VGGT: Visual Geometry Grounded Transformer

Jianyuan Wang[1,2] Minghao Chen[1,2] Nikita Karaev[1,2] Andrea Vedaldi[1,2]
Christian Rupprecht[1] David Novotny[2]

[1]Visual Geometry Group, University of Oxford [2]Meta AI

Figure 1. **VGGT** is a large feed-forward transformer with minimal 3D-inductive biases trained on a trove of 3D-annotated data. It accepts up to hundreds of images and predicts cameras, point maps, depth maps, and point tracks for all images at once in less than a second, which often outperforms optimization-based alternatives without further processing.

Abstract

We present VGGT, a feed-forward neural network that directly infers all key 3D attributes of a scene, including camera parameters, point maps, depth maps, and 3D point tracks, from one, a few, or hundreds of its views. This approach is a step forward in 3D computer vision, where models have typically been constrained to and specialized for single tasks. It is also simple and efficient, reconstructing images in under one second, and still outperforming alternatives that require post-processing with visual geometry optimization techniques. The network achieves state-of-the-art results in multiple 3D tasks, including camera parameter estimation, multi-view depth estimation, dense point cloud reconstruction, and 3D point tracking. We also show that using pretrained VGGT as a feature backbone significantly enhances downstream tasks, such as non-rigid point tracking and feed-forward novel view synthesis. Code and models are publicly available at https://github.com/facebookresearch/vggt.

1. Introduction

We consider the problem of estimating the 3D attributes of a scene, captured in a set of images, utilizing a feed-forward neural network. Traditionally, 3D reconstruction has been approached with visual-geometry methods, utilizing iterative optimization techniques like Bundle Adjustment (BA) [44]. Machine learning has often played an important complementary role, addressing tasks that cannot be solved by geometry alone, such as feature matching and monocular depth prediction. The integration has become increasingly tight, and now state-of-the-art Structure-from-Motion (SfM) methods like VGGSfM [124] combine machine learning and visual geometry end-to-end via differentiable BA. Even so, visual geometry *still* plays a major role in 3D reconstruction, which increases complexity and computational cost.

As networks become ever more powerful, we ask if, finally, 3D tasks can be solved *directly* by a neural network, eschewing geometry post-processing almost entirely. Recent contributions like DUSt3R [128] and its evolution

MASt3R [61] have shown promising results in this direction, but these networks can only process two images at once and rely on post-processing to reconstruct more images, fusing pairwise reconstructions.

In this paper, we take a further step towards removing the need to optimize 3D geometry in post-processing. We do so by introducing *Visual Geometry Grounded Transformer* (VGGT), a feed-forward neural network that performs 3D reconstruction from one, a few, or even hundreds of input views of a scene. VGGT predicts a full set of 3D attributes, including camera parameters, depth maps, point maps, and 3D point tracks. It does so in a single forward pass, in seconds. Remarkably, it often outperforms optimization-based alternatives even without further processing. This is a substantial departure from DUSt3R, MASt3R, or VGGSfM, which still require costly iterative post-optimization to obtain usable results.

We also show that it is unnecessary to design a special network for 3D reconstruction. Instead, VGGT is based on a fairly standard large transformer [118], with no particular 3D or other inductive biases (except for alternating between frame-wise and global attention), but trained on a large number of publicly available datasets with 3D annotations. VGGT is thus built in the same mold as large models for natural language processing and computer vision, such as GPTs [1, 28, 147], CLIP [85], DINO [9, 77], and Stable Diffusion [33]. These have emerged as versatile backbones that can be fine-tuned to solve new, specific tasks. Similarly, we show that the features computed by VGGT can significantly enhance downstream tasks like point tracking in dynamic videos, and novel view synthesis.

There are several recent examples of large 3D neural networks, including DepthAnything [141], MoGe [127], and LRM [48]. However, these models only focus on a single 3D task, such as monocular depth estimation or novel view synthesis. In contrast, VGGT uses a shared backbone to predict all 3D quantities of interest together. We demonstrate that *learning* to predict these interrelated 3D attributes enhances overall accuracy despite potential redundancies. At the same time, we show that, during *inference*, we can derive the point maps from separately predicted depth and camera parameters, obtaining better accuracy compared to directly using the dedicated point map head.

To summarize, we make the following contributions: (1) We introduce VGGT, a large feed-forward transformer that, given one, a few, or even hundreds of images of a scene, can predict all its key 3D attributes, including camera intrinsics and extrinsics, point maps, depth maps, and 3D point tracks, in seconds. (2) We demonstrate that VGGT's predictions are directly usable, being highly competitive and usually better than those of state-of-the-art methods that use slow post-processing optimization techniques. (3) We also show that, when further combined with BA post-processing, VGGT achieves state-of-the-art results across the board, even when compared to methods that specialize in a subset of 3D tasks, often improving quality substantially.

We make our code and models publicly available at https://github.com/facebookresearch/vggt. We believe that this will facilitate further research in this direction and benefit the computer vision community by providing a new foundation for fast, reliable, and versatile 3D reconstruction.

