Text-guided editing of 3D models with generative AI

MSc Geomatics Graduation Plan

Yingxin Feng
1st supervisor: Nail Ibrahimli
2nd supervisor: Ken Arroyo Ohori

January, 2024

1 Introduction

There has been a fast-growing demand for 3D models in recent years and the increasingly easily accessible camera-captured real-world objects are also widely used. Both explicit and implicit methods can be used to represent 3D models and neural implicit representations based on multi-view images are a promising alternative to traditional methods due to the high quality and potential in reconstructing complex objects (P. Wang et al., 2021). High-fidelity 3D model representation from multi-view images is also an interesting topic for photogrammetry.

Apart from the creation or reconstruction of the 3D models, the editing of the handcraft or real-world 3D models is inevitable to fully meet the requirements of the users. However, to perform such editing, one needs to receive basic training and the process is still relatively time-consuming even for professionals (Haque et al., 2023). Therefore, a (semi-)automotive tool to edit the 3D model will help improve the efficiency of people working in related fields. For the tool, easy instruction will be an advantage and thus text instruction will be a good choice. The text-guided generative AI based edit tool requires far less manual work. Such a tool will be useful in different scenarios. One example is in the field of architectural design, where text-instructed edits can help designers visualize their improved ideas on the initial models more easily, especially in the concept design stage when various adaptations are needed. Another scenario is in the game or movie fields where realistic 3D scenes are needed and the edit tool can help to convert the captured real-world objects to meet the demands, which can add value to the existing huge amount of geo-data in the future.

Generative AI models have been developed rapidly recently and have proved to be applicable in various domains, among which 2D images and 3D models are important branches. There are also various instruction types for such models, including sketches, edges, texts and so on, and texts receive high attention due to the easy-to-use nature, and the huge amount of available data(Kamata et al., 2023). Many existing research focus on text-guided 3D content generation, as well as 2D image generation and editing, and have achieved satisfying results and attracted attention from beyond the academic community. On the contrary, in the field of 3D model editing, there is relatively less research and improvements are needed for a more easy-to-use tool to gain consistent and high-quality results.

Therefore, the thesis aims to contribute to advanced 3D model editing with generative AI methods. A novel pipeline will be developed to achieve text-guided multi-view consistent editing of geometry and texture of 3D and higher fidelity and efficiency will also be directions to be explored. Multiple common objects and text instructions will be tested and comparisons will be made with possible alternative elements and related existing models.

2 Problem statement

The main research question of the thesis is:

- How to perform text-guided editing of the geometry and texture of 3D models with generative AI methods?

To solve it, the following sub-questions need to be answered:

1. What method should be chosen to represent the high-fidelity 3D model?
2. What generative AI methods should be used?
3. How to improve multi-view consistency for the final result?
4. What are the limits of the final proposed text-guided editing pipeline?

The scope of the thesis is also defined. As for the input, the 3D models are common 3D objects, including ordinary buildings. They are initially represented by multi-view images with camera parameters. The text prompts are common and straightforward instructions, like simple modification of a certain part (e.g. make it a flat roof building), and overall style changes (e.g. make it Minecraft style). The output edited 3D models are represented by the chosen 3D representation method, and can be further processed to generate corresponding multi-view images and mesh.

3 Related work

3.1 3D representation

There are two main types of 3D representation methods, namely explicit and implicit ones. The typical explicit methods include point cloud, voxel and mesh. They can naturally keep the fine details of original objects while their explicit nature usually makes it more difficult to fully incorporate them into the neural networks pipeline, though there are some successful attempts, such as the mesh-based DMTet method that represents 3D models with deformable tetrahedral grid and differentiable Marching Tetrahedral layer(Shen et al., 2021). It is also used in 3D content generation model Fantasia3D(R. Chen et al., 2023).

Implicit ways can represent 3D models with various types of functions or even neural networks. Neural networks show a promising future for their increasing quality and ability to directly reconstruct 3D models from multi-view images, which lowers the requirements of capturing real-world 3D models. There are two major ideas for implicit methods, namely surface representation and volumetric rendering. The Signed Distance Function (SDF) that can define the signed distances between points and the surface is a base for the former one. It is good at extracting high-quality surfaces, and one example is the Implicit Differentiable Renderer (IDR) (Yariv et al., 2020). The latter one performs better at dealing with abrupt depth changes and Nerual Radiance Fields (NeRF) is a representative, which utilizes volume rendering and volume representation (Mildenhall et al., 2020).

Neural Implicit Surfaces (NeuS) applies the volume rendering approach to learn SDF for surface representation and is a successful attempt to combine the advantages of surface and volume representations. Similar to Nerf, a neural network is constructed to represent each individual 3D model and the colour of each pixel of the multi-view images is used as ground truth to train the network. The rendered pixel colour from the network is calculated by accumulating the colour of sampled points along the ray of the corresponding camera view at that pixel. The adaptation is that instead of merely using the volumetric density as weight, the signed distance and an unbiased and occlusion-aware weight function are combined as the new weight. In this way, the SDF is integrated into the neural network. Neus achieves better results compared to the original NeRF (also called Vanilla NeRF) (Mildenhall et al., 2020) with more detailed and less noisy surfaces at the cost of longer computation time(P. Wang et al., 2021). To improve the efficiency, multi-resolution hash encoding, fully fused networks, and occupancy grid pruning and rendering techniques are used for acceleration(Müller et al., 2022).

