3D Model Reconstruction from Image(s)

Zhen Lin
Stanford University
jamerust@stanford.edu

June 13, 2025

Abstract

We divide this paper into 6 major sections.

1. Introduction presents the key problem, its motivation, definition, and background
2. Related work review state of art approaches and our focus
3. Methods details our technical steps and design decisions
4. Data presents important data construction details
5. Experiments covers our results and observations
6. Conclusion summarizes our findings and contributions

1. Introduction

1.1. Problem definition

This work focus on the problem of generating 3D model from 2D image(s).

Concretely, as input, we are given a set of N images $\{(I^{W\times H\times C=3})_{i=1}^{N}\}$, each with height H, width W and color channels $C = 3$. These images are unposed and camera parameters are unknown.

Our objective is to generate a corresponding set of per-pixel 3D points—commonly referred to as a point map $\{(X^{W\times H\times D=3})_{i=1}^{N}\}$ where $D = 3$ for 3D space coordinates, typically in world frame of the first image.

We also like to estimate camera intrinsics $K \in \mathcal{R}^{3\times3}$, as well as a set of pose P^m that transform coordinates from the camera frame m to a shared world coordinate system.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where f_x and f_y are focal length in pixels along x and y axes on image plane, and c_x and c_y are coordinates of the principal point (typically near the image center).

1.2. Motivation

This is a challenging yet fundamental problem in computer vision. We pursue this problem from both practical use point of view, to assess current approaches and state of art choice. We also approach it from intellectual point of view, aiming to unravel the inner working of solutions, by modifying one such model and retrain it, and by developing a visualization for another model.

1.3. Main results

We demonstrated the validity of the SOA approach in real data generation - using Structure-from-Motion (SfM [1]) software of Colmap [2], we are able to provide a ground truth data point of object focused images, along with camera parameters, and point map and depth map for each image.

With above data point as running example, we compared the model produced by the following different approaches. We calculate Earth Mover's Distance (EMD) and Chamfer distances (defined in later section) for discriminative models. All these establish that VGGT model is SOA in terms of fidelity and speed.

- combining off-the-shelf models (GroundingDino [3], SegmentAnything [4], DepthAnything [5])
- discriminative models (Dust3R [6], Fast3R [7], VGGT [8])

- generative models (3D-LMNet [9], Point-E [10]), and LRM [11]

For 3D-LMNet, we modify its architecture by introducing DG-CNN for point of cloud auto encoder, and DINO for 2D image encoder. We are able to train for a few epochs on ShapeNet dataset. We observe the improvement of generated point of cloud model.

For VGGT model, we measure the impact on generated model quality and inference speed, when we input a varying number of images. We find inference time is $O(n \log n)$ with n being number of input images. We also find in general, more images lead to better model quality. We also develop a visualization tool that shows the inter image attention mask, for any patch at source image, w.r.t all other images. This helps us to begin to understand the most important factor for this SOA performance.

1.4. Background

Physical understanding begins with modeling the spatial structure of objects and scenes in our world. Reconstructing 3D structures from 2D images is a long-standing goal in computer vision, as it forms the foundation for interpreting the visual world. This capability has broad applications across industrial design, animation, gaming, augmented and virtual reality (AR/VR), and robotics.

Traditional Structure-from-Motion (SfM) approaches, exemplified by systems like COLMAP, rely heavily on multi-view geometry and require accurate calibration of both camera intrinsics and extrinsics. While widely adopted, these methods are often computationally intensive and may fail to converge in challenging scenarios..

Recent advances in deep learning have led to significant progress in 3D reconstruction methods that no longer require known camera parameters at inference time—that is, input images can be unposed, with no prior calibration. This shift has been driven by improvements in 3D datasets (both synthetic and real-world), scalable training infrastructure, and innovations in deep learning architectures. State-of-the-art methods leverage transformer-based architectures—particularly self-attention and cross-attention mechanisms—to allow image embeddings to interact and jointly infer the 3D structure of a scene. Foundational models such as VGGT unify multiple tasks within a single backbone, including camera parameter estimation, relative and global pose prediction, per-pixel point mapping, and dense point cloud reconstruction.

In 3D object reconstruction from one or a few images of the same scene, the goal is to recover a 3D model of the central object. The ground truth typically consists of image–model pairs, along with corresponding camera poses and intrinsic parameters. However, all the models explored in this work perform inference without access to camera pose or intrinsics—they require only one or more images of the same scene.

We adopt point clouds as the primary 3D representation in most of our learning-based models. Point clouds (PC) offer an explicit, lightweight structure that facilitates both quantitative comparison and fast rendering. For broader comparison, we also include the LRM model, which generates NeRF and mesh representations in addition to point clouds.

Reconstructing high-fidelity 3D structure at inference time—especially from uncalibrated cameras and unposed images—is a fundamentally challenging and ill-posed problem. Humans perform this task effectively by leveraging extensive prior knowledge about object shapes, spatial configurations, and scene semantics. Learning-based models must similarly compensate for the absence of full geometric specifications by incorporating both 2D scene understanding and 3D shape priors. For instance, DINO [13] embeddings capture rich 2D perceptual features, while architectures such as PointNet [14], DGCNN [15], and various encoder–decoder models for point clouds encode 3D structural knowledge. Successfully training such models typically requires large-scale datasets, often exceeding one million images and corresponding 3D point cloud models.

Over the years, the input images for these learning models have evolved to become increasingly diverse and realistic. Early datasets consisted of simple, category-based, computer-generated images—such as ShapeNet used in 3D-LMNet. More recent datasets, like Objaverse [16] and MVImgNet [17] used in LRM, feature a broader range of real-world object images often presented against white backgrounds. The latest datasets, such as Co3D [18] used in VGGT, contain diverse real-world scene images with complex backgrounds and varying contexts.

Learning model architectures typically follow either a discriminative approach, framing the task as a regression problem, or a generative approach, employing autoencoders and diffusion models conditioned on input images. Increasingly, architectures are designed to promote coupling and self-consistency among multiple observed images, reducing the need for explicit inductive biases in model design.

2. Related work

There are broadly four categories of approaches, presented here in chronological order.

2.1. SfM (structure of motion) based approach

Structure-from-Motion (SfM) is effective when consecutive images contain a large overlapping portion of the scene. Given a 3D point and the optical centers of two consecutive images, epipolar geometry defines the relationship between corresponding points. Using epipolar constraints, SfM determines the intersection of rays projecting from each camera's optical center through matching image points, thereby estimating the 3D location of the point. Additionally, global consistency is improved through bundle adjustment, which refines 3D estimates by jointly optimizing camera poses and 3D points—especially when a sequence of images forms a loop closure.