2. Related Work

Structure from Motion is a classic computer vision problem [44, 76, 79] that involves estimating camera parameters and reconstructing sparse point clouds from a set of images of a static scene captured from different viewpoints. The traditional SfM pipeline [2, 35, 69, 93, 102, 133] consists of multiple stages, including image matching, triangulation, and bundle adjustment. COLMAP [93] is the most popular framework based on the traditional pipeline. In recent years, deep learning has improved many components of the SfM pipeline, with keypoint detection [20, 30, 115, 148] and image matching [10, 66, 91, 98] being two primary areas of focus. Recent methods [5, 101, 108, 111, 112, 117, 121, 124, 130, 159] explored end-to-end differentiable SfM, where VGGSfM [124] started to outperform traditional algorithms on challenging phototourism scenarios.

Multi-view Stereo aims to densely reconstruct the geometry of a scene from multiple overlapping images, typically assuming known camera parameters, which are often estimated with SfM. MVS methods can be divided into three categories: traditional handcrafted [37, 38, 95, 129], global optimization [36, 73, 132, 146], and learning-based methods [41, 71, 83, 144, 156]. As in SfM, learning-based MVS approaches have recently seen a lot of progress. Here, DUSt3R [128] and MASt3R [61] directly estimate aligned dense point clouds from a pair of views, similar to MVS but without requiring camera parameters. Some concurrent works [110, 126, 140, 155] explore replacing DUSt3R's test-time optimization with neural networks, though these attempts achieve only suboptimal or comparable performance to DUSt3R. Instead, VGGT outperforms DUSt3R and MASt3R by a large margin.

Tracking-Any-Point was first introduced in Particle Video [90] and revived by PIPs [43] during the deep learning era, aiming to track points of interest across video sequences including dynamic motions. Given a video and some 2D query points, the task is to predict 2D correspondences of these points in all other frames. TAP-Vid [22] proposed three benchmarks for this task and a simple baseline method later improved in TAPIR [23]. CoTracker [54, 55] utilized correlations between different points to track through occlusions, while DOT [59] enabled dense tracking through occlusions. Recently, TAPTR [62] proposed

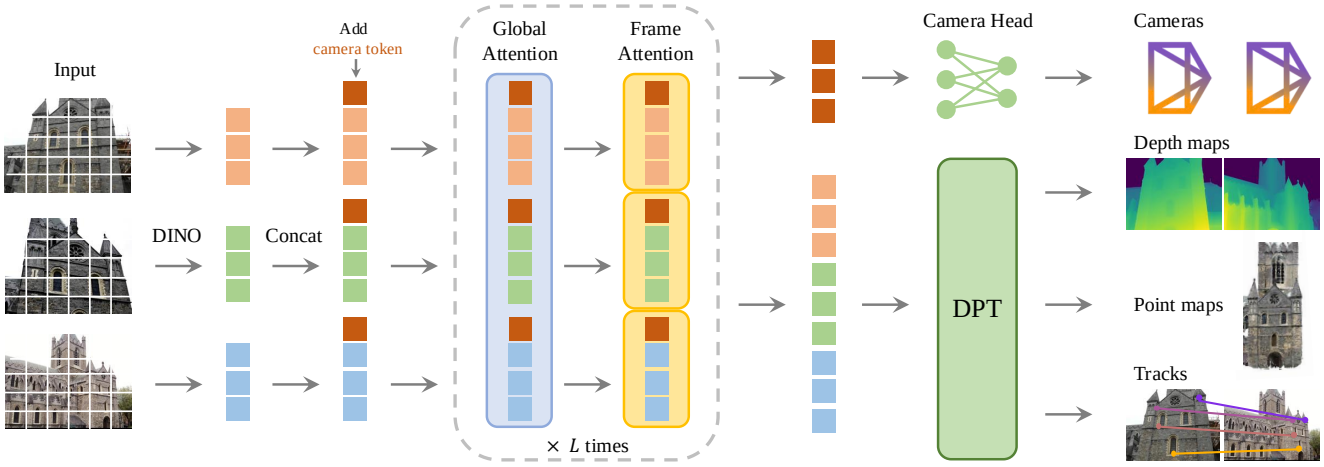

Figure 2. **Architecture Overview.** Our model first patchifies the input images into tokens by DINO, and appends camera tokens for camera prediction. It then alternates between frame-wise and global self attention layers. A camera head makes the final prediction for camera extrinsics and intrinsics, and a DPT [86] head for any dense output.

an end-to-end transformer for this task, and LocoTrack [12] extended commonly used pointwise features to nearby regions. All of these methods are specialized point trackers. Here, we demonstrate that VGGT's features yield state-of-the-art tracking performance when coupled with existing point trackers.

3. Method

We introduce VGGT, a large transformer that ingests a set of images as input and produces a variety of 3D quantities as output. We start by introducing the problem in Sec. 3.1, followed by our architecture in Sec. 3.2 and its prediction heads in Sec. 3.3, and finally the training setup in Sec. 3.4.

