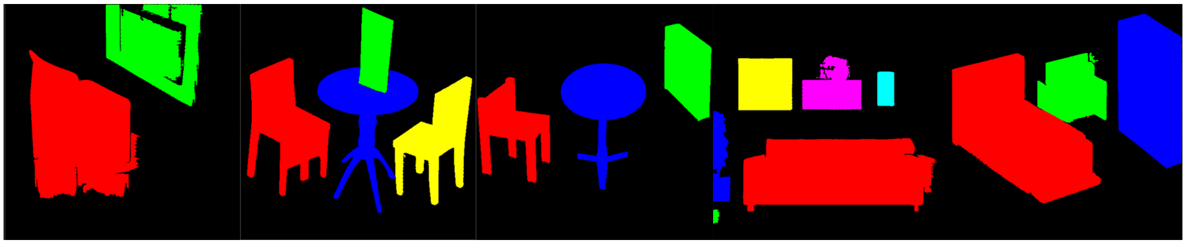

Figure 12: After Stage 3(manually), results

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.

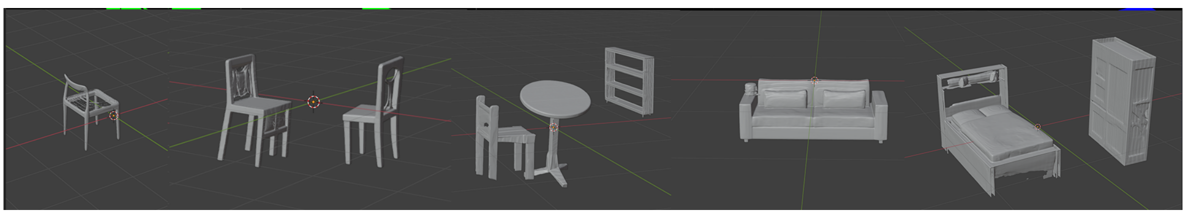

Figure 13: After Stage 4(automatically), results

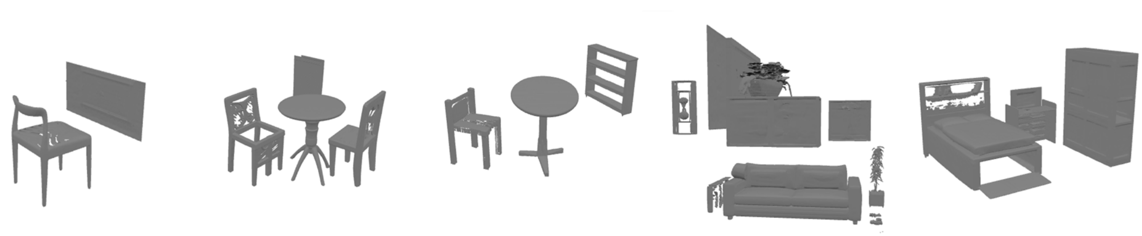

Figure 14: After Stage 4(manually), results

5 Critical Self - Evaluation

5.1 Experiment Results Analysis

We evaluated the full pipeline with five test sketches. Overall, each stage of the pipeline achieved its intended function, but several limitations were observed:

Sketch-to-Image (Stage 1): Our real-time LCM-based model quickly generates an initial colored image from the sketch. It performs well on simple, single-object scenes (e.g., an isolated chair), effectively recovering object structure and adding basic colors. However, in complex scenes with multiple interacting objects (e.g., a living room setup), we observed irregularities: object proportions became distorted and fine details were blurred. For instance, in multi-item sketches the relative sizes of furniture (such as chairs versus tables) were inconsistent. **Image Refinement (Stage 2):** The Stable Diffusion-based refinement significantly improves photorealism and texture details. Wood grains, fabric textures, and lighting in the scene become more realistic. Nonetheless, the refinement can introduce elements not intended by the sketch. Details not indicated by the original outline (such as wall decorations or specific plant shapes) may be hallucinated or misinterpreted, requiring the user to iteratively adjust text prompts to correct these. **Image Segmentation (Stage 3):** Automatic segmentation using Grounding DINO reliably identifies and segments objects when they have clear, separate boundaries. In scenes where objects are disjoint (e.g., a standalone lamp or chair), segmentation is generally accurate. However, in cluttered scenes with overlapping objects (e.g., a sofa partially covering a picture frame), the automatic segmentation often fails or mixes objects. To address this, we allow manual intervention: the user can draw bounding boxes to refine the segments. This manual segmentation (as demonstrated in our workflow's Stage 3 interface) improves accuracy but requires extra user effort and time. **3D Model Generation (Stage 4):** The MIDI-3D model generation produces complete 3D meshes for single-object scenes. For example, a sketch of a chair yields a structurally sound 3D chair model with reasonable proportions. In multi-object scenes, however, the generated scene often has flawed layouts. Objects may overlap unnaturally or appear at conflicting depths, and their materials or lighting may not be consistent. For instance, the bedroom scene with a bed, table, and lamps results in an implausible arrangement and inconsistent shadowing. Thus, complex multi-object scene reconstruction remains unreliable.