The dominant offline solution currently is COLMAP, a battle-tested and widely adopted system known for its high accuracy. It emphasizes precision and detail rather than real-time performance. Given a collection of unposed, unordered images, COLMAP can estimate camera intrinsics and poses, and reconstruct per-image point maps. These point maps can then be fused into a dense 3D model, represented as a point cloud or a mesh. Like other methods in this category, COLMAP requires substantial computational resources—particularly, its dense stereo reconstruction step (patch_match_stereo) relies on GPU acceleration.

The leading real-time online solution, often used in SLAM (Simultaneous Localization and Mapping), is DROID-SLAM (Deep Visual SLAM) [19]. It prioritizes speed, robustness, and adaptability to dynamic environments by combining classical geometric methods with deep neural networks. Given a sequence of ordered images, DROID-SLAM continuously localizes the camera while incrementally building a map of the environment.

Our study focuses on reconstructing objects from an unordered set of images in an offline setting, unlike SLAM methods which primarily focus on camera pose estimation. To establish ground truth dense models of objects, we use COLMAP as our reference system.

2.2. Combination of specialized learning based models

When camera intrinsics are known, a single image can provide sufficient information to reconstruct a 3D point cloud of the central object. By leveraging 2D perceptual priors—such as those from models like DINO—and applying established deep learning models trained for single-image depth or correspondence estimation, depth values can be inferred for each pixel. Combining these per-pixel depth estimates with the known camera intrinsics enables projecting the image into 3D space, thereby generating a point cloud representation.

2.2.1 2D perceptive model

State-of-the-art models have evolved from CNN-based supervised approaches to self-supervised transformer-based methods. The latter offer superior model expressiveness and scalability, benefiting from large unlabeled datasets. Most of these models are trained on millions of images.

For example, MAE (Masked Autoencoder) [20] learns low- to mid-level features by reconstructing missing parts of images, optimizing an L2 reconstruction loss. In contrast, DINO (Self-Distillation with No Labels) learns mid- to high-level features by enforcing consistency between feature representations from a student network and a teacher network (which uses exponential moving average weights). This is done via a cross-entropy loss over softmaxed features computed from differently transformed versions of the same image.

In our work, we explore 3D-LMNET for 3D point cloud generation, which requires rich semantic image information. To enhance the expressiveness of the image encoder, we replace the original custom convolutional encoder with DINO.

2.2.2 Language-Prompted Object detection model

These models typically include dual encoders for language and images, cross-modal fusion mechanisms to facilitate interaction and alignment, and an image decoder that predicts bounding boxes and object probabilities. Open-vocabulary and zero-shot detection capabilities are enabled by strong visual-semantic alignment.

YOLO-World [21] combines CNN and CLIP [22] encoders and fuses them with RepVL-PAN (Re-parameterizable Vision-Language Path Aggregation

Network). Its prompt-then-detect paradigm allows real-time performance, though with slightly reduced accuracy.

GroundingDINO [3] integrates DINO and GLIP, enhanced with feature fusion techniques, achieving higher accuracy at the cost of increased computational resources.

Since our study is offline and prioritizes high accuracy, we adopt GroundingDINO as our detection backbone.

2.2.3 Segmentation model

Encoder-decoder architectures are commonly used for image segmentation to produce dense, pixel-wise predictions. Typically, the encoder backbone is based on CNNs or Vision Transformers (ViT), while the decoder employs upsampling layers such as transposed convolutions, U-Net [23], FCN [24], or DPT [25], often with skip connections to preserve spatial details.

DINO combined with k-means clustering forms a self-supervised model that uses contrastive loss. It provides class-agnostic masks with moderate accuracy and a medium model size (80 million parameters).

DeepLabV3 [26] is a supervised segmentation model delivering high semantic segmentation accuracy with a medium model size (60 million parameters).

SAM (Segment Anything Model) [4] is a foundation model that offers zero-shot segmentation with excellent accuracy, but at a much larger scale (2 billion parameters).

Since our study is offline, requires high accuracy, and does not need semantic segmentation, we adopt SAM for mask generation.

2.2.4 Depth estimation model

The encoder-decoder structure with skip connections is a standard architecture in monocular depth estimation models. The encoder typically uses CNNs or Vision Transformers (ViTs) [25], while the decoder employs transposed convolutions or upsampling methods such as bilinear or nearest-neighbor interpolation.

MiDaS [27] leverages ResNet and ViT backbones to deliver fast, zero-shot predictions of relative depth. It achieves moderate accuracy with a medium-sized model (300M parameters).

DepthAnything [5], also ViT-based, provides zero-shot relative depth estimation with high accuracy. It operates at a moderate inference speed and has a large model size (600M parameters).

ZeroDepth [28] uses a DPT backbone to predict metric depth maps in a zero-shot manner. It offers high accuracy but slower inference speed, with a medium model size (300M parameters).

Metric3D [29], [30] is also a more recent SOTA model to predict depth.

Since our objective is high-accuracy relative depth estimation, and for consistency with reconstruction models that operate in relative scale, we adopt the DepthAnything model.

2.3. 3D reconstruction from unposed image(s)

3D reconstruction generally involves a multi-stage architecture that jointly estimates camera parameters and 3D scene representations. These pipelines typically consist of: a feature extraction backbone (e.g., ResNet, ViT, or DINO), an attention-based feature alignment or matching module, and prediction heads for camera intrinsics/extrinsics, per-pixel depth, or dense point maps.

There are three common output formats for 3D representations:

1. 3D Point Cloud (PC) models generate raw point sets or meshes.
2. NeRF [31] models reconstruct continuous radiance fields.
3. 3D Gaussian Splatting (3DGS) [32] models output point clouds augmented with learned Gaussian parameters.

Until recently, only 3D PC-based models were capable of handling unposed and uncalibrated images. However, recent advances have extended this capability to implicit representations. Notably, LRM (a NeRF-style model) can reconstruct 3D scenes from a single unposed image, and Dust3R demonstrates the ability to handle multiple unposed views in a point cloud-based framework.

Each representation has trade-offs:

- 3D PC models are well-suited for robotics, SLAM, and general-purpose scene understanding, due to their explicit and interpretable structure.

- NeRF and 3DGS models excel in high-fidelity rendering and real-time AR/VR applications, owing to their photorealism and differentiable rendering techniques.

Traditionally, NeRF and 3DGS models rely on differentiable rendering and require multiple posed views. NeRF leverages volume rendering, while 3DGS projects 3D Gaussians onto the 2D image plane. In contrast, 3D PC models typically avoid differentiable rendering and operate on single, unposed views. This distinction is now blurring: LRM shows implicit representations can work with a single unposed image, while Dust3R advances 3D PC models to multi-view unposed scenarios.

In this study, we focus on 3D PC-based models, given their compatibility with unposed input and explicit reconstruction. As a point of comparison, we also examine LRM, which generates implicit 3D representations (e.g., SDF, NeRF) from a single unposed image, from which meshes and point clouds can be extracted.

We now proceed to survey state-of-the-art methods in 3D point cloud reconstruction, organized by the discriminative vs. generative modeling paradigm.

