Single-Image 3DGS Scene Reconstruction with Geometry-Aware Priors

Machine Visual Perception Course Project Report

December 1, 2025

Information

Authors: TEST
Group Number: TEST

1 Chapter 1: Introduction and Motivation

1.1 Section 1.1: Introduction to the problem

[Provide a thorough introduction to the problem and why it is important. Briefly explain what general techniques there are and how your project fits.]
RCONSTRUCTING 3D MODELS FROM IMAGES IS ACHANLEGNING TASK THAT REQUIRES LARGES NUMBER OF REFERNCE/INPTU IMAGES AND INTENSE PROCESSING TYPICALLY DONE WITH ENRF OR OTHER MODERN ADVANCEMENTS FOCUS ON REDUCING INPOUT PHOTO NUMBER ND PROCESSIGN COMPELXITY SPECIFICALLY, Developments in 3D Gaussian Splatting methods allow for 3D scene reconstruction using single or few RGB images.

While faster than other scene reconstruction techniques and requiring only a ”one-shot” pass, these approaches often suffer from challenges such as layout/scale drift, over-smooth geometry and hallucinations in occluded regions.

This project focuses on one recent method, SplatterImage [1], as a baseline. By storing 3D Gaussians in an image, it reduces reconstruction to learning an image-to-image neural network, allowing the use of a 2D U-Net to form the representation. Each pixel stores the parameters for a corresponding 3D Gaussian, allowing for reconstruction in a single feed-forward pass. This reduces the need for large amounts of compute.

Despite its speed, SplatterImage does have some issues that have been noted in related works, particularly in reconstructing structures unseen in the input view, including for views significantly different from the source. This project aims to address this by proposing a lightweight augmentation to the model, by integrating explicit geometry priors (such as planes, normals, visibility cues, depth, segmentation, edge maps) with minimal architectural changes.

EXPAND/REWORD ABOVE

1.2 Section 1.2: Background and related work

[Include a few very relevant related works and how your work relates to those, expanding on the previous section. We do not expect you to cover all previous works.]

Needs elaboration and rewording

Over the years, other representations for single-view 3D reconstruction have been used. Traditional methods typically use explicit 3D representations such as point clouds [2] or meshes [3]. Implicit representations like NeRF [4] have also been used, but are slow to render.

discuss triplane?

The triplane representation was proposed to efficiently and expressively represent 3D volumes [5], as a compromise between rendering speed and memory consumption. They were shown to scale to large datasets like Objaverse [6][7], but at the cost of hundreds of GPUs for multiple days [8].

discuss 3dgs

3D Gaussian splatting [9] was proposed to offer real-time radiance field rendering by introducing a 3D Gaussian scene representation, speeding up scene optimization and novel view synthesis while maintaining a high quality.

discuss splatterimage

Splatter Image [1] then applies Gaussian Splatting to monocular reconstrution by using a set of 3D Gaussians as the 3D representation. It predicts a 3D Gaussian for each of the input image pixels and uses a 2D image as the container of the 3D Gaussians, storing the parameters of one Gaussian per pixel. This reduces the reconstruction problem to learning an image-to-image neural network, allowing the reconstructor to be implemented utilizing only efficient 2D operators. The use of Gaussian Splatting in this approach increases rendering and space efficiency, which benefits inference and training. Our work continues to expand on this method through investigating different geometry priors and integrating them into the current model as appropriate.

1.3 Section 1.3: Overview of the idea

[Provide an overview stating why the idea of the project makes sense and what the main motivation is.] splatteriamge suffers from hallucination and problems simply due to lack of data feeding models additional data improves reconstruction with modern compute and ml advancements there now exist many good quality pretrained geometry related models for example generating depth, normal maps, segmentation we propose using the knowledge/capacibiltiy of these models to predict additional priors of input images, creating a modified model that accepts these priors these should result in an improved recostruction quality we propose performing ablation study to see which priors are most effective/significant in changing the reconstruction quality

2 Chapter 2: Method

2.1 Section 2.1: Baseline algorithm

[Explain the baseline architecture you used to build your algorithm on. You may reproduce figures from the original papers.]