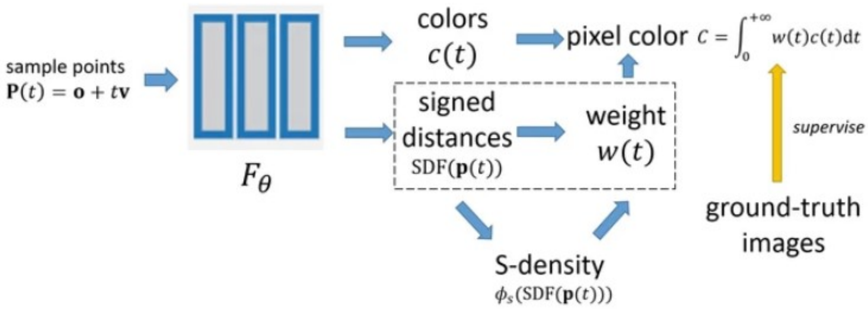

sample points
$\mathbf{P}(t) = \mathbf{o} + t\mathbf{v}$
F_θ
colors
$c(t)$
pixel color $C = \int_0^{+\infty} w(t)c(t)dt$
signed distances
$\mathrm{SDF}(\mathbf{p}(t))$
weight
$w(t)$
supervise
ground-truth images
S-density
$\phi_s(\mathrm{SDF}(\mathbf{p}(t)))$

Figure 1: NeuS model (P. Wang et al., 2021): utilize both volume rendering and surface representation

3.2 General generative AI methods

Generative adversarial networks (GAN), Variational Autoencoders (VAE) and Diffusion Model are typical generative AI methods used in 2D and 3D content generation and edit (C. Li et al., 2023). The Diffusion Model outperforms the others in general. It's easier to scale and train and can generate more diverse and higher-quality results (Dhariwal & Nichol, 2021). As shown in the figure below, the general idea of the Diffusion Model is to reverse the process of transforming an image to noise by adding Gaussian noise in each time step, and conditions can be added to generate target results (Po et al., 2023). In the conditioned Diffusion Model, which is used in most actual cases, a neural network is trained to minimise the difference between the predicted noise and the actual noise added (sampled noise) using images with conditions (e.g. text descriptions of images) as the training dataset. It can subtract the noise added at the given time step and thus output the desired image with conditions, time step and noise (or noisy image) as input.

There are relatively outstanding image generation and edit models that utilize the Diffusion Model idea. They will be used as a base for the thesis and below is a brief introduction to them. The Stable Diffusion Model is a popular text-to-image generation model with text-prompt as the condition. It uses Latent Space, which largely reduces the trainable parameter size, to represent the original pixel space by encoding and decoding at the beginning and the end respectively (Rombach et al., 2021). Pre-trained CLIP is also used to connect the text and the image (Radford et al., 2021). DeepFloyd IF is another popular text-to-image generation model, using pixel-based and triple-cascaded techniques (Saharia et al., 2022). InstructPix2Pix is a text-guided image editing model. It's based on Stable Diffusion and uses both the original image and the text prompt as conditions. Its loss function consists of three parts, the classifier-free part (to maintain diversity), the image-conditioned part and the text-conditioned part. In inference time, the weights (scales) of the two instructions can be adjusted to control the similarity with the original image and the consistency with the text(Brooks et al., 2022).

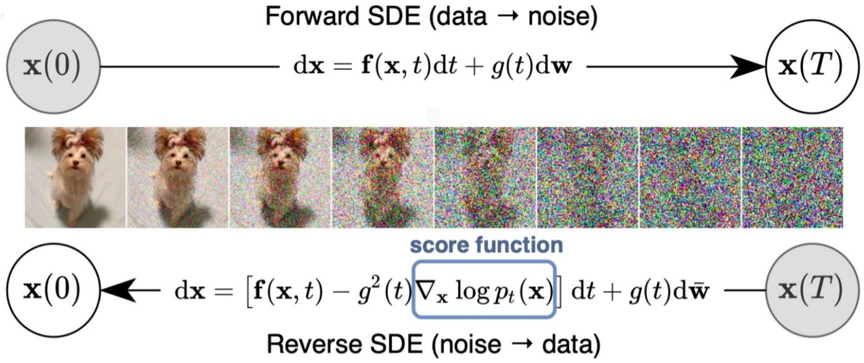

Forward SDE (data → noise)
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$
$\mathbf{x}(T)$
score function
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$
$\mathbf{x}(T)$
Reverse SDE (noise → data)

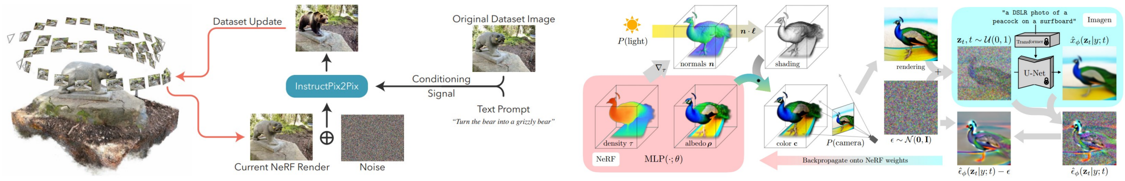
Figure 2: General process of the Diffusion Model (Po et al., 2023): add Gaussian noise at each time step and then reverse

3.3 General 3D generation and edit pipelines

To train a high-quality generative AI model, large datasets are needed. However, compared to 2D images, 3D model datasets are relatively scarce (Poole et al., 2022). Additionally, the 3D generative model also contains more parameters and greatly increases the computation cost. Therefore, it's a natural idea to utilize 2D text-guided image generative models to guide 3D content generation and editing.