2.3.1 Discriminative approach

Discriminative models reconstruct 3D point clouds by explicitly inferring geometry from input images using cues such as correspondence, depth, or structure. These models are typically formulated as supervised regression tasks. They follow a standard pipeline: encoding image(s) into latent representations, performing frame-wise and global cross-attention for alignment, and mapping the aggregated features to a 3D point map prediction.

1. DUSt3R employs a transformer-based encoder (CroCo, similar to DINO) to extract rich image features. It uses cross-attention between embeddings of two images to form a fused representation, which is then decoded to predict both the 3D point map and camera poses. However, it is limited to two input images, and for handling more than two views, it must perform global alignment at runtime, resulting in an $O(N^2)$ complexity.

2. Fast3R addresses the scalability limitation of DUSt3R by introducing a frame index embedding, which enables fusion of an arbitrary number of images. It operates in two stages: first, per-view ViTs extract tokens independently; second, a global fusion layer aligns tokens across views using cross-attention. This design removes the two-image constraint while preserving the ability to integrate multiple perspectives.

3. VGGT represents the current state-of-the-art. It is designed as a vision foundation model, jointly predicting multiple vision-related quantities—camera intrinsics and extrinsics, per-pixel depth and point maps, confidence scores, and tracking keypoints—using a shared backbone. Its key architectural innovation is a stack of 24 alternating attention blocks, each consisting of a frame-level attention layer followed by a global attention layer. Like other recent models, it uses DINO as its image feature extractor and DPT for dense per-pixel prediction.

We evaluate all three models—DUSt3R, Fast3R, and VGGT—on our ground truth examples to benchmark their effectiveness and limitations in reconstructing 3D point clouds from unposed image sets.

2.3.2 Generative approach

Generative models synthesize 3D point clouds from latent features, text, or images using architectures such as autoencoders, diffusion models, or normalizing flows. These models are also typically formulated as supervised regression tasks. At inference time, they condition the generation of 3D representations—such as point clouds or NeRFs—on input images.

1. PointFlow [33] employs PointNet [14] to generate a latent code for each 3D point. A continuous normalizing flow (CNF) then models the distribution of these latent codes. Finally, a decoder reconstructs the 3D point cloud. Although PointFlow performs unconditional generation, it provides a strong foundation for extending into conditional 3D generation using flows.

2. 3D-LMNet is a two-stage model. It first trains a variational autoencoder (VAE) on 3D point cloud data, using PointNet as the encoder. The decoder from this VAE is then combined with a separate image encoder (originally CNN-based) to learn mappings from image-to-point cloud pairs. It is trained on the ShapeNet dataset. In our work, we replace the CNN encoder with DINO to improve semantic expressiveness.

3. Point-E is a diffusion-based generative model. During inference, it typically uses GLIDE to generate synthetic images. In its second stage, a CLIP

encoder extracts image embeddings, which condition a diffusion model to generate a coarse 3D point cloud (1K points). A third-stage diffusion model then upsamples this to a high-resolution point cloud (4K points). In our study, we bypass the first GLIDE stage and directly input real images into the second stage.

4. LRM (Large Reconstruction Model) represents 3D by NeRF representation. It does not learn the NeRF MLP directly like traditional NeRF. It accepts a single unposed image (preferably with a white background) and predicts a NeRF, from which mesh or point cloud representations can be extracted. During training, DINO-generated image features and position encodings are fused via cross-attention layers, guided by camera features. The latent features are passed through deconvolution layers and an MLP to produce a NeRF field. At inference time, the autoencoder directly conditions on a single image to output the predicted NeRF.

We evaluate the last three models—3D-LMNet, Point-E, and LRM—on our ground truth examples to assess their effectiveness for real-world 3D reconstruction from unposed images.

3. Methods

3.1. Design decision

At high level, we comes from the angle of practical application. We find it is rather grounded to use the same real world ground truth data point, to test performance of competing models. As background section shows, our choice of models is driven by pragmatism and established research results.

From pedagogical and realistic point, we decide to use 3D-LMNET for in depth investigation. Among all models above, only 3D-LMNET stands at around 10M parameters, that affords a training on T4 or V100 for a couple of hours for a few epoch. All the other models stands at at least 500M parameters beyond, out of our machine resource reach. We take the path of modifying 3D-LMNET architecture with better subcomponent, and retrain with the same ShapeNet data, which is around 100K instances.

VGGT model does not yet have full training code released. We focus on experiments on its inference code, by varying numbers of input images, and by visualization through an innovative approach to capture the attention masks from its global blocks.

3.2. Produce PC model from a single image via Combination of specialized learning based models

There are steps to produce a PC from a single input image

1. Camera intrinsics are obtained either from COLMAP (see discussion in the Data section) or inferred using the Fast3R model (see subsection below).

2. Run the GroundingDINO model on the image with the text prompt ”Miffy toy” to obtain a bounding box.

3. Run the SAM model on the image, using the bounding box from the previous step and a point on the target object as constraints, to produce an object mask.

4. Run the DepthAnything model on the image to obtain a depth map.

5. Multiply the object mask with the depth map to extract the depth information corresponding to the target object.

6. Apply perspective projection, using the intrinsics from Step 1, to recover the 3D point cloud of the object.

Most of the models require GPU acceleration; therefore, we conduct our experiments on an AWS T4 instance. We use the ground truth data point (frame_003.png) as the input image. Code references are provided in the Appendix.

3.3. Produce PC model from Discriminative models

We run inference on our ground truth data point to generate 3D point clouds using each of the discriminative models.

1. DUSt3R is evaluated with both 2 and 5 input images to assess the impact of input image count.

2. Fast3R is evaluated with 5 input images.

3. VGGT is evaluated with 5 input images.

All inference steps are successfully executed on a Mac M1 system. Code references are provided in the Appendix.

3.4. Produce PC model from Generative models

We run inference on our ground truth data point to generate 3D point clouds using each of the generative models.

1. 3D-LMNet takes a single input image.
2. Point-E is evaluated with both 1 and 3 input images to assess the impact of input image count.
3. LRM uses 5 input images and is capable of producing both video and mesh representations.

These models require GPU support; thus, we conduct our experiments on an AWS T4 instance.

3.5. Assessing attention weights from different images for a VGGT model

These are the steps taken to assess the attention weights:

1. We modified the VGGT model (in attention.py) and the inference code to expose the attention weights from the last layer of the global_blocks.
2. We performed inference on 5 images and extracted the actual attention weights.
3. We visualized the relative attention weights across the images for a selected 3D point.

All these steps were executed on a Mac M1. The code references are provided in the Appendix.

3.6. Impove the 3D-LMNet model

3D-LMNET training on ShapeNet is barely small enough for us to perform rudimentary training (full training requires several days). We were able to complete training and evaluation using the original codebase, albeit stopping at a fraction of the recommended full training epochs.