explain 3dshg output explain spaltteriamge architecture with diagram

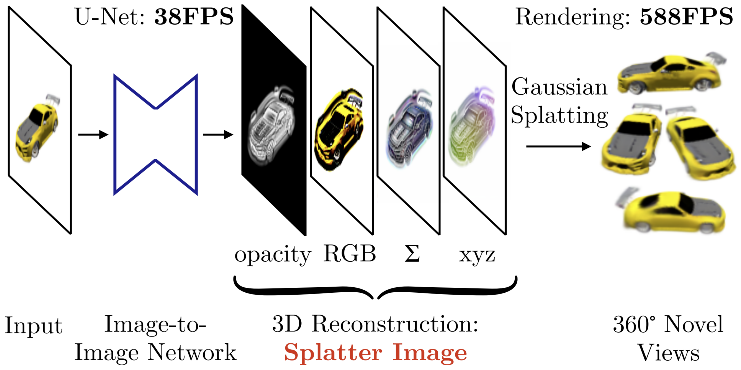


Figure 1: Overview of SplatterImage taken from [1]

Splatter Image uses a standard image-to-image neural network architecture to predict a Gaussian for each pixel of the input image I, generating the output image M as the Splatter Image. Learning to predict the Splatter Image can be done on a single GPU using at most 20GB of memory at training time for most single-view reconstruction experiments (except for Objaverse, where 2 GPUs were used and 26GB of memory was used on each). Most of this neural network architecture is identical to the SongUNet of [10], but the last layer is replaced with a 1×1 convolutional layer with $12 + k_c$ output channels, where $k_c \in \{3, 12\}$ depending on the colour model. The output tensor codes for parameters that are then transformed to opacity, offset, depth, scale, rotation and colour respectively. These parameters are then activated by non-linear functions to obtain the Gaussian paramters, such as the opacity and depth. The Gaussian Splatting implementation of [9] is used for rasterization to generate 360° views of the original input image.

2.2 Section 2.2: Algorithm improvements

[Explain what you implemented to improve over the baseline. You may include figures to explain the idea and logic. Focus on the ideas and not the implementation.]

explain insertion of additional layers explain addition of transofrmer/FiLM layers to allow multimodal input from segmentation tokens explain addition of LORA matrices due to compute limitations/allowign working of exisitng model

2.3 Section 2.3: Implementation details

[Explain how you implemented the improvements. You may include code snippets with the corresponding explanations.]

2.3.1 Normal Map Exploration

We studied using normal maps as a 3D spatial prior to augment the visual tokens for the Splatter Image. A normal map stores surface normal data as RGB colour information, which shows how light interacts with the surface at a per-pixel level, hence we wanted to investigate if that could help the model generate the 3D surface of the input image.

For our ground truths we use [11] which provides a dataset of higher resolution images of the ShapeNet models from [12] each paired with a depth image, a normal map and an albedo image at `https://github.com/Xharlie/ShapenetRender_more_variation`. We then feed these images into the normal map generation models and compare them to the given normal maps to evaluate their performance.

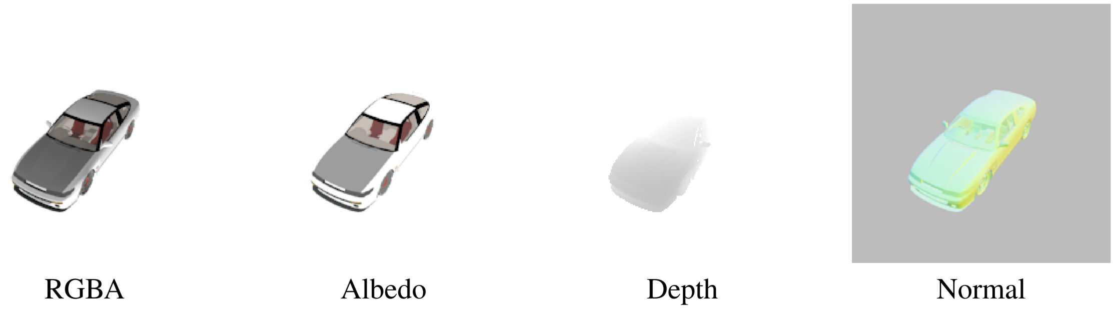
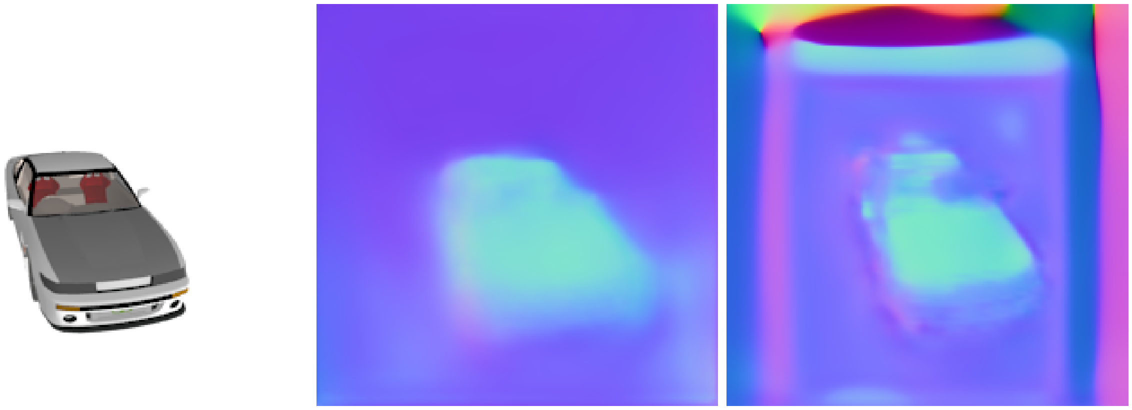