A straightforward idea is to edit the base multi-view images dataset directly. Instruct-nerf2nerf applies the iterative dataset strategy. It represents the 3D model with NeRF and chooses InstructPix2Pix to edit the rendered images with text-prompt and original images as conditions at each iteration (Haque et al., 2023).

Another more elegant way is to backpropagate the image loss to the 3D implicit neural model to guide 3D content generation or editing. Dreamfusion focuses on the text-guided generation of 3D content and puts forward the concept of Score Distillation Sampling (SDS) loss, which proves that the image loss from the frozen 2D Diffusion Model can be backpropagated to the 3D model. The loss is calculated by multiplying the difference between the sampled noise (from the rendered image by the 3D model) and the predicted noise by the 2D Diffusion Model with a designed weight function. Using SDS loss needs to maintain the gradient flow, meaning that both the 2D Diffusion Model and the 3D representation should be differentiable (Poole et al., 2022). Instruct 3D-to-3D also uses the SDS loss to guide the edit of NeRF model (Kamata et al., 2023). ProlificDreamer argues that Variational Score Distillation (VSD) loss, which has a similar format as SDS loss but adds additional camera parameters to the condition embeddings of the network, is a more general version of SDS loss and has better performance (Z. Wang et al., 2023).


Dataset Update
Original Dataset Image
InstructPix2Pix
Conditioning Signal
Text Prompt
"Turn the bear into a grizzly bear"
Current NeRF Render
Noise
P(light)
normals $\boldsymbol{n}$
$\boldsymbol{n} \cdot \boldsymbol{\ell}$
shading
∇_τ
rendering
"a DSLR photo of a peacock on a surfboard"
Imagen
$\mathbf{z}_t, t \sim \mathcal{U}(0,1)$
Transformer
U-Net
$\hat{x}_\phi(\mathbf{z}_t|y;t)$
density τ
albedo $\boldsymbol{\rho}$
NeRF
MLP$(\cdot;\theta)$
color $\mathbf{c}$
P(camera)
$\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
Backpropagate onto NeRF weights
$\hat{\epsilon}_\phi(\mathbf{z}_t|y;t) - \epsilon$
$\hat{\epsilon}_\phi(\mathbf{z}_t|y;t)$

(a) Instruct-nerf2nerf (Haque et al., 2023): iterative dataset update

(b) Dreamfusion (Poole et al., 2022): backpropagate 2D SDS loss to 3D model

Figure 3: Pipelines of representative 3D models edit and generation

3.4 Techniques for models improvement

Generative AI for 3D content is a fast-growing field and many recent works put forward new models and techniques to improve the current results (C. Li et al., 2023). How to ensure multi-view consistency is an important topic and possible techniques include changing the loss function to regularize different views (Armandpour et al., 2023; Hong et al., 2023), adding additional pre-trained modules to incorporate camera parameters (W. Li et al., 2023), directly modifying 2D Diffusion model to add camera parameters (Shi et al., 2023) or additional loss terms to gain multi-view consistent images (He et al., 2023), and generating a more consistent edited images dataset with multiple pre-processing steps (Dong & Wang, 2023; Fang et al., 2023).

How to improve computation efficiency is another topic and researches show that updating the whole multi-view images dataset for 3D neural representation with the adjusted 2D Diffusion Model (Song et al., 2023), editing 3D model in latent space (M. Chen et al., 2023), and using different 3D representations in training stages (first low-fidelity and then high-fidelity) (Z. Wang et al., 2023) are possible techniques.

3.5 Summary

In the above literature review, the answers to the research sub-questions of the thesis are explored. To sum up, NeRF is a widely used 3D representation for the current models and thus the models inherit the limitations of NeRF, which has difficulties in generating high-quality surfaces. NeuS, which can overcome the problem (to some extent), is an ideal choice for developing the new pipeline currently. It's also differentiable, making it easier to fully incorporate into the pipeline.

Diffusion Model has been the dominant generative AI research direction very recently(Po et al., 2023). There are also already several relatively high-quality and popular open-source pre-trained models. Due to realistic considerations, 2D text-guided image Diffusion Models are the usual choices to guide the convergence of 3D content. Thus, choosing Diffusion-based methods as the base and lifting the 2D generative model to 3D with different tricks is a reasonable direction to explore. Additionally, there is also more research about 3D generation, which also achieves more promising results, than editing, adding to the high potential and importance of developing such a new pipeline for 3D editing.

Multi-view inconsistency is a common and complex issue in current models, which strongly influences the final result and attracts high attention. As for efficiency, many attempts are made, though some are at the cost of lower quality (M. Chen et al., 2023). There is also space for improvement for many models to be useful in actual scenarios regarding fidelity, diversity, consistency, and efficiency.