We modified the architecture by leveraging the stronger expressive power of DINO as the image feature extractor, replacing the original custom CNN feature extractor. DG-CNN has demonstrated superior performance in segmentation and classification, particularly in representing local neighborhood features with only $O(N \log N)$ complexity. Therefore, we replaced the original PointNet encoder in the autoencoder with DG-CNN. We successfully trained the modified model and used it for inference to generate 3D point clouds conditioned on input images.

For implementation, we find these help improve the model learning

- We unfroze the last two layers of DINO to allow more capacity for fine tuning
- We add repulsion loss via knn distance to promote point distribution uniformity and to reduce point clustering
- We improve upon the original training code to allow checkpoint and train from the checkpoint.
- We also integrate tensorboard so we can track and debug the training process.

We observed that the learning rate for autoencoder training could be significantly increased. We also adjusted the batch size to avoid GPU out-of-memory errors while maximizing GPU utilization.

The T4 GPU is underpowered for training; we found that a V100 machine on GCP is up to four times faster. The code references are provided in the Appendix.

3.7. Study on VGGT model

We vary the number of images feed into VGGT model at inference time to assess the impact on inference time and model quality. We again measure quality by Chamfer and EMD distances. We use input size 1, 5, 10, 15, 20. The images are picked uniformly and evenly on a 360 degree video around a Miffy toy.

To visualize the inter image attention mask, we build a visualization tool based on these algorithms

- Correspondence
  - We would like to find all corresponding pixel on other images, when we click one pixel on a source image
  - The source pixel at source image, can be projected to its 3D point, given the depth map and camera parameters produced by VGGT model.
  - That 3D point can then be projected back to other images, given camera poses produced by VGGT model
  - The reprojected pixel will be checked to ensure that are within corresponding image's boundary.
- Capture of attention mask during inference
  - We modify attention layer to output attention mask
  - We modify VGGT model to use the modified attention layer and store attention mask value

- Summary of layers of attention masks
  - At each layer, we max over head dimension, then discard lowest attention values
  - We accumulate the product of attention mask through each layer

$$A = (A + I)/2 \tag{2}$$

$$A = \frac{A}{\sum A} \tag{3}$$

- Presentation
  - Due to the asymmetry of the attention matrix, we use a grid layout where the first column contains the source images, and subsequent columns show attention maps on the target images.
  - Selecting a pixel in any source image reveals its corresponding locations in the other source images.
  - For each selected source pixel, a corresponding patch is extracted, and the summarized global attention is visualized across the counterpart images.

4. Dataset

One of major challenge in 3D model reconstruction research is the difficulty to prepare ground truth data. Currently, there are two approaches

- Computer generate data
  - When we define a 3D model, we can use computer graphics technique to generate images from arbitrary angle and lightning condition
  - This method can generate unlimited number of data points, yet there is always gap between reality and virtual scene
- Real world data
  - Either hardware or software based (SfM) method can help us produce camera pose estimate, and therefor 3D model of scene object, over real world video.
  - While this method requires considerable computational resources, it achieves a degree of realism that is unparalleled.

We use the first approach when training 3D-LMNET model, and use the second approach to prepare our running example of ground truth datapoint.

4.1. Prepare our ground truth data point for inference

The ground truth data point was produced through a data processing pipeline.

- We began by capturing a 14-second video of a Miffy toy indoors using iPhone 13, covering a full 360-degree view of the scene.

- From the video, 42 images were sampled using ffmpeg.

- We applied a multi-step COLMAP-based processing pipeline to generate a dense 3D point cloud model. The most computationally intensive step, PatchMatch stereo, requires approximately 20 minutes on a V100 GPU.

- The final output consists of images with associated camera poses and intrinsics, along with depth maps and the 3D point cloud.

This is one of the images selected as our running test example.

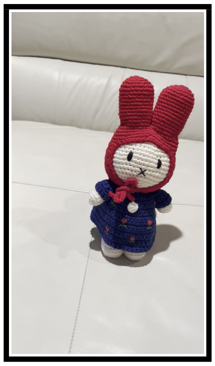

Figure 1. Ground truth image

Our ground truth data point consistent of 20 images, at resolution 720 × 1280, each with its depth map, annotated by camera pose and intrinsics.

This process is inspired by the CO3D dataset methodology, which we adopt to prepare our own data point. It validates the traditional SfM approach and provides a real-world ground truth data point for all subsequent experiments.

This data point is fed into VGGT and pad and resize to resolution 518 × 518, before inference step. See VGGT preprocess code.

4.2. Leveraging ShapeNet dataset in 3D-LMNet model for training

ShapeNet is a repository of CAD models. We adopt this dataset following the 3D-LMNet setup. The paired synthesized images and point clouds serve as data for training and evaluation. The train/validation/test split is as follows

Dataset prepared for 3D autoencoder:
Length of train set: 26271
Length of val set: 8758
Length of test set: 8755

In addition to synthesized images, camera azimuth angle is also provided as input the training. It acts as a supervision signal for pose-related consistency between 2D and 3D representations, during variational autoencoder training for point of cloud data.

5. Experiments and Results

5.1. 3D PC model via discriminative model

5.1.1 Quantitative result

Model size is in terms of number of parameters.

	Model size	Inference time (5 images)
Dust3R	571M	2min8s
Fast3R	647M	2min13s
VGGT	1.2B	1min31s

Table 1. General attributes of discriminative models

Clearly, VGGT is nearly twice as large in model size, yet its inference speed on CPU surpasses the other two models.

To measure the discrepancy between the 3D point cloud generated and our COLMAP-produced ground truth data point, we normalize both point clouds and perform multi-step ICP alignment prior to calculating the Earth Mover's Distance (EMD) and Chamfer distances.

Chamfer Distance computes the average closest-point distance in both directions, from PC P to Q, and Q to P

$$\mathrm{CD}(P,Q) = \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}\|p-q\|^2 + \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}\|q-p\|^2$$

Earth Mover's Distance measures the minimum total cost of transporting mass between two PCD.

$$\mathrm{EMD}(P,Q) = \min_{\phi:P\to Q}\frac{1}{|P|}\sum_{p\in P}\|p-\phi(p)\|$$

Model	Chamfer Distance	EMD
DUST3R	38.15	0.113
FAST3R	59.80	0.155
VGGT	40.86	0.108

Table 2. Comparison of 3D reconstruction models using Chamfer Distance and Earth Mover's Distance (EMD). Lower is better.

Since all methods provide scene-level point clouds, we rely on confidence level predictions to remove extraneous points that do not belong to the central object. However, the metrics below indicate that this approach is less than satisfactory. As future work, we note that combining object masks to precisely filter pixels belonging to the object could yield much cleaner point cloud reconstructions.

5.1.2 Qualitative result

COLMAP provides a reasonably smooth representation of the object as a point cloud model.