Figure 2: Example of image with maps used as ground truth taken from [13]

For the models we referenced [13] which implements a network which estimates the per-pixel surface normal probability distribution and uses uncertainty-guided sampling to improve the quality of prediction of surface normals. The paper provided code at `https://github.com/baegwangbin/surface_normal_uncertainty` that implemented this method on a network trained on ScanNet [14], with the ground truth and data split provided by FrameNet [15], and another trained on NYUv2 [16], with the ground truth and data split provided by GeoNet [17] [18]. Both models take in the original image and dimensions of the image as input and return a corresponding normal map with the same dimensions as the given input dimensions.

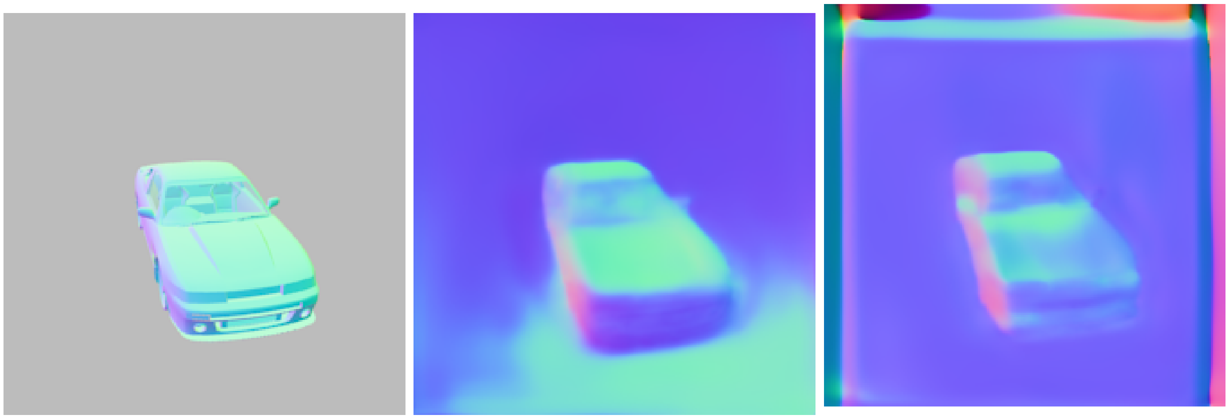
We run both pretrained models on the dataset.



(a) Original (b) ScanNet output (224x224) (c) NYUv2 output (224x224)

Figure 3: Comparison of original input and two model outputs

We then pass in input dimensions larger than the actual ones into the models, such that a normal map larger than the original input is produced. We then resize the image to the original input dimensions.