4 Methodology

4.1 Overview

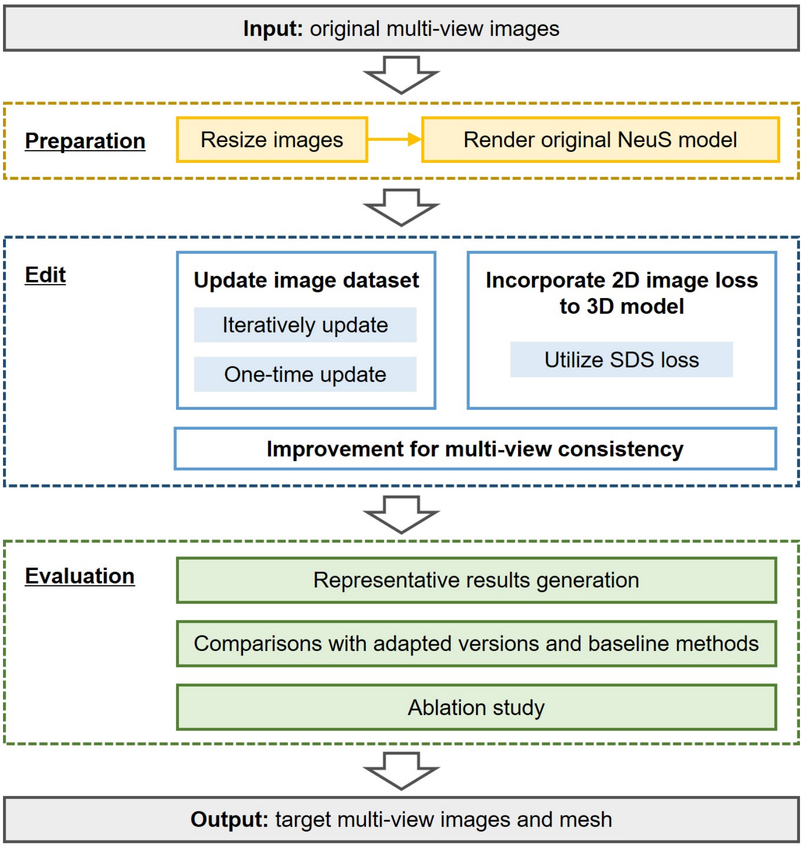

Input: original multi-view images
Preparation
Resize images
Render original NeuS model
Edit
Update image dataset
Iteratively update
One-time update
Incorporate 2D image loss to 3D model
Utilize SDS loss
Improvement for multi-view consistency
Evaluation
Representative results generation
Comparisons with adapted versions and baseline methods
Ablation study
Output: target multi-view images and mesh

Figure 4: Flowchart of preparation, edit and evaluation: two different base methods and corresponding improvement techniques are explored for editing.

As suggested by the related work section, to achieve the goal of text-guided 3D model editing, a 2D generative AI model based pipeline needs to be developed. Certain preparations are needed to generate the proper input for the pipeline and evaluations will be made to validate the pipeline. The flowchart of the process is as below.

4.2 Preparation

A proper 3D representation needs to be selected. Based on the literature review, NeuS is chosen for its potential to represent high-fidelity 3D models and its differentiable nature. Thus, the initial input is multi-view images with camera parameters. To fit the 2D generative model to increase efficiency and quality, the images from the dataset also need to be resized accordingly. Then an initial NeuS model is trained to represent the original 3D model.

4.3 Edit

As the thesis aims to explore different ideas to construct the final edit pipeline, multiple methods will be discussed here and experimented with. The current 2D text-guided image editing pre-trained model chooses the Diffusion Model based ones due to its outstanding performance compared to other types and a popular one is the InstructPix2Pix model.

There are two major directions to explore, namely updating the image dataset and incorporating 2D image loss into the 3D model. For the dataset update direction, both the iterate update and the one-time update can be experimented with. In the one-time update, multi-view correspondence regularization can be introduced as an addition loss to better ensure multi-view consistency (Song et al., 2023). The loss aims to minimize the distance of corresponding points across different views. Besides, only editing key views and for other views applying mixing-up, which mixes original and edited images with projection, blending, which first edit conditioned with original images and then with mixed images, and averaging multiple results tricks(Dong & Wang, 2023) is also a possible trail for consistency. When it comes to the second direction, the SDS loss can be used to guide the editing of the 3D model. VSD loss can also be tested as an alternative (Z. Wang et al., 2023).

Some general techniques can also be used to improve the multi-view consistency. Firstly, the loss calculation methods can be updated, especially for objects with clear front and back views like the toy bear to avoid the Janus problem (multiple faces in the 3D model with one only figure). Score debiasing and prompt debiasing (Hong et al., 2023), as well as negative text prompts (Armandpour et al., 2023), can be added. Secondly, image editing can be focused more on key views. Furthermore, if time permitted (as training a new Diffusion Model is highly time-consuming and requires more computation ability), camera parameters can be incorporated into the neural network by adding additional pre-trained modules (W. Li et al., 2023) or modifying existing 2D edited diffusion models (Shi et al., 2023). More details about the edit methods will be discussed in the preliminary result section.