3.1. Problem definition and notation

The input is a sequence $(I_i)_{i=1}^N$ of N RGB images $I_i \in \mathbb{R}^{3\times H\times W}$, observing the same 3D scene. VGGT's transformer is a function that maps this sequence to a corresponding set of 3D annotations, one per frame:

$$f\left((I_i)_{i=1}^N\right) = (\mathbf{g}_i, D_i, P_i, T_i)_{i=1}^N . \quad (1)$$

The transformer thus maps each image I_i to its camera parameters $\mathbf{g}_i \in \mathbb{R}^9$ (intrinsics and extrinsics), its depth map $D_i \in \mathbb{R}^{H\times W}$, its point map $P_i \in \mathbb{R}^{3\times H\times W}$, and a grid $T_i \in \mathbb{R}^{C\times H\times W}$ of C-dimensional features for point tracking. We explain next how these are defined.

For the **camera parameters** $\mathbf{g}_i$, we use the parametrization from [124] and set $\mathbf{g} = [\mathbf{q}, \mathbf{t}, \mathbf{f}]$ which is the concatenation of the rotation quaternion $\mathbf{q} \in \mathbb{R}^4$, the translation vector $\mathbf{t} \in \mathbb{R}^3$, and the field of view $\mathbf{f} \in \mathbb{R}^2$. We assume that the camera's principal point is at the image center, which is common in SfM frameworks [94, 124].

We denote the domain of the image I_i with $\mathcal{I}(I_i) = \{1,\dots,H\}\times\{1,\dots,W\}$, *i.e.*, the set of pixel locations. The **depth map** D_i associates each pixel location $\mathbf{y} \in \mathcal{I}(I_i)$ with its corresponding depth value $D_i(\mathbf{y}) \in \mathbb{R}^+$, as observed from the i-th camera. Likewise, the **point map** P_i associates each pixel with its corresponding 3D scene point $P_i(\mathbf{y}) \in \mathbb{R}^3$. Importantly, like in DUSt3R [128], the point maps are *viewpoint invariant*, meaning that the 3D points $P_i(\mathbf{y})$ are defined in the coordinate system of the first camera $\mathbf{g}_1$, which we take as the world reference frame.

Finally, for **keypoint tracking**, we follow track-any-point methods such as [24, 56]. Namely, given a fixed query image point $\mathbf{y}_q$ in the query image I_q, the network outputs a track $\mathcal{T}^\star(\mathbf{y}_q) = (\mathbf{y}_i)_{i=1}^N$ formed by the corresponding 2D points $\mathbf{y}_i \in \mathbb{R}^2$ in all images I_i.

Note that the transformer f above does not output the tracks directly but instead features $T_i \in \mathbb{R}^{C\times H\times W}$, which are used for tracking. The tracking is delegated to a separate module, described in Sec. 3.3, which implements a function $\mathcal{T}((\mathbf{y}_j)_{j=1}^M, (T_i)_{i=1}^N) = ((\hat{\mathbf{y}}_{j,i})_{i=1}^N)_{j=1}^M$. It ingests the query point $\mathbf{y}_q$ and the dense tracking features T_i output by the transformer f and then computes the track. The two networks f and $\mathcal{T}$ are trained jointly end-to-end.

Order of predictions. The order of the images in the input sequence is arbitrary, except that the first image is chosen as the reference frame. The network architecture is designed to be permutation equivariant for all but the first frame.

Over-complete predictions. Notably, not all quantities predicted by VGGT are independent. For example, as shown by DUSt3R [128], the camera parameters $\mathbf{g}$ can be inferred from the invariant point map P, for instance, by solving the Perspective-n-Point (PnP) problem [34, 60].