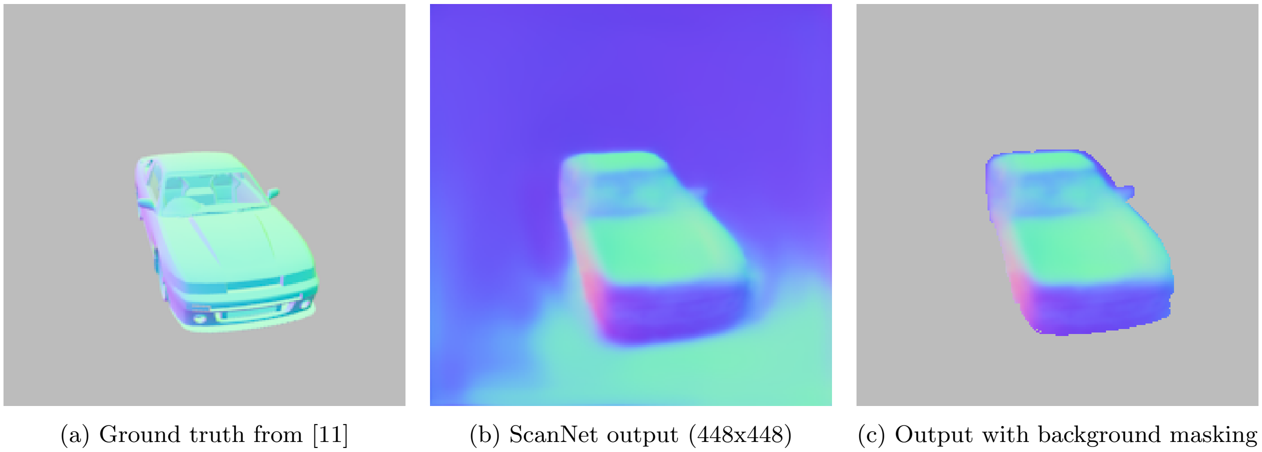


(a) Ground truth from [11] (b) ScanNet output (448x448) (c) NYUv2 output (448x448)

Figure 4: Comparison of model outputs when setting input dimensions as 448x448 instead of 224x224 alongside ground truth

The normal map generated for images when given larger input dimensions seem to have more clearly defined edges and surface contouring. It is also important to note that the ground truth for NYUv2 is only defined for the centre crop of the image and the prediction is therefore not accurate outside the centre. This is shown in figures 3c and 4c where noise is generated around the borders of the normal maps.

To compare our generated normal maps to the ground truth normal maps provided in [11], we first mask out the background of the generated normal maps such that the difference in background colour does not contribute to the evaluation metrics for normal map generation.



(a) Ground truth from [11]

(b) ScanNet output (448x448)

(c) Output with background masking

Figure 5: Example of masking out background for model evaluation against ground truth

We then use Pixel Based Visual Information Fidelity to compare the normal maps generated by the two models to the ground truth. Visual Information Fidelity is a reference image quality metric that quantifies the amount of visual information preserved after image processing [19] and can be used to measure various image quality attributes such as noise level and sharpness [20].