4.4 Evaluation

To fully evaluate the effect of the final editing pipeline, a series of representative results that include different types of objects and text prompts need to be generated. Comparisons with adapted versions of the chosen pipeline that modify certain elements, and existing open-source baseline methods (e.g. InstructNeRF2NeRF) will also be made. Additionally, the ablation study, which generates the typical failure cases, will be conducted to test the limitations of the pipeline.

5 Data and tools

5.1 Data

The data required for this thesis is 3D model datasets which contain common objects. The 3D model should be represented by multi-view images with camera parameters. In addition, the ideal datasets need to be popular in the 3D generative AI community to allow easier comparison between chosen pipelines and other baseline methods. Therefore, the major data sets selected are the DTU MVS Dataset and NeRF-Synthetic Dataset. The first one is available *here* and a prepossessed version for 3D model generation can be found *here*. The second one is available *here*.

5.2 Tools

The programming language for this thesis is Python as it is one of the most popular languages in machine learning and many related open-source packages are available. The major packages for 3D representation and generative AI include Pytorch-lightning, Threestudio, Diffusers and Transformers (from Hugging Face). For the visualization of the final mesh, the open-source software Meshlab is used.

6 Preliminary result

6.1 Experimental pipeline

To test the proposed edit methods, preliminary experiments are conducted. The first chosen type of editing method is iterative dataset update as it's relatively easy to implement. The image resolution is resized to 256×256 to fit in the initial setting of the 2D edit model, as well as to improve testing efficiency. The detailed steps for one edit iteration are as below:

1. Randomly select one view and render the image from the current target Neus model.
2. Use the pre-trained InstructPix2Pix Diffusion Model to edit the rendered images with text prompt and the corresponding images from the original dataset as conditions.
3. Update the target dataset with the edited image.
4. Train the target Neus model for certain steps with the updated target dataset.

Two adaptations to the original pipeline are also tested. The former one is to change the image editing pre-trained model to InstructEdit. It claims to achieve better results compared to InstructPix2Pix, with the utilization of a language processor to pre-process the prompt, a segmenter to identify the target editing part and a mask-based image editor from DiffEdit. Its main idea is to segment and edit accordingly the parts that need to be changed to preserve the original information of the other parts(Q. Wang et al., 2023). The latter one is adding a procedure to the pipeline: update the whole multi-view images dataset with both the edited image and the rendered images of all the other views from the current Neus model. It aims to relieve the possible divergence problem caused by inconsistent edited images from different views as with such a method the images in the current dataset always come from the same 3D model and tend to be more consistent.

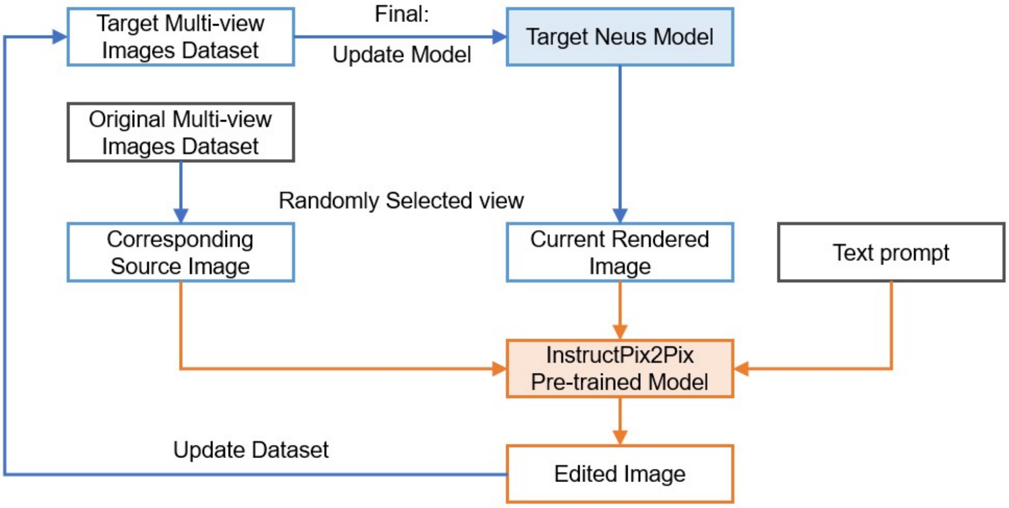
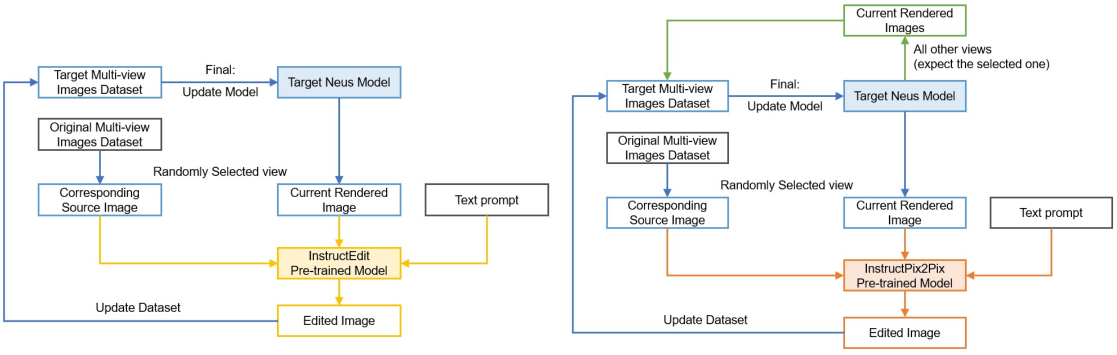