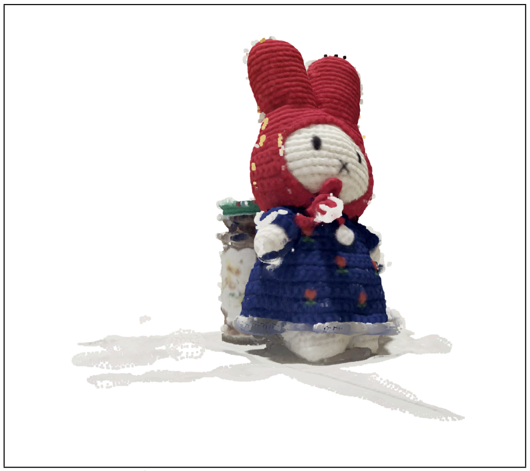

Figure 2. COLMAP produced point cloud model for the miffy object, shown at 80 percent confidence level

The combined method using GroundingDINO, SegmentAnything, and DepthAnything produces a reliable object-specific depth map.

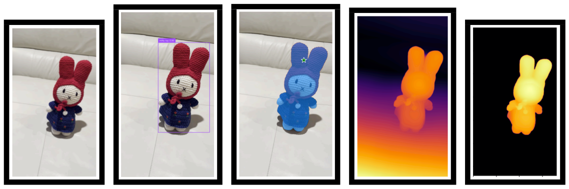

Figure 3. Combined methods: original image, object detection, segmentation mask, depth map, masked depth map

However, our camera parameter estimations from COLMAP and Fast3R are less accurate, which affects the fidelity of the reconstructed 3D point cloud.

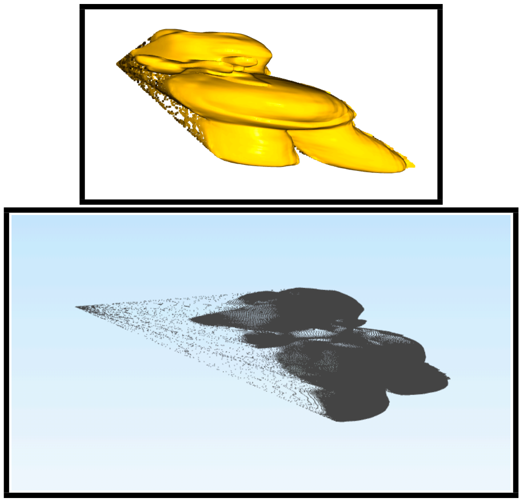

Figure 4. Combined method produced Miffy PC

DUSt3R demonstrates accurate correspondence between the two input images.

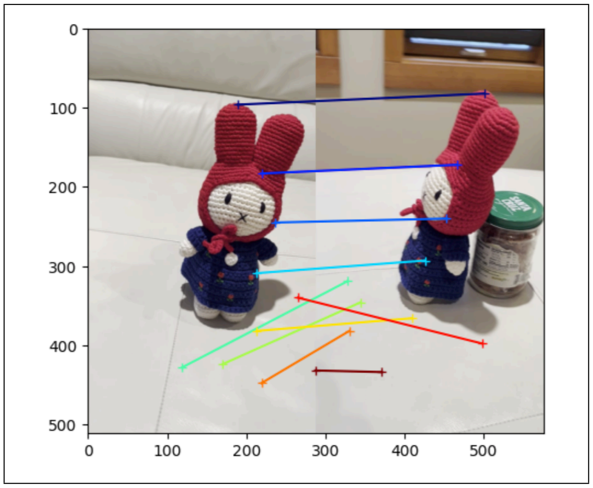

Figure 5. DUSt3R shows image correspondence.

DUSt3R produces less well-aligned point clouds when given 5 input images. We note as future work that post-processing techniques may help address some of these limitations.

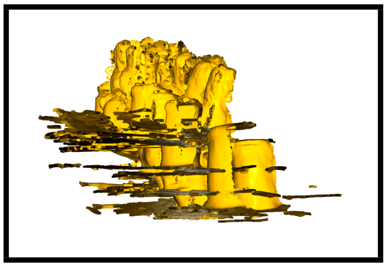

Figure 6. Dust3r for Miffy: using 5 images

FAST3R demonstrates a decent reconstruction model at a 75% confidence level.

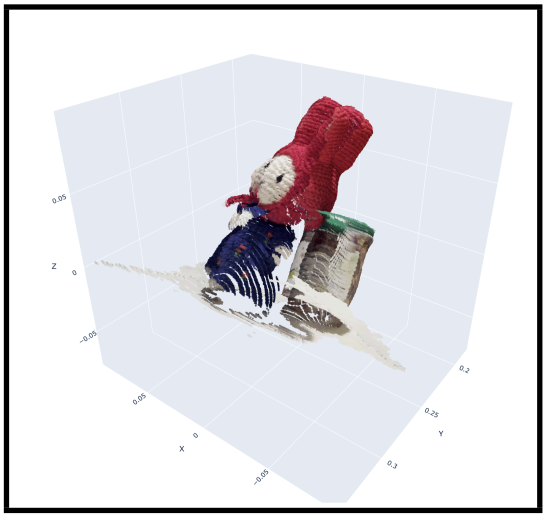

Figure 7. Fast3r for Miffy

VGGT produces the following depth map, which is visually satisfactory.

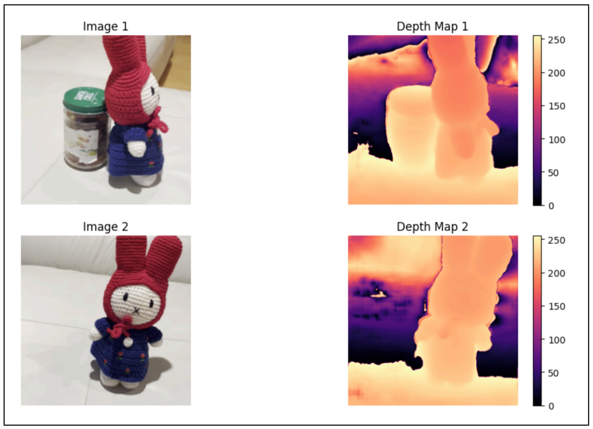

Figure 8. VGGT produced depth map, based on 5 images.

VGGT provides a point cloud for the scene. Many background points have low confidence levels.

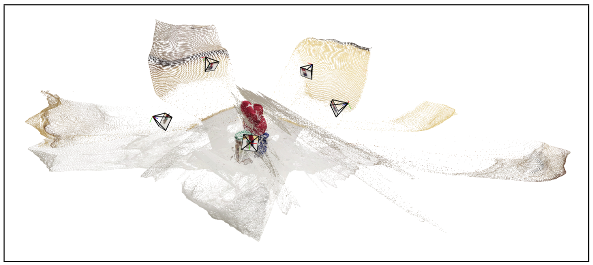

Figure 9. VGGT produced point cloud model for the scene, based on 5 images.