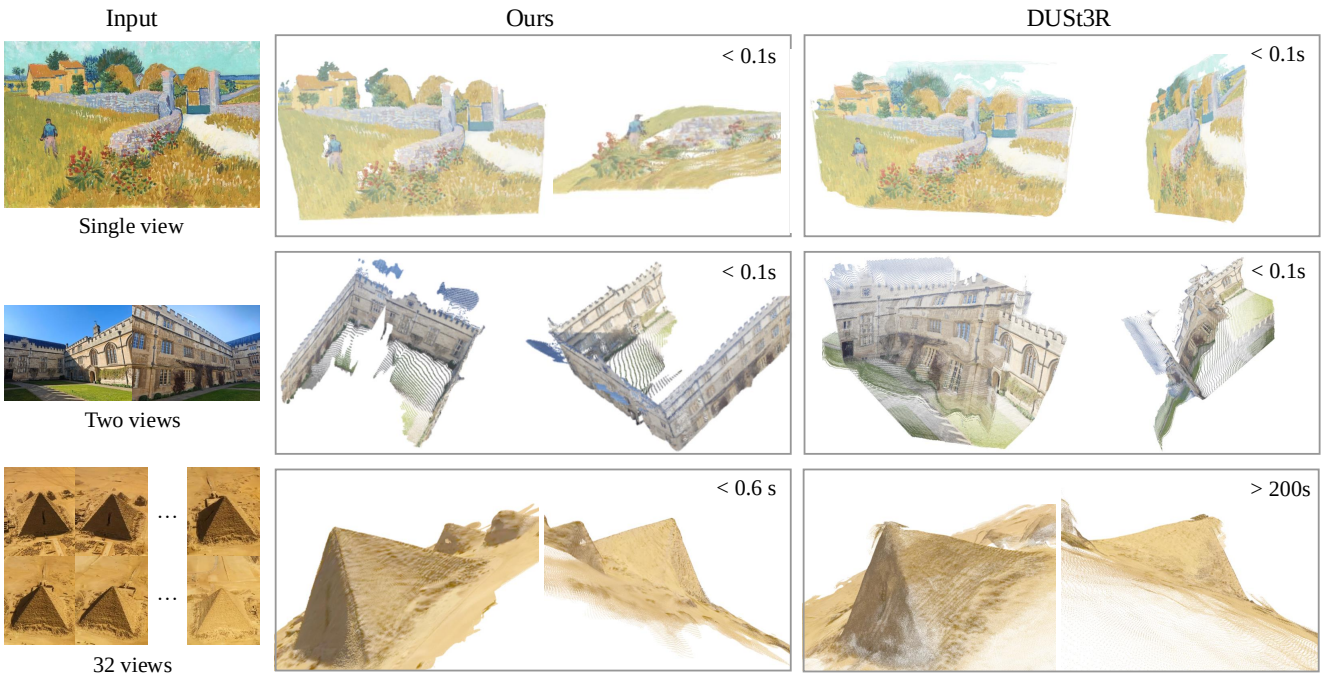

Figure 3. **Qualitative comparison of our predicted 3D points to DUSt3R on in-the-wild images.** As shown in the top row, our method successfully predicts the geometric structure of an oil painting, while DUSt3R predicts a slightly distorted plane. In the second row, our method correctly recovers a 3D scene from two images with no overlap, while DUSt3R fails. The third row provides a challenging example with repeated textures, while our prediction is still high-quality. We do not include examples with more than 32 frames, as DUSt3R runs out of memory beyond this limit.

Figure 4. **More Visualizations of Point Map Estimation.** From top left to the bottom right, our VGGT estimates point maps using 32, 24, 2 and 6 input images respectively, all in seconds.

Furthermore, the depth maps can be deduced from the point map and the camera parameters. However, as we show in Sec. 4.5, tasking VGGT with explicitly predicting *all* aforementioned quantities during training brings substantial performance gains, even when these are related by closed-form relationships. Meanwhile, during inference, it is observed that combining independently estimated depth maps and camera parameters produces more accurate 3D points compared to directly employing a specialized point map branch.

3.2. Feature Backbone

Following recent works in 3D deep learning [52, 128, 131], we design a simple architecture with minimal 3D inductive biases, letting the model learn from ample quantities of 3D-annotated data. In particular, we implement the model f as a large transformer [118]. To this end, each input image I is initially patchified into a set of K tokens[1] $\mathrm{t}^I \in \mathbb{R}^{K \times C}$ through DINO [77]. The combined set of image tokens from all frames, *i.e.*, $\mathrm{t}^I = \cup_{i=1}^N \{\mathrm{t}_i^I\}$, is subsequently processed through the main network structure, alternating frame-wise and global self-attention layers.

Alternating-Attention. We slightly adjust the standard transformer design by introducing Alternating-Attention (AA), making the transformer focus within each frame and globally in an alternate fashion. Specifically, *frame-wise self-attention* attends to the tokens t_k^I within each frame separately, and *global self-attention* attends to the tokens t^I across all frames jointly. This strikes a balance between integrating information across different images and normalizing the activations for the tokens within each image. By default, we employ $L = 24$ layers of global and frame-wise attention. In Sec. 4, we demonstrate that our AA architecture brings significant performance gains. Note that our architecture does not employ any cross-attention layers, only self-attention ones.

3.3. Prediction heads

Here, we describe how f predicts the camera parameters, depth maps, point maps, and point tracks. First, for each input image I_i, we augment the corresponding image tokens t_i^I with an additional camera token $\mathrm{t}_i^{\mathbf{g}} \in \mathbb{R}^{1 \times C'}$ and four register tokens [18] $\mathrm{t}_i^R \in \mathbb{R}^{4 \times C'}$. The concatenation of $(\mathrm{t}_i^I, \mathrm{t}_i^{\mathbf{g}}, \mathrm{t}_i^R j)_{i=1}^N$ is then passed to the AA transformer, yielding output tokens $(\hat{\mathrm{t}}_i^I, \hat{\mathrm{t}}_i^{\mathbf{g}}, \hat{\mathrm{t}}_i^R)_{i=1}^N$. Here, the camera token and register tokens of the first frame ($\mathrm{t}_1^{\mathbf{g}} := \bar{\mathrm{t}}^{\mathbf{g}}$, $\mathrm{t}_1^R := \bar{\mathrm{t}}^R$) are set to a different set of learnable tokens $\bar{\mathrm{t}}^{\mathbf{g}}, \bar{\mathrm{t}}^R$ than those of all other frames ($\mathrm{t}_i^{\mathbf{g}} := \bar{\bar{\mathrm{t}}}^{\mathbf{g}}, \mathrm{t}_i^R := \bar{\bar{\mathrm{t}}}^R, i \in [2, \ldots, N]$), which are also learnable. This allows the model to distinguish the first frame from the rest, and to represent the 3D predictions in the coordinate frame of the first camera. Note that the refined camera and register tokens now become frame-specific——this is because our AA transformer contains frame-wise self-attention layers that allow the transformer to match the camera and register tokens with the corresponding tokens from the same image. Following common practice, the output register tokens $\hat{\mathrm{t}}_i^R$ are discarded while $\hat{\mathrm{t}}_i^I, \hat{\mathrm{t}}_i^{\mathbf{g}}$ are used for prediction.