Target Multi-view Images Dataset
Final:
Update Model
Target Neus Model
Original Multi-view Images Dataset
Randomly Selected view
Corresponding Source Image
Current Rendered Image
Text prompt
InstructPix2Pix Pre-trained Model
Update Dataset
Edited Image

Figure 5: Experimental pipeline: use iterative dataset update method and pre-trained InstructPix2Pix for image editing


Target Multi-view Images Dataset
Final:
Update Model
Target Neus Model
Original Multi-view Images Dataset
Randomly Selected view
Corresponding Source Image
Current Rendered Image
Text prompt
InstructEdit Pre-trained Model
Update Dataset
Edited Image
Current Rendered Images
All other views (expect the selected one)
Target Multi-view Images Dataset
Final:
Update Model
Target Neus Model
Original Multi-view Images Dataset
Randomly Selected view
Corresponding Source Image
Current Rendered Image
Text prompt
InstructPix2Pix Pre-trained Model
Update Dataset
Edited Image

(a) Change the image editing pre-trained model

(b) Update the whole multi-view images dataset in one iteration

Figure 6: Adaptations to the experimental pipeline

6.2 Result

Experiments are made on several representative objects with the pre-tuned text prompt, text and image guidance scales for InstructPix2Pix to ensure that the initial editing of images is roughly desired, both preserving original details and changing towards the text-prompt direction. To balance efficiency and edited effect, 25 diffusion steps are performed for the image-edited model and the target Neus model is retrained 100 steps on the updated dataset in each iteration. The pipeline doesn't meet the expectation and the 3D model can't reach convergence in current experiments.

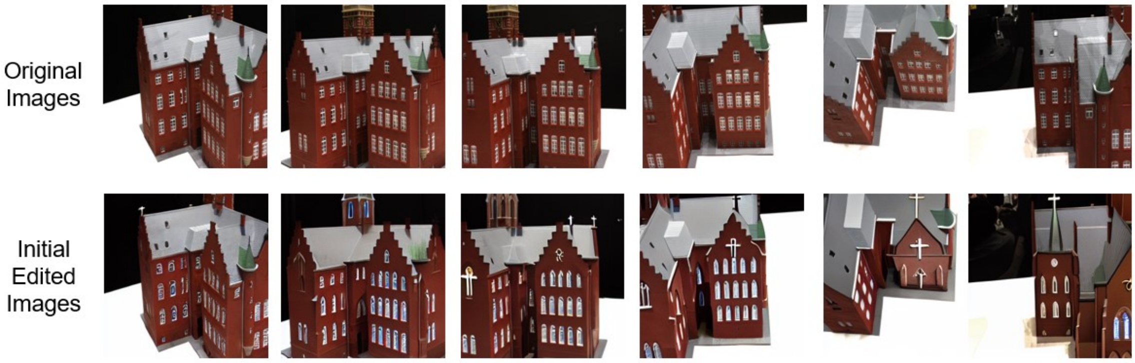


Figure 7: Original images and edited images during the first round (49 iterations) of editing from represented views: all views change towards the instructed direction in general but the multi-view consistency problem exists, and normal views editing outperforms acute views. [Number of multi-view images in the original dataset: 49; Text prompt: make it a church; Text guidance scale: 10; Image guidance scale: 1.5]

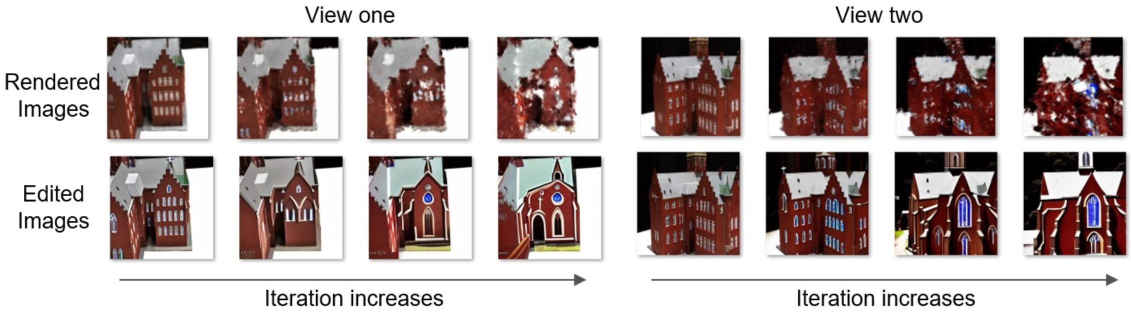


Figure 8: Represented rendered images and edited images during the training process: rendered images become blurry and edited images differ more from the original ones and the desired direction.

There are several problems existing in the current results. The most severe one is that the edited images vary a lot across different views. Such inconsistency is so severe in the tested examples that it's difficult for the target Neus model to converge and the newly rendered images become blurred. This further leads to undesired edited images, which differ a lot from the original image. Such a trend seems to show little possibility for the Neus model to converge after certain iterations as the updated dataset becomes further away from the desired direction, and thus the training is stopped later on. The second one is that it inevitably inherits the limits of InstructPix2Pix, meaning that it's impossible to make big geometrical changes and the prompt and guidance scales need to be carefully tuned beforehand. Besides, the normal views (like the parallel front view) editing can preserve more original details and in general performs much better in the tested examples than the acute angles, which may generate strange results at the very beginning. The last one is that the pipeline is not very efficient and one iteration costs more than half a minute on a single RTX 3090 GPU (memory size of 24GB).