VGGT produces a clean point cloud model at a high confidence level of 72%. Five frustums indicate the poses of the input images.

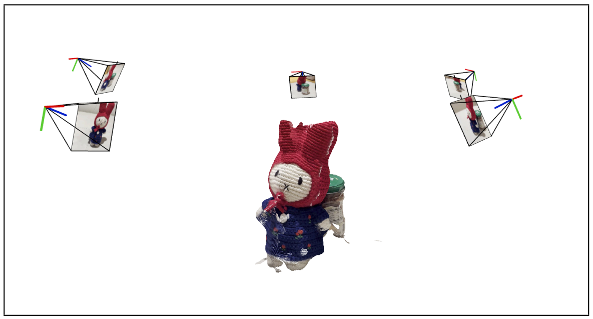

Figure 10. VGGT produced point cloud model for the miffy object, based on 5 images.

The point cloud results are presented in the following order: Colmap, VGGT, FAST3R, and DUST3R. Both VGGT and FAST3R utilize 5 input images, whereas DUST3R uses 2 input images.

- VGGT result is the best, in terms of completeness and accuracy
- Fast3R shows artifacts, and missing cloud points
- Dust3R miss lots of points

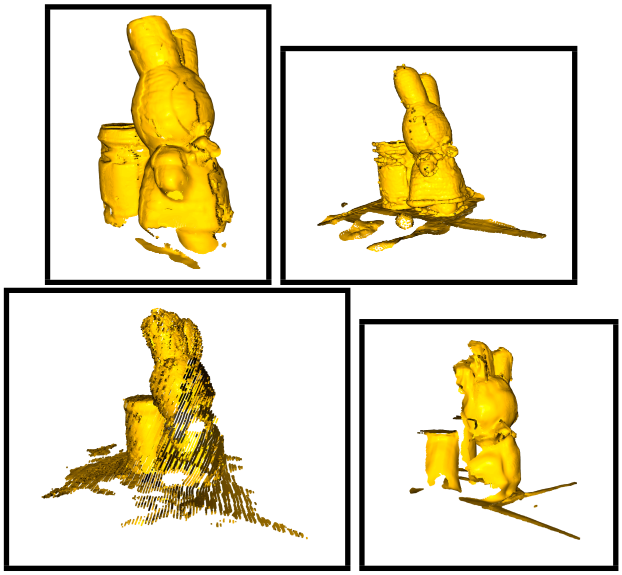

Figure 11. CD results

5.2. 3D PC model via generative model

5.2.1 Quantitative result

Model size is measured by the number of parameters.

	Model size	Inference time
Point-E	80M	5min(5) 2min(1)
LRM	260M	1m2s (1 img)

5.2.2 Qualitative result

3D-LMNet is trained on the ShapeNet dataset, which primarily consists of office utensils. An object like the Miffy toy is therefore out of its training distribution. With our limited training—conducted for 1 epoch and 2 epochs respectively—we observe the following changes in the image-conditioned generation of the point cloud.

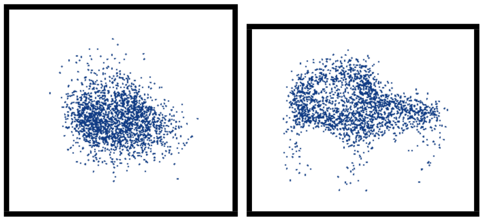

Figure 12. 3D-LMNet resulting PC

Point-E model produces the following point cloud when given a single input image.

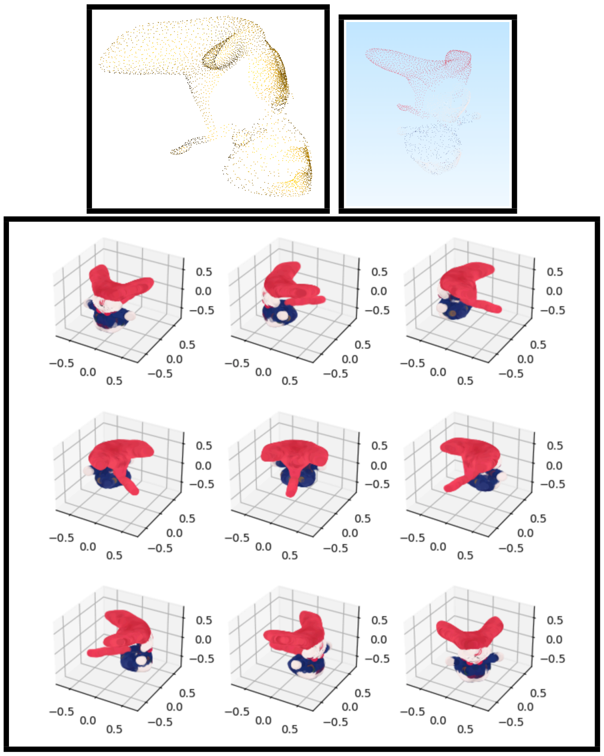

Figure 13. Point-E model produced PC with 1 input image

Point-E model produces this point cloud given 3 input images (5 input images cause GPU out-of-memory error).

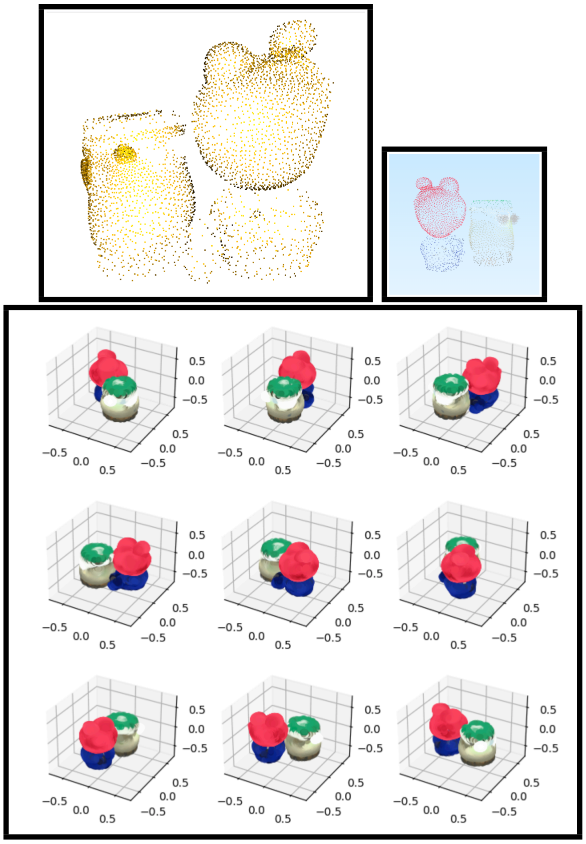

Figure 14. Point-E model produced PC with 3 input image

LRM model accepts only a single input image. It generates both a NeRF representation and a mesh model.