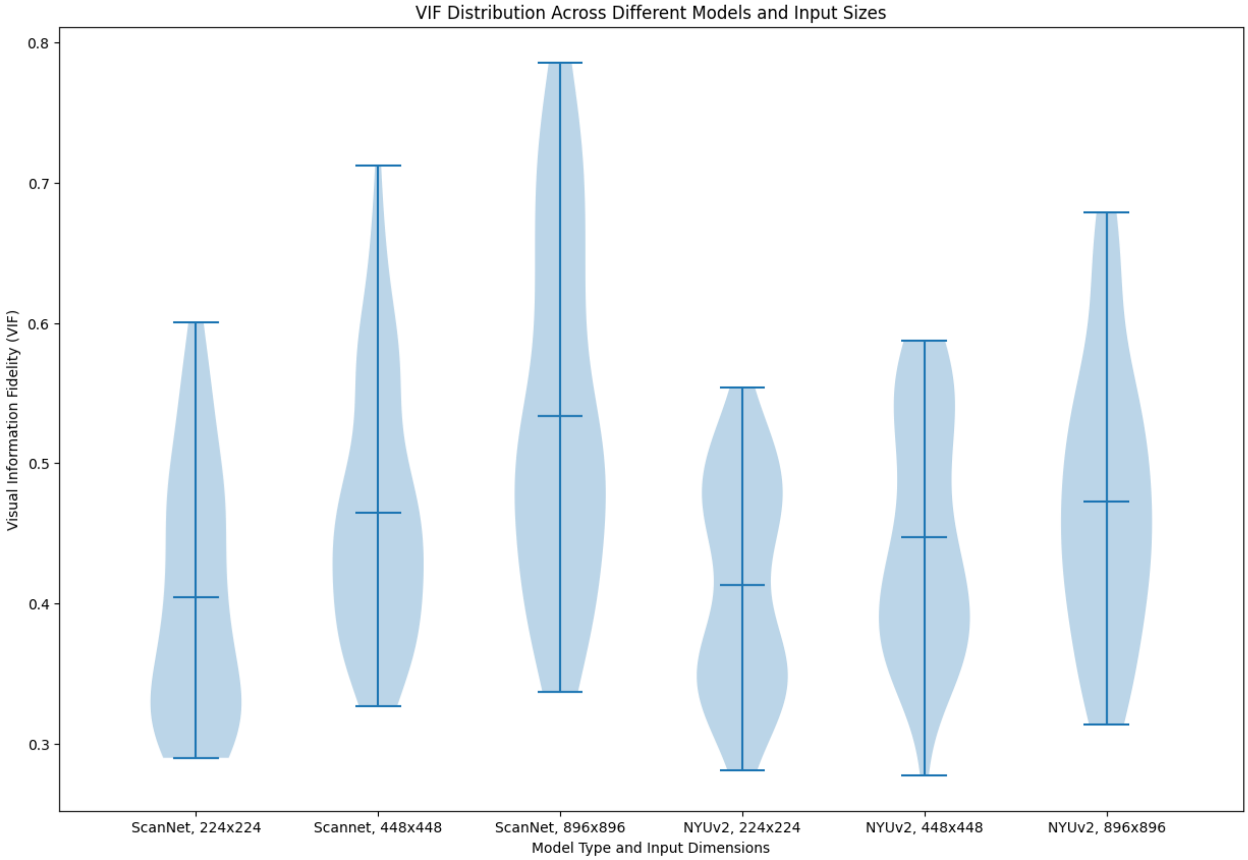
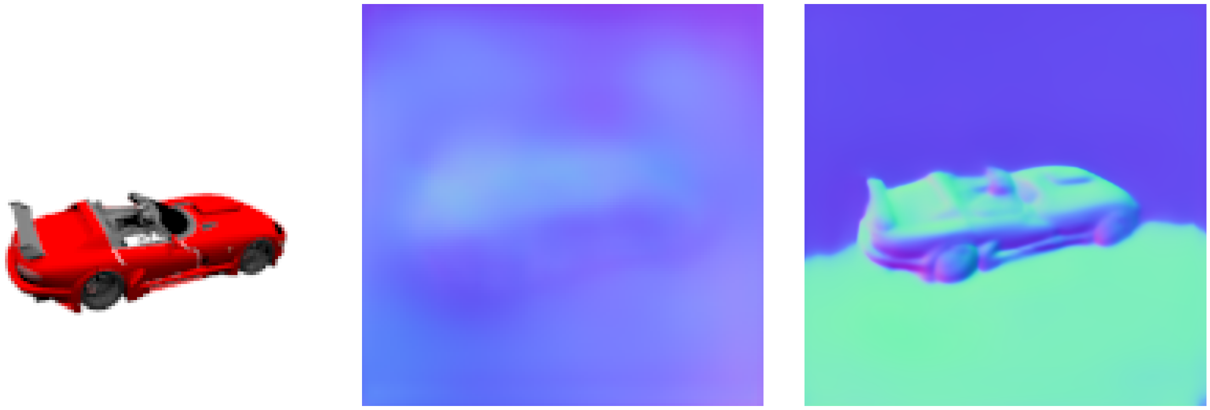


Figure 6: Comparison of VIF between ground truth and different models

From Figure 6 we see that the model trained on ScanNet generates normal maps that are closer to the ground truth compared to that trained on NYUv2 on average. Hence, in the final model we decided to use the model trained on ScanNet on the ShapeNet database in [12].



(a) Original image from [12] of size 128x128

(b) Output from ScanNet model with 128x128 passed in as input dimensions

(c) Output from ScanNet model with 512x512 passed in as input dimensions

Figure 7: Original ShapeNet image and normal map outputs

Without passing in dimensions larger than the input image into the model, we can see from comparing Figures 3a and 3b to Figures 7a and 7b that the quality of the normal map generated decreases as the resolution of the original input image decreases. Hence, we pass in much larger input dimensions (512x512) to generate a normal map of higher quality, as shown in Figure 7c.