5.2 Strengths

Innovative Technical Integration: The pipeline combines several state-of-the-art open-source tools (LCM(Luo et al. 2023) for real-time sketch generation, Stable Diffusion(Rombach et al. 2022) for style refinement, Grounding DINO(Liu et al. 2024) for segmentation, and MIDI-3D(Huang et al. 2024) for mesh creation) into a cohesive end-to-end system. This integration offers a low-cost framework for transforming 2D sketches into 3D scenes, potentially lowering the barrier for non-expert users to create 3D content.

Interactive Flexibility: Each stage of the pipeline supports parameter adjustment and manual intervention. For example, after Stage 1 the user can re-style the image, and in Stage 3 the user can provide additional segmentation hints. This modularity allows users to iteratively improve specific parts of the output without rerunning the entire pipeline, reducing trial-and-error and providing more creative control.

Real-Time Performance: The use of a Latent Consistency Model enables real-time sketch-to-image synthesis. This significantly enhances the user experience, as users receive immediate visual feedback when drawing sketches. A responsive Stage 1 encourages exploration and quick iteration on the initial image, a usability advantage over slower models.

5.3 Weakness

Limited Complex Scene Handling: The system struggles with scenes containing small details and multiple interacting objects. Sketch inputs with occlusions or many objects often result in distorted geometry or missing elements. The model has difficulty encoding multiple instances and their spatial relationships, leading to semantic inconsistencies (e.g., objects intersecting improperly).

High Manual Intervention Cost: Several stages rely heavily on user input. In Stage 3, manual bounding boxes are often needed to correct segmentation errors. Stage 4 requires the user to manually adjust scene layout or remap textures if the generated 3D scene is unsatisfactory. This high dependency on manual steps makes the pipeline time-consuming and less scalable for generating many scenes.

Cross-Stage Error Accumulation: Errors introduced in early stages can accumulate. For example, if Stage 1 produces a blurry or inaccurate image, subsequent stages (refinement, segmentation, modeling) tend to exacerbate these flaws, as they work with imperfect inputs. This cascading effect can significantly deviate the final 3D scene from the user's original sketch intent.

Time-Consuming: The 3D generation in Stage

4 (using MIDI-3D) is computationally intensive. Users must wait several minutes for a single scene to be generated and optimized. This slow turnaround is a practical limitation for iterative design or for use cases requiring quick previews.

5.4 Future Outlook

Model Optimization: Future improvements could focus on joint multi-modal learning. For example, integrating a unified model that is trained on text, image, and 3D data simultaneously may enhance the system's semantic understanding of complex scenes. Hierarchical or multi-resolution diffusion techniques could also be applied to better capture small details and improve fidelity on fine-grained objects.

Automation Upgrades: Automating more pipeline stages would improve usability. For instance, introducing automatic object interaction reasoning (so the system infers plausible spatial arrangements) and intelligent layout optimization could reduce the need for manual corrections. Further development of end-to-end workflows (possibly with "few-shot" or prompt-free generation modes) could streamline the process, making it more accessible to non-technical users.

Application Expansion: The pipeline could be adapted and tested in application domains such as architectural design, interior planning, or game environment prototyping. Integrating scene presets (e.g., predefined furniture arrangements or material libraries) and enabling export to common 3D formats would increase practicality. Expanding the system to handle user-provided photos or semantic maps as input (in addition to sketches) could also broaden its utility in professional content-creation pipelines.

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.