Coordinate frame. As noted above, we predict cameras, point maps, and depth maps in the coordinate frame of the first camera $\mathbf{g}_1$. As such, the camera extrinsics output for the first camera are set to the identity, *i.e.*, the first rotation quaternion is $\mathbf{q}_1 = [0, 0, 0, 1]$ and the first translation vector is $\mathbf{t}_1 = [0, 0, 0]$. Recall that the special camera and register tokens $\mathrm{t}_1^{\mathbf{g}} := \bar{\mathrm{t}}^{\mathbf{g}}, \mathrm{t}_1^R := \bar{\mathrm{t}}^R$ allow the transformer to identify the first camera.

Camera prediction. The camera parameters $(\hat{\mathbf{g}}^i)_{i=1}^N$ are predicted from the output camera tokens $(\hat{\mathrm{t}}_i^{\mathbf{g}})_{i=1}^N$ using four additional self-attention layers followed by a linear layer. This forms the *camera head* that predicts the camera intrinsics and extrinsics.

Dense predictions. The output image tokens $\hat{\mathrm{t}}_i^I$ are used to predict the dense outputs, *i.e.*, the depth maps D_i, point maps P_i, and tracking features T_i. More specifically, $\hat{\mathrm{t}}_i^I$ are first converted to dense feature maps $F_i \in \mathbb{R}^{C'' \times H \times W}$ with a DPT layer [86]. Each F_i is then mapped with a 3×3 convolutional layer to the corresponding depth and point maps D_i and P_i. Additionally, the DPT head also outputs dense features $T_i \in \mathbb{R}^{C \times H \times W}$, which serve as input to the tracking head. We also predict the aleatoric uncertainty [57, 75] $\Sigma_i^D \in \mathbb{R}_+^{H \times W}$ and $\Sigma_i^P \in \mathbb{R}_+^{H \times W}$ for each depth and point map, respectively. As described in Sec. 3.4, the uncertainty maps are used in the loss and, after training, are proportional to the model's confidence in the predictions.

Tracking. In order to implement the tracking module $\mathcal{T}$, we use the CoTracker2 architecture [56], which takes the dense tracking features T_i as input. More specifically, given a query point $\mathbf{y}_j$ in a query image I_q (during training, we always set $q = 1$, but any other image can be potentially used as a query), the tracking head $\mathcal{T}$ predicts the set of 2D points $\mathcal{T}((\mathbf{y}_j)_{j=1}^M, (T_i)_{i=1}^N) = ((\hat{\mathbf{y}}_{j,i})_{i=1}^N)_{j=1}^M$ in all images I_i that correspond to the same 3D point as $\mathbf{y}$. To do so, the feature map T_q of the query image is first bilinearly sampled at the query point $\mathbf{y}_j$ to obtain its feature. This feature is then correlated with all other feature maps $T_i, i \neq q$ to obtain a set of correlation maps. These maps are then processed by self-attention layers to predict the final 2D points $\hat{\mathbf{y}}_i$, which are all in correspondence with $\mathbf{y}_j$. Note that, similar to VGGSfM [124], our tracker does not assume any temporal ordering of the input frames and, hence, can be applied to any set of input images, not just videos.

[1] The number of tokens depends on the image resolution.

3.4. Training

Training losses. We train the VGGT model f end-to-end using a multi-task loss:

$$\mathcal{L} = \mathcal{L}_{\text{camera}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pmap}} + \lambda \mathcal{L}_{\text{track}}. \tag{2}$$

We found that the camera ($\mathcal{L}_{\text{camera}}$), depth ($\mathcal{L}_{\text{depth}}$), and point-map ($\mathcal{L}_{\text{pmap}}$) losses have similar ranges and do not need to be weighted against each other. The tracking loss $\mathcal{L}_{\text{track}}$ is down-weighted with a factor of $\lambda = 0.05$. We describe each loss term in turn.

The camera loss $\mathcal{L}_{\text{camera}}$ supervises the cameras $\hat{\mathbf{g}}$: $\mathcal{L}_{\text{camera}} = \sum_{i=1}^{N} \|\hat{\mathbf{g}}_i - \mathbf{g}_i\|_\epsilon$, comparing the predicted cameras $\hat{\mathbf{g}}_i$ with the ground truth $\mathbf{g}_i$ using the Huber loss $|\cdot|_\epsilon$.