2.3.2 Depth Map Exploration

Depth maps store the distance of a surface from the camera per-pixel. These distances vary in type, such as metric, which considers the physical distance from the camera to the observed point, and relative (such as those produced by the models below). Monocular depth estimation (MDE) models input just a singular image, and produce a depth map (relative distance).

Produced depth maps were compared against the "ground truths" produced by `https://github.com/Xharlie/ShapenetRender_more_variation`, as was done in the normal priors exploration. An example of the depth map produced by them is visible in Figure 2. However, it is important to note that these depth map "ground truths" were not always perfect, as can be seen in the following example:

- (INSERT) Image needed here of poor ground truth.

This inclined us to take the quantitative results produced by comparing MDE models tested against these ground truths with a pinch of salt. For each produced depth map, the following metrics were used to compare against the ground truths.

1. **Absolute Relative Error**: Measures the average difference between the predicted depth and the ground truth, normalised by the ground truth depth.

2. **Root Mean Squared Error (RMSE)**: Calculates the standard deviation of the residual errors.

3. **Scale-invariant RMSE (SI-RMSE)**: Computes the RMSE while ignoring the unknown absolute scale and shift between the prediction and ground truth.

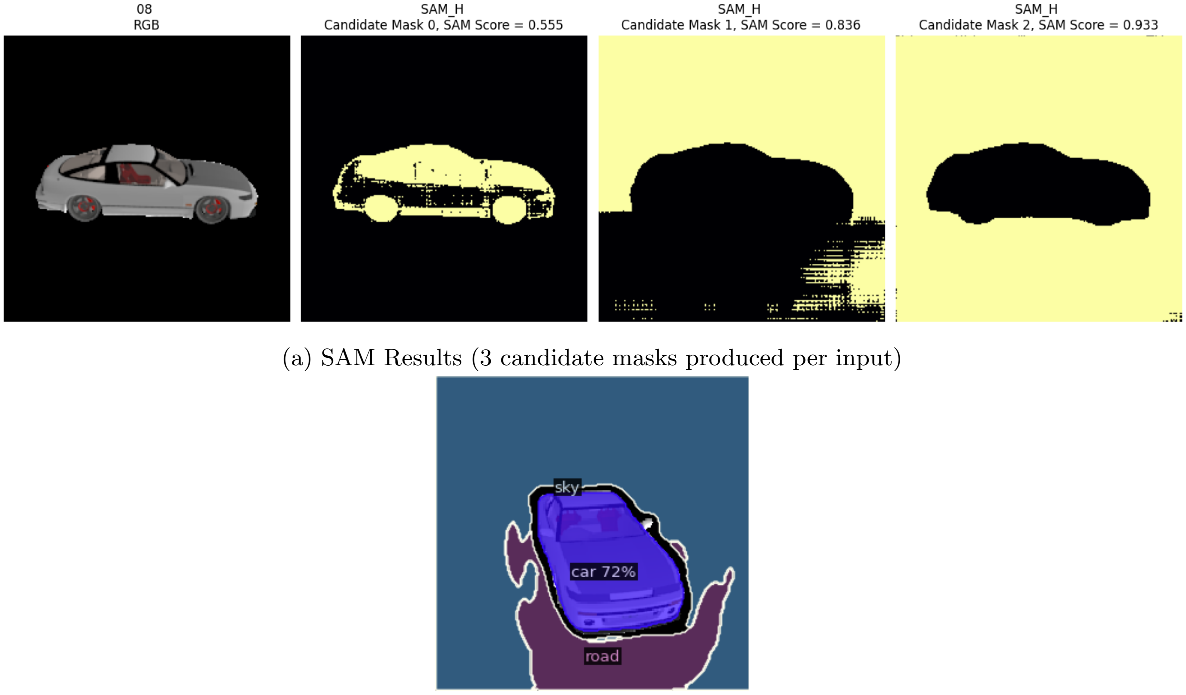
4. δ **at 1.25** ($\delta_{1.25}$): Represents the percentage of predicted pixels p that satisfy the condition $\max(\frac{p}{p^{gt}}, \frac{p^{gt}}{p}) < 1.25$, which takes into account close pixel-wise agreement.

The following table (LINK to label) summarises the mean metrics across the MiDaS models tested.