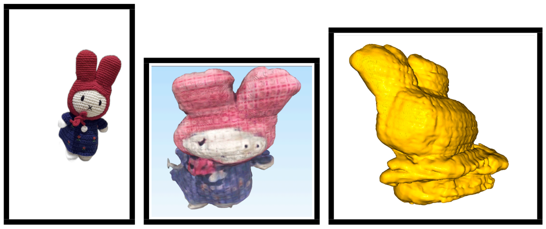

Figure 15. OpenLRM result: input image with white background, mesh model

5.3. Impact of number of input images in VGGT model

5.3.1 Inference time

Inference time grows as $O(n \log n)$ with the number of input images.

Number of Images (n)	Processing Time (s)
1	23.8
3	30.0
5	54.8
8	114
10	152
15	282
20	415

Table 3. Processing time vs number of images

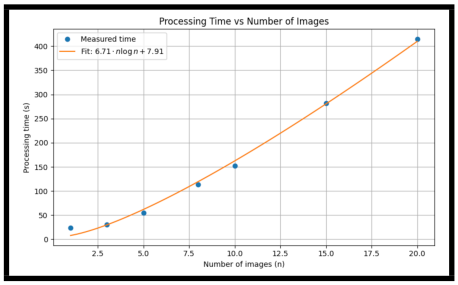

Figure 16. Inference time vs. number of input images for VGGT. The observed runtime increases approximately as $\mathcal{O}(N \log N)$, indicating efficient scalability as more views are aggregated during reconstruction.

We hypothesize that the use of FlashAttention enables this improved computational complexity.

$$\text{Time}(n) \approx 6.71 \cdot n \log n + 7.91$$

5.3.2 Reconstruction quality

Our results indicate that increasing the number of input images leads to more accurate 3D reconstructions. We measure the latency in seconds on Apple M1 machine, using CPU.

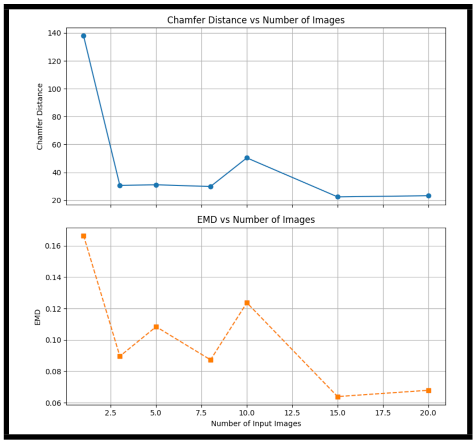

Figure 17. This shows the relationship between the number of input images and the quality of reconstructed point clouds using VGGT. As the number of images increases from 1 to 20, both Chamfer Distance and EMD generally decrease, indicating improved reconstruction quality. Although some fluctuations exist (e.g., at 10 images), the overall trend suggests that more input views provide better geometric fidelity.

The distances decrease in general, when we have more input images.

Number of Images	Chamfer Distance	EMD
1	137.92	0.1665
3	30.63	0.0896
5	31.08	0.1084
8	29.85	0.0873
10	50.39	0.1237
15	22.40	0.0639
20	23.24	0.0679

Table 4. VGGT reconstruction quality vs. number of input images

5.4. Attention weight in VGGT model

We call our visualization tool the VGGT Viewer. On our machine (Mac M1), we find that it can handle at most three layers of global attention masks before running out of memory.

When comparing the first three and last three layers, we observe subtle differences in the attention patterns. For instance, in the first three layers (top panel), consider the comparison between image 3 in the second row, third column and the fourth row, fourth column—there is a noticeable difference in attention around the lower-right region of the toy. Between the two layer sets, these differences become even more pronounced, highlighting distinct attention behavior across layers.

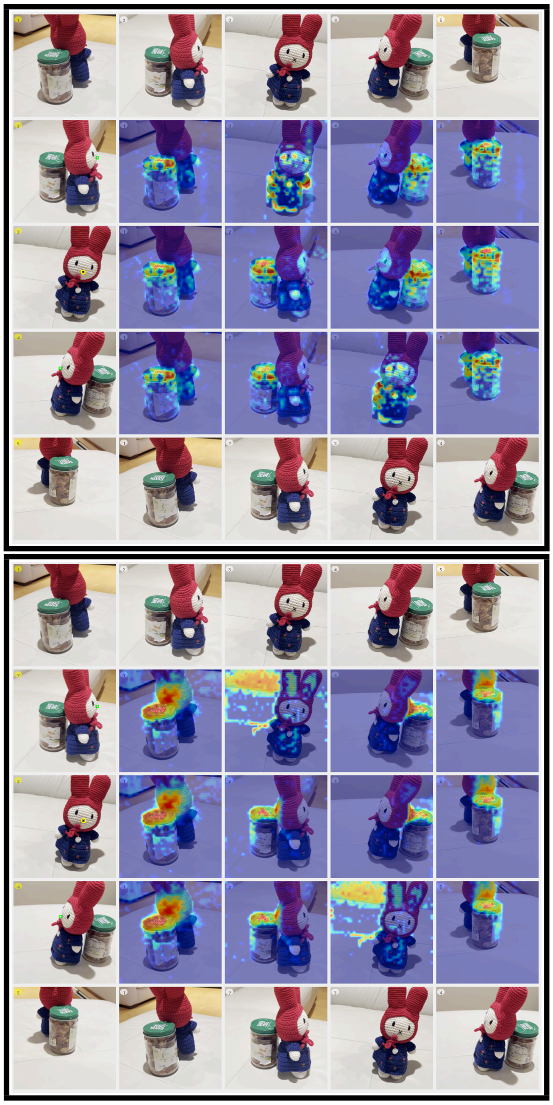

Figure 18. VGGT global level attention where we click the nose pixel of the miffy object: top is for first 3 layers, and bottom is for last 3 layers

6. Conclusion

6.1. Comparison of approaches

Running the real-world ground truth data point through all these models provides evidence that:

- Discriminative models excel in accuracy and speed when given sufficient input.
- Generative models produce uncertain yet diverse outputs, conditioned on input image(s).
- Within discriminative models:
  - DUST3R achieves adequate Chamfer Distance (CD) with 2-image input but produces

unusable results with more than 2 images unless post-global processing is applied.

  - Fast3R yields significantly better CD with 2+ images, outperforming DUST3R slightly.
  - VGGT delivers high-fidelity CD for 2+ input images without compromising inference speed.

- Within generative models:
  - 3D-LMNET effectively generates similar point clouds within categories represented in the ShapeNet dataset.
  - Point-E can utilize multiple images as prior and produces scenes increasingly resembling the inputs as the number of images increases.
  - LRM accepts a single input image and produces an adequate mesh result at a speed comparable to VGGT, while using only one-sixth of the model size.