The depth loss $\mathcal{L}_{\text{depth}}$ follows DUSt3R [128] and implements the aleatoric-uncertainty loss [58, 74] weighing the discrepancy between the predicted depth $\hat{D}_i$ and the ground-truth depth D_i with the predicted uncertainty map $\hat{\Sigma}_i^D$. Differently from DUSt3R, we also apply a gradient-based term, which is widely used in monocular depth estimation. Hence, the depth loss is $\mathcal{L}_{\text{depth}} = \sum_{i=1}^{N} \|\Sigma_i^D \odot (\hat{D}_i - D_i)\| + \|\Sigma_i^D \odot (\nabla \hat{D}_i - \nabla D_i)\| - \alpha \log \Sigma_i^D$, where $\odot$ is the channel-broadcast element-wise product. The point map loss is defined analogously but with the point-map uncertainty Σ_i^P: $\mathcal{L}_{\text{pmap}} = \sum_{i=1}^{N} \|\Sigma_i^P \odot (\hat{P}_i - P_i)\| + \|\Sigma_i^P \odot (\nabla \hat{P}_i - \nabla P_i)\| - \alpha \log \Sigma_i^P$.

Finally, the tracking loss is given by $\mathcal{L}_{\text{track}} = \sum_{j=1}^{M} \sum_{i=1}^{N} \|\mathbf{y}_{j,i} - \hat{\mathbf{y}}_{j,i}\|$. Here, the outer sum runs over all ground-truth query points $\mathbf{y}_j$ in the query image I_q, $\mathbf{y}_{j,i}$ is $\mathbf{y}_j$'s ground-truth correspondence in image I_i, and $\hat{\mathbf{y}}_{j,i}$ is the corresponding prediction obtained by the application $\mathcal{T}((\mathbf{y}_j)_{j=1}^{M}, (T_i)_{i=1}^{N})$ of the tracking module. Additionally, following CoTracker2 [56], we apply a visibility loss (binary cross-entropy) to estimate whether a point is visible in a given frame.

Ground-truth coordinate normalization. If we scale a scene or change its global reference frame, the images of the scene are not affected at all, meaning that any such variant is a legitimate result of 3D reconstruction. We remove this ambiguity by normalizing the data, thus making a canonical choice and task the transformer to output this particular variant. We follow [128] and, first, express all quantities in the coordinate frame of the first camera $\mathbf{g}_1$. Then, we compute the average Euclidean distance of all 3D points in the point map P to the origin and use this scale to normalize the camera translations $\mathbf{t}$, the point map P, and the depth map D. Importantly, unlike [128], we do *not* apply such normalization to the predictions output by the transformer; instead, we force it to learn the normalization we choose from the training data.

Implementation Details. By default, we employ $L = 24$ layers of global and frame-wise attention, respectively. The model consists of approximately 1.2 billion parameters in total. We train the model by optimizing the training loss (2) with the AdamW optimizer for 160K iterations. We use a cosine learning rate scheduler with a peak learning rate of 0.0002 and a warmup of 8K iterations. For every batch, we randomly sample 2–24 frames from a random training scene. The input frames, depth maps, and point maps are resized to a maximum dimension of 518 pixels. The aspect ratio is randomized between 0.33 and 1.0. We also randomly apply color jittering, Gaussian blur, and grayscale augmentation to the frames. The training runs on 64 A100 GPUs over nine days. We employ gradient norm clipping with a threshold of 1.0 to ensure training stability. We leverage bfloat16 precision and gradient checkpointing to improve GPU memory and computational efficiency.

Training data. The model was trained using a large and diverse collection of datasets, including: Co3Dv2 [87], BlendMVS [145], DL3DV [68], MegaDepth [63], Kubric [40], WildRGB [134], ScanNet [17], HyperSim [88], Mapillary [70], Habitat [106], Replica [103], MVS-Synth [49], PointOdyssey [158], Virtual KITTI [7], Aria Synthetic Environments [81], Aria Digital Twin [81], and a synthetic dataset of artist-created assets similar to Objaverse [19]. These datasets span various domains, including indoor and outdoor environments, and encompass synthetic and real-world scenarios. The 3D annotations for these datasets are derived from multiple sources, such as direct sensor capture, synthetic engines, or SfM techniques [94]. The combination of our datasets is broadly comparable to those used by MASt3R [29] in size and diversity.

4. Experiments

This section compares our method against state-of-the-art approaches across multiple tasks to demonstrate its effectiveness.