2.3.3 Segmentation and Salient Object Detection Exploration

Separating pixels belonging to the foreground object, through a segmentation mask or, as will be detailed below, using a salient object detection (SOD) model, can be another prior. This involves producing a binary mask that separates an object from its background.

Initially, we explored standard semantic and panoptic segmentation models, such as those found in the Detectron2 [21] model zoo, and the Segment Anything Model (SAM) [22]. These models are often used for segmentation, but as illustrated in Figure 8, these produced non-contiguous masks that often had sections that included more background pixels. Segmentation models are also limited on their training classes, and despite being tested on categories in this set, their masks were improved on by salient object detection models.


08
RGB
SAM_H
Candidate Mask 0, SAM Score = 0.555
SAM_H
Candidate Mask 1, SAM Score = 0.836
SAM_H
Candidate Mask 2, SAM Score = 0.933
sky
car 72%
road

(a) SAM Results (3 candidate masks produced per input)

(b) Panoptic Segmentation (Detectron2 Model) Result

Figure 8: Sample outputs from standard segmentation approaches.

SOD models identify the most visually distinct object in a scene, which allows producing a binary mask that tightly hugs the object boundary.

To quantitatively evaluate SOD models, we noted that the ShapeNet images used (the same as in the normal and depth priors section) had transparent backgrounds, allowing using the alpha channel to be used as the 'ground truth' for the object silhouette.

We tested three SOD architectures: rembg (based on the U-2-Net architecture) [23], InSPyReNet [24], and BiRefNet [25]. We evaluated performance using Mean Absolute Error (MAE), Intersection over Union (IoU), and F_β-Measure.

2.3.4 Prior Model Integration

TWO MODEL SCRIPTS, PREGENERATED FOR TRAINING/EVAL/VAL FINAL MODAL ORCHESTRATOR PATTERN PER MODEL SCRIPT BACTEHD, SAVES TO IMAGE DUE TO SPACE LIMITATIONS GENERATE PARQUET PER PRIOR PARQUETS UPLOADED TO HF

ON THE FLY VERSION? ANOTEHR OCRHSESTOROR, IMAGES ADDED AND FED INTO MODEL TO ALLOW NOVEL IMAGES

2.3.5 Model Changes

MODEL CHANGES added layer parameters on creation if congfig, inject lora and perform weight graft

2.3.6 Training and Evaluation Changes

new priors stacked into input eval/train on ready dataset have minimal changes added new dataloader code novel evaluator?

2.4 Section 2.4: Data pipelines

[Explain your data format, how you consume the data in your algorithms, and data augmentation.]

EXPLAIN SRN CARS format We took our input data from the ShapeNet-SRN dataset from [26] at 128 × 128 resolution.

EXPLAIN TRANSFORMATION TO PARQUET EXPLAIN STORE OF uint8s UPLOADED TO HF

TRAINIGN LOADS HF DATASET LOAD HF PRETRAINED WEIGHTS PERFORMS WEIGHT GRAFT TRAINING OCCURS WEIGHTS FUSED UPLOADED TO HF

2.5 Section 2.5: Training procedures

[Explain which framework and optimizers you use, how you implemented the training logic.]

USE TORCH LIST LORA INSERTTION LIRBARY LIST BASELINE WEIGHT PARAMS

WE WRITE TRAINIG NNTOEBOOK RUSN ON CLOAB WITH A100 LSIT TRAINING PARAMS WEIGHTS FUSED AND UPLAODED TO HF

2.6 Section 2.6: Testing and validation procedures

[Explain which framework you use, how you implemented the testing/ validation logic.]

3 Chapter 3: Experiments and Evaluation

3.1 Section 3.1: Datasets

[Explain the datasets utilized: what they contain, why they are utilized, assumptions, limitations, possible extensions.]

The standard benchmark for evaluating single-view 3D reconstruction is ShapeNet-SRN [26], hence we used this to test and evaluate our main model implementation. For this dataset, we specifically use the "Car" class, which used the "car" class of ShapeNet v2 [12] with 2.5k 3D CAD model instances. The SRN dataset was generated by disabling transparencies and specularities and training on 50 observations of each instance at a resolution of 128 × 128 pixels, with camera poses being randomly generated on a sphere with the object at the origin. A limitation of this dataset is the lack of subject variety in the dataset as the model may end up overfitting to cars. A possible extension to address this limitation could be to include other classes in the ShapeNet-SRN database to make sure that the model can still generalise to other types of objects.