As for the adaptations, the results still fail to meet expectations in the tested examples. For the first adaptation that changes the 2D diffusion model, the edited results for the whole multi-view images dataset generally perform worse than the InsturctPix2Pix model. The possible reason is that the segmenter sometimes fails to identify the correct part, which causes failure in the following edit. Additionally, larger GPU memory size and more computation time are needed, adding to the existing inefficient problem. For the second adaption that updates the whole dataset in one iteration, the target Neus model still fails to converge. The rendered images also become blurry and the edited images change towards the wrong direction. The cause is that the relatively highly inconsistent multi-view images in the tested examples guide the target dataset and the 3D model to change to conflicting directions at each iteration and thus finally result in an apparent divergence trend.

6.3 Discussion

To further test the current pipeline, more examples of 3D models and text prompts with different features are needed. Based on the observed results, certain improvement techniques can be further applied to this base pipeline. One possible trick is to focus on editing the views from normal angles. Another one is to apply debiasing losses for image edit, which edits the text prompt based on view parameters and clips out the unneeded parts of the edited image according to the view parameters (Hong et al., 2023).

Another direction of constructing an editing pipeline can also be tested. It seems to be better in dealing with the multi-view inconsistency as such a method can even be used in 3D model generation, which may naturally face more severe inconsistent problems. The detailed procedures are as below:

1. Randomly select one view and render the image from the current target Neus model.
2. Calculate the SDS loss between the rendered image and the corresponding edited result using the pre-trained InstructPix2Pix Diffusion Model.
3. Backpropagate the 2D image loss to the Neus model and update it together with other losses, like the Eikonal loss to regularize the SDF (P. Wang et al., 2021).

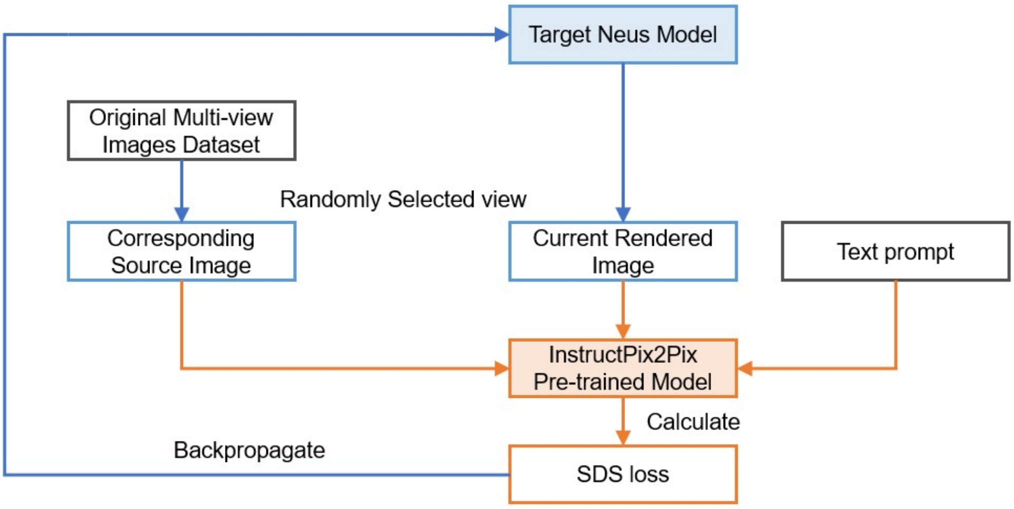
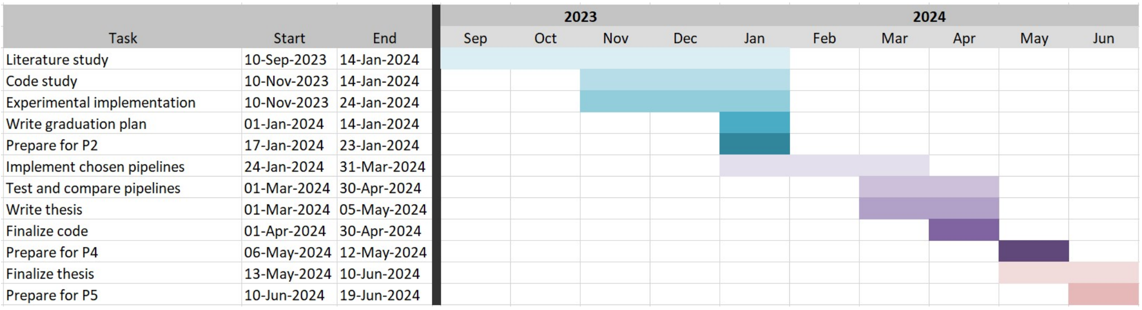


Figure 9: Proposed pipeline: backpropagate image SDS loss calculated by pre-trained InstructPix2Pix back to 3D model

7 Time planning

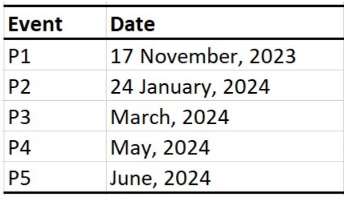
The key activities to be completed during the graduation period and the corresponding periods are planned as below. The key events and dates are also listed.