4.1. Camera Pose Estimation

We first evaluate our method on the CO3Dv2 [87] and RealEstate10K [160] datasets for camera pose estimation, as shown in Tab. 1. Following [123], we randomly select 10 images per scene and evaluate them using the standard metrics and AUC@30, which combines RRA and RTA. RRA (Relative Rotation Accuracy) and RTA (Relative Translation Accuracy) calculate the relative angular errors in rotation and translation, respectively, for each image pair. These angular errors are then thresholded to determine the accuracy scores. AUC is the area under the accuracy-threshold curve of the minimum values between RRA and RTA across varying thresholds. The methods in Tab. 1 have been trained on Co3Dv2 (if learnable) but **never** trained on RealEstate10K. Our feed-forward model con-

Methods	Re10K *(unseen)* AUC@30 ↑	CO3Dv2 AUC@30 ↑	Time
Colmap+SPSG [91]	45.2	25.3	∼ 15s
PixSfM [65]	49.4	30.1	> 20s
PoseDiff [123]	48.0	66.5	∼ 7s
DUSt3R [128]	67.7	76.7	∼ 7s
MASt3R [61]	76.4	81.8	∼ 9s
VGGSfM v2 [124]	78.9	83.4	∼ 10s
MV-DUSt3R [110] ‡	71.3	69.5	∼ 0.6s
CUT3R [126] ‡	75.3	82.8	∼ 0.6s
FLARE [155] ‡	78.8	83.3	∼ 0.5s
Fast3R [140] ‡	72.7	82.5	∼ **0.2s**
Ours (Feed-Forward)	85.3	88.2	∼ **0.2s**
Ours (with BA)	**93.5**	**91.8**	∼ 1.8s

Table 1. **Camera Pose Estimation on RealEstate10K [160] and CO3Dv2 [87]** with 10 random frames. All metrics the higher the better. None of the methods were trained on the Re10K dataset. Runtime were measured using one H100 GPU. Methods marked with ‡ represent concurrent work.

Known GT camera	Method	Acc.↓	Comp.↓	Overall↓
✓	Gipuma [39]	**0.283**	0.873	0.578
✓	MVSNet [143]	0.396	0.527	0.462
✓	CIDER [138]	0.417	0.437	0.427
✓	PatchmatchNet [120]	0.427	0.377	0.417
✓	MASt3R [61]	0.403	0.344	0.374
✓	GeoMVSNet [156]	0.331	**0.259**	**0.295**
✗	DUSt3R [128]	2.677	0.805	1.741
✗	Ours	**0.389**	**0.374**	**0.382**

Table 2. **Dense MVS Estimation on the DTU [50] Dataset.** Methods operating with known ground-truth camera are in the top part of the table, while the bottom part contains the methods that do not know the ground-truth camera.

Methods	Acc.↓	Comp.↓	Overall↓	Time
DUSt3R	1.167	0.842	1.005	∼ 7s
MASt3R	0.968	0.684	0.826	∼ 9s
Ours (Point)	0.901	0.518	0.709	∼ 0.2s
Ours (Depth + Cam)	**0.873**	**0.482**	**0.677**	∼ 0.2s

Table 3. **Point Map Estimation on ETH3D [96].** DUSt3R and MASt3R use global alignment while ours is feed-forward and, hence, much faster. The row *Ours (Point)* indicates the results using the point map head directly, while *Ours (Depth + Cam)* denotes constructing point clouds from the depth map head combined with the camera head.

sistently outperforms competing methods across all metrics on both datasets, including those that employ computationally expensive post-optimization steps, such as Global Alignment for DUSt3R/MASt3R and Bundle Adjustment for VGGSfM, typically requiring more than 10 seconds. In contrast, VGGT *achieves superior performance while only operating in a feed-forward manner*, completing in merely 0.2 seconds. Compared to concurrent works [110, 126, 140, 155] (indicated by ‡), our method demonstrates significant performance advantages, with speed similar to the fastest variant Fast3R [140]. Furthermore, our model's performance advantage is even more pronounced on the RealEstate10K dataset, which none of the methods presented in Tab. 1 were trained on. This validates the superior generalization of VGGT.

Method	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
SuperGlue [91]	16.2	33.8	51.8
LoFTR [104]	22.1	40.8	57.6
DKM [31]	29.4	50.7	68.3
CasMTR [8]	27.1	47.0	64.4
Roma [32]	31.8	53.4	70.9
Ours	**33.9**	**55.2**	**73.4**

Table 4. **Two-View matching comparison on ScanNet-1500 [17, 91].** Although our tracking head is not specialized for the two-view setting, it outperforms the state-of-the-art two-view matching method Roma. Measured in AUC (higher is better).

At the same time, our results show that VGGT can be improved even further by combining it with classical visual-geometry methods. Specifically, refining the predicted camera poses and tracks with bundle adjustment further improves accuracy. Note that our method directly predicts close-to-accurate point/depth maps, which can serve as a good initialization for BA. This eliminates the need for triangulation and iterative refinement of BA as in [124], making our approach significantly faster (only around 2 seconds even with BA). In summary, while the feed-forward mode of the VGGT outperforms all previous alternatives (whether they are feed-forward or not), there is still room for improvement since post-optimization still brings benefits.