An extension of this dataset is implemented in [11], which presents a Deep Implicit Surface Network to generate a 3D mesh from a 2D image by predicting the underlying signed distance fields. In the paper, they generated a 2D dataset composed of renderings of the models in ShapeNet Core [12]. For each mesh model, the dataset provides 36 renderings with smaller variation and 36 views with larger variation (bigger yaw angle range and larger distance variation). The object is allowed to move away from the origin, which provides more degrees of freedom in terms of camera parameters, and the "roll" angle of the camera is ignored since it was deemed very rare in real-world scenarios. The images were rendered at a higher resolution of 224 × 224 pixels and were paired with a depth image, a normal map and an albedo image as shown in figure 2. This dataset was mainly used as a ground truth to evaluate the generation of geometry priors (e.g. normal map and depth map). A limitation of this dataset would be its small size since only 72 samples are available for us to use, such that the performance of geometry prior generation may not be evaluated correctly. However, in the same GitHub repository, the script to generate these images from the ShapeNet Core dataset is provided, so a possible extension given more time could be to include more images by running the script on other objects in the ShapeNet Core dataset.

3.2 Section 3.2: Training and testing results

[Explain the training and testing results with graphs and elaborating on why they make sense, what could be improved.]

3.3 Section 3.3: Qualitative results

[Show in figures and explain visual results. Include different interesting cases covering different aspects/ limitations/ dataset diversity. If not converged, explain what we can expect once converged. Include any other didactic examples here.]

3.4 [Optional] Section 3.4: Quantitative results

[A table and associated explanations for quantitative results.]

3.5 [Optional] Section 3.5: Comparison to state-of-the-art

[Qualitative and/ or quantitative comparisons to one or more recent works, especially the baseline work.]

4 Chapter 4: Conclusions and Future Directions

4.1 Section 4.1: Conclusions

[Summarize what the project was about and the main conclusions.]

4.2 Section 4.2: Discussion of limitations

[Explain the limitations of your technique. You may want to refer to previous sections or show figures on the limitations.]

halluciantion in hidden areas still a problem data, lot of data and compute needed

4.3 Section 4.3: Future directions

[State a few future directions for research and development. These typically follow from the discussion on limitations.]

4.4 Section 4.4: Project Contribution

”You may find the template for the project report here. We do not enforce any page limits but please make sure to address each section appropriately as explained in the document. In particular, please pay special attention to clarifying the contribution of each group member.” Should we clarify here or throughout document?

References

[1] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction, 2024.

[2] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image, 2020.

[3] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn, 2020.

[4] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks, 2022.

[6] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects, 2023.

[7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects, 2022.

[8] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d, 2024.

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023.

[10] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[11] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019.

[12] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015.

[13] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *International Conference on Computer Vision (ICCV)*, 2021.

[14] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[15] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas Guibas. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. *arXiv preprint arXiv:1903.12305*, 2019.

[16] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[17] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.

[18] Xiaojuan Qi, Zhengzhe Liu, Renjie Liao, Philip HS Torr, Raquel Urtasun, and Jiaya Jia. Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[19] Saeed Mahmoudpour and Manbae Kim. Chapter 10 - a study on the relationship between depth map quality and stereoscopic image quality using upsampled depth maps. In Leonidas Deligiannidis and Hamid R. Arabnia, editors, *Emerging Trends in Image Processing, Computer Vision and Pattern Recognition*, pages 149–160. Morgan Kaufmann, Boston, 2015.

[20] Xinwei Liu, Marius Pedersen, and Renfang Wang. Survey of natural image enhancement techniques: Classification, evaluation, challenges, and perspectives. *Digital Signal Processing*, 127:103547, 2022.

[21] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[23] Daniel Gatis. rembg. `https://github.com/danielgatis/rembg`, 2025. Version 2.0.66.

[24] Taehun Kim, Kunhee Kim, Joonyeong Lee, Dongmin Cha, Jiho Lee, and Daijin Kim. Revisiting image pyramid structure for high resolution salient object detection. In *Proceedings of the Asian Conference on Computer Vision*, pages 108–124, 2022.

[25] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 3:9150038, 2024.

[26] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations, 2020.