As for meetings with supervisors, weekly meetings will be held with the first supervisor when necessary. Bi-weekly meetings are scheduled with both first and second supervisors.



			2023				2024					
Task	Start	End	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Literature study	10-Sep-2023	14-Jan-2024	■	■	■	■	■					
Code study	10-Nov-2023	14-Jan-2024			■	■	■					
Experimental implementation	10-Nov-2023	24-Jan-2024			■	■	■					
Write graduation plan	01-Jan-2024	14-Jan-2024					■					
Prepare for P2	17-Jan-2024	23-Jan-2024					■					
Implement chosen pipelines	24-Jan-2024	31-Mar-2024					■	■	■			
Test and compare pipelines	01-Mar-2024	30-Apr-2024							■	■		
Write thesis	01-Mar-2024	05-May-2024							■	■		
Finalize code	01-Apr-2024	30-Apr-2024								■		
Prepare for P4	06-May-2024	12-May-2024									■	
Finalize thesis	13-May-2024	10-Jun-2024									■	■
Prepare for P5	10-Jun-2024	19-Jun-2024										■

Figure 10: Graduation time planning divided into three parts: P1-P2 (blue), P3-P4 (purple) and P5 (pink)



Event	Date
P1	17 November, 2023
P2	24 January, 2024
P3	March, 2024
P4	May, 2024
P5	June, 2024

Figure 11: Dates of key events

References

Armandpour, M., Zheng, H., Sadeghian, A., Sadeghian, A., & Zhou, M. (2023). Reimagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*.

Brooks, T., Holynski, A., & Efros, A. A. (2022). Instructpix2pix: Learning to follow image editing instructions. *arXiv e-prints*, arXiv–2211.

Chen, M., Xie, J., Laina, I., & Vedaldi, A. (2023). Shap-editor: Instruction-guided latent 3d editing in seconds. *arXiv preprint arXiv:2312.09246*.

Chen, R., Chen, Y., Jiao, N., & Jia, K. (2023). Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*.

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, *34*, 8780–8794.

Dong, J., & Wang, Y.-X. (2023). Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. *Thirty-seventh Conference on Neural Information Processing Systems*.

Fang, S., Wang, Y., Yang, Y., Tsai, Y.-H., Ding, W., Zhou, S., & Yang, M.-H. (2023). Editing 3d scenes via text prompts without retraining. *arXiv e-prints*, arXiv–2309.

Haque, A., Tancik, M., Efros, A. A., Holynski, A., & Kanazawa, A. (2023). Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*.

He, Y., Bai, Y., Lin, M., Sheng, J., Hu, Y., Wang, Q., Wen, Y.-H., & Liu, Y.-J. (2023). Text-image conditioned diffusion for consistent text-to-3d generation. *arXiv preprint arXiv:2312.11774*.

Hong, S., Ahn, D., & Kim, S. (2023). Debiasing scores and prompts of 2d diffusion for view-consistent text-to-3d generation. *Thirty-seventh Conference on Neural Information Processing Systems*.

Kamata, H., Sakuma, Y., Hayakawa, A., Ishii, M., & Narihira, T. (2023). Instruct 3d-to-3d: Text instruction guided 3d-to-3d conversion. *arXiv preprint arXiv:2303.15780*.

Li, C., Zhang, C., Waghwase, A., Lee, L.-H., Rameau, F., Yang, Y., Bae, S.-H., & Hong, C. S. (2023). Generative ai meets 3d: A survey on text-to-3d in aigc era. *arXiv preprint arXiv:2305.06131*.

Li, W., Chen, R., Chen, X., & Tan, P. (2023). Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. *arXiv preprint arXiv:2310.02596*.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*.

Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, *41*(4), 1–15.

Po, R., Yifan, W., Golyanik, V., Aberman, K., Barron, J. T., Bermano, A. H., Chan, E. R., Dekel, T., Holynski, A., Kanazawa, A., et al. (2023). State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204*.

Poole, B., Jain, A., Barron, J. T., & Mildenhall, B. (2022). Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, *35*, 36479–36494.

Shen, T., Gao, J., Yin, K., Liu, M.-Y., & Fidler, S. (2021). Deep marching tetrahedra: A hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, *34*, 6087–6101.

Shi, Y., Wang, P., Ye, J., Long, M., Li, K., & Yang, X. (2023). Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*.

Song, L., Cao, L., Gu, J., Jiang, Y., Yuan, J., & Tang, H. (2023). Efficient-nerf2nerf: Streamlining text-driven 3d editing with multiview correspondence-enhanced diffusion models. *arXiv preprint arXiv:2312.08563*.

Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., & Wang, W. (2021). Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*.

Wang, Q., Zhang, B., Birsak, M., & Wonka, P. (2023). Instructedit: Improving automatic masks for diffusion-based image editing with user instructions. *arXiv preprint arXiv:2305.18047*.

Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., & Zhu, J. (2023). Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*.

Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., & Lipman, Y. (2020). Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, *33*, 2492–2502.