We also demonstrate that:

- The 3D-LMNET model can be modified to enhance both its image feature extractor and PointNet encoder expressiveness.
- It is possible to extract the global attention distribution values at inference time and visualize them to understand the relative impact of different input images on the generation of a single 3D point.

6.2. Further study

These directions merit further exploration:

- More precise metric calculation
  - Utilize object masks to accurately define the ground truth object model, potentially supplementing confidence scores for selecting point cloud segments belonging to the target object.
- Deeper analysis of VGGT
  - Upon availability of VGGT training code, apply methods such as Grad-CAM for regression tasks to investigate the alignment mechanisms within VGGT.
  - Leverage VGGT's tracking head to identify corresponding 2D points for given 3D points on the point cloud and study their relationship with Grad-CAM results.
- Further tuning of the 3D-LMNET model
  - Due to limited GCP credits and time constraints in setting up AWS V100, completing training and benchmarking our modified 3D-LMNET against the original version remains an important future project to provide insights into model design choices.

7. Appendices

7.1. Video

LRM generated mesh video and VGGT generated PLY file are shared in this Google drive public folder. You can use a web based PLY viewer like here to view PLY file.

7.2. List of code

Code content

```
Step01-co3d-view.ipynb
explore CO3D dataset

Step02-dust3r-inference.ipynb
run dust3r inference on 2 and 5 images

Step03-fast3r.ipynb
run fast3r inference on 5 images

Step04-vggt.ipynb
run vggt inference on 5 images

Step05-colmap.ipynb
calculate Chamfer distance and EMD

Step06-ShapE.ipynb
shapE inference, text to model

Step07-3d-lmnet-dino-dgcnn.ipynb
train, eval and inference on 1 image

Step08-GroundingDINO-SingleImage.ipynb
test GroundingDINO

Step09-SegmentAnything-predictor_example.ipynb
test SAM

Step10-DepthAnything-test-depth-anything.ipynb
test DepthAnything

Step11-Combined.ipynb
combine above 3 to infer 3D model from 1 input image

Step12-pointe-image2pointcloud.ipynb
infer 3D PC from 1 image

Extra code for VGGT modification:
demo_viser.py and attention.py

Extra code for DG-CNN and DINO integration to 3D-LMNET
plus training and 1 image conditioned 3D PC generation code.
```

8. Contribution and Acknowledgement

We would like to express our sincere gratitude to the authors of the following code projects, whose work enabled us to extend and build this valuable project.

1. Dust3r repo
2. Fast3r repo
3. VGGT repo
4. 3d-lmnet-pytorch repo
5. GroundingDINO repo
6. SAM repo

7. DepthAnything repo

8. Point-E repo

9. OpenLRM repo

We would also like to thank Tiange Xiang for his thoughtful comments, suggestions, and encouragement throughout this project.

References

[1] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4104–4113, 2016.

[2] A. Fisher, R. Cannizzaro, M. Cochrane, C. Nagahawatte, and J. L. Palmer, “Colmap: A memory-efficient occupancy grid mapping framework,” Robotics and Autonomous Systems, vol. 142, p. 103755, 2021.

[3] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al., “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” in European Conference on Computer Vision, pp. 38–55, Springer, 2024.

[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al., “Segment anything,” in Proceedings of the IEEE/CVF international conference on computer vision, pp. 4015–4026, 2023.

[5] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, “Depth anything: Unleashing the power of large-scale unlabeled data,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10371–10381, 2024.

[6] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, “Dust3r: Geometric 3d vision made easy,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20697–20709, 2024.

[7] J. Yang, A. Sax, K. J. Liang, M. Henaff, H. Tang, A. Cao, J. Chai, F. Meier, and M. Feiszli, “Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass,” arXiv preprint arXiv:2501.13928, 2025.

[8] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “Vggt: Visual geometry grounded transformer,” arXiv preprint arXiv:2503.11651, 2025.

[9] P. Mandikal, K. Navaneet, M. Agarwal, and R. V. Babu, “3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image,” arXiv preprint arXiv:1807.07796, 2018.

[10] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, “Point-e: A system for generating 3d point clouds from complex prompts,” arXiv preprint arXiv:2212.08751, 2022.

[11] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, “Lrm: Large reconstruction model for single image to 3d,” arXiv preprint arXiv:2311.04400, 2023.

[12] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., “Shapenet: An information-rich 3d model repository,” arXiv preprint arXiv:1512.03012, 2015.

[13] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, “Dino: Detr with improved denoising anchor boxes for end-to-end object detection,” arXiv preprint arXiv:2203.03605, 2022.

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652–660, 2017.

[15] X.-Z. Xie, J.-W. Niu, X.-F. Liu, Q.-F. Li, Y. Wang, J. Han, and S. Tang, “Dg-cnn: Introducing margin information into convolutional neural networks for breast cancer diagnosis in ultrasound images,” Journal of Computer Science and Technology, vol. 37, no. 2, pp. 277–294, 2022.

[16] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, “Objaverse: A universe of annotated 3d objects,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 13142–13153, 2023.

[17] X. Yu, M. Xu, Y. Zhang, H. Liu, C. Ye, Y. Wu, Z. Yan, C. Zhu, Z. Xiong, T. Liang, et al., “Mvimgnet: A large-scale dataset of multi-view images,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9150–9161, 2023.

[18] L. Lebegue, E. Cazala-Hourcade, F. Languille, S. Artigues, and O. Melet, “Co3d, a worldwide one one-meter accuracy dem for 2025,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 43, pp. 299–304, 2020.

[19] Z. Teed and J. Deng, “Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras,” Advances in neural information processing systems, vol. 34, pp. 16558–16569, 2021.

[20] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16000–16009, 2022.

[21] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16901–16911, 2024.

[22] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14829–14838, 2022.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pp. 234–241, Springer, 2015.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440, 2015.

[25] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in Proceedings of the IEEE/CVF international conference on computer vision, pp. 12179–12188, 2021.

[26] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in Proceedings of the European conference on computer vision (ECCV), pp. 801–818, 2018.

[27] R. Birkl, D. Wofk, and M. Müller, "Midas v3. 1–a model zoo for robust monocular relative depth estimation," arXiv preprint arXiv:2307.14460, 2023.

[28] V. Guizilini, I. Vasiljevic, D. Chen, R. Ambruș, and A. Gaidon, "Towards zero-shot scale-aware monocular depth estimation," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9233–9243, 2023.

[29] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen, "Metric3d: Towards zero-shot metric 3d prediction from a single image," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9043–9053, 2023.

[30] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen, "Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[31] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7210–7219, 2021.

[32] Y. Bao, T. Ding, J. Huo, Y. Liu, Y. Li, W. Li, Y. Gao, and J. Luo, "3d gaussian splatting: Survey, technologies, challenges, and opportunities," IEEE Transactions on Circuits and Systems for Video Technology, 2025.

[33] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," in Proceedings of the IEEE/CVF international conference on computer vision, pp. 4541–4550, 2019.