4.2. Multi-view Depth Estimation

Following MASt3R [61], we further evaluate our multi-view depth estimation results on the DTU [50] dataset. We report the standard DTU metrics, including Accuracy (the smallest Euclidean distance from the prediction to ground truth), Completeness (the smallest Euclidean distance from the ground truth to prediction), and their average Overall (*i.e.*, Chamfer distance). In Tab. 2, DUSt3R and our VGGT are the only two methods operating without the knowledge of ground truth cameras. MASt3R derives depth maps by triangulating matches using the ground truth camera parameters. Meanwhile, deep multi-view stereo (MVS) methods like GeoMVSNet use ground truth cameras to construct cost volumes.

Our method substantially outperforms DUSt3R, reducing the Overall score from 1.741 to 0.382. More importantly, it achieves results comparable to methods that know

Figure 5. **Visualization of Rigid and Dynamic Point Tracking.** Top: VGGT's tracking module $\mathcal{T}$ outputs keypoint tracks for an unordered set of input images depicting a static scene. Bottom: We finetune the backbone of VGGT to enhance a dynamic point tracker CoTracker [55], which processes sequential inputs.

ground-truth cameras at test time. The significant performance gains can likely be attributed to our model's multi-image training scheme that teaches the model to reason about multi-view triangulation natively, instead of relying on adhoc alignment procedures, such as in DUSt3R, which only averages multiple pairwise camera triangulations.

4.3. Point Map Estimation

We also compare the accuracy of our predicted point cloud to DUSt3R and MASt3R on the ETH3D [96] dataset. For each scene, we randomly sample 10 frames. The predicted point cloud is aligned to the ground truth using the Umeyama [116] algorithm. The results are reported after filtering out invalid points using the official masks. We report Accuracy, Completeness, and Overall (Chamfer distance) for point map estimation. As shown in Tab. 3, although DUSt3R and MASt3R conduct time-consuming post-optimization (global alignment—around 10 seconds per scene), our method still outperforms them significantly in a simple feed-forward regime at only 0.2 seconds per reconstruction.

Meanwhile, compared to directly using our estimated point maps, we found that leveraging predictions from our depth and camera heads (*i.e.*, unprojecting depth maps to 3D using camera parameters) yields higher accuracy. We attribute this improvement to the benefits of decomposing a complex task (point map estimation) into simpler subproblems (depth map and camera prediction), even though camera, depth maps, and point maps are jointly supervised during training.

We present a qualitative comparison with DUSt3R on in-the-wild scenes in Fig. 3 and further examples in Fig. 4. VGGT demonstrates high-quality predictions and strong generalization ability. It excels in challenging situations, including out-of-domain oil paintings, non-overlapping frames, and scenes with repeating textures like deserts.

ETH3D Dataset	Acc.↓	Comp.↓	Overall↓
Cross-Attention	1.287	0.835	1.061
Global Self-Attention Only	<u>1.032</u>	<u>0.621</u>	<u>0.827</u>
Alternating-Attention	**0.901**	**0.518**	**0.709**

Table 5. **Ablation Study for Transformer Backbone** on ETH3D. We compare our alternating-attention architecture against two variants: one using only global self-attention and another employing cross-attention.

4.4. Image Matching

Two-view image matching is a widely-explored topic [67, 92, 104] in computer vision. It represents a specific case of rigid point tracking, which is restricted to only two views, and hence a suitable evaluation benchmark to measure our tracking accuracy, even though our model is not specialized for this task. We follow the standard protocol [32, 92] on the ScanNet dataset [17] and report the results in Tab. 4. For each image pair, we extract the matches and use them to estimate an essential matrix, which is then decomposed to a relative camera pose. The final metric is the relative pose accuracy, measured by AUC. For evaluation, we use ALIKED [157] to detect keypoints, treating them as query points $\mathbf{y}_q$. These are then passed to our tracking branch $\mathcal{T}$ to find correspondences in the second frame. We adopt the evaluation hyperparameters (*e.g.*, the number of matches, RANSAC thresholds) from Roma [32]. Despite not being explicitly trained for two-view matching, Tab. 4 shows that VGGT achieves the highest accuracy among all baselines.

4.5. Ablation Studies

Feature Backbone We first validate the effectiveness of our proposed Alternating-Attention design by comparing it against two alternative attention architectures: (a) *global self-attention only*, and (b) *cross-attention*. To ensure a fair comparison, all model variants maintain an identical

w. $\mathcal{L}_{\text{camera}}$	w. $\mathcal{L}_{\text{depth}}$	w. $\mathcal{L}_{\text{track}}$	Acc.↓	Comp.↓	Overall↓
✗	✓	✓	1.042	0.627	0.834
✓	✗	✓	0.920	0.534	0.727
✓	✓	✗	0.976	0.603	0.790
✓	✓	✓	**0.901**	**0.518**	**0.709**

Table 6. **Ablation Study for Multi-task Learning**, which shows that simultaneous training with camera, depth and track estimation yields the highest accuracy in point map estimation on ETH3D.

number of parameters, using a total of $2L$ attention layers. For the cross-attention variant, each frame independently attends to tokens from all other frames, maximizing cross-frame information fusion although significantly increasing the runtime, particularly as the number of input frames grows. The hyperparameters such as the hidden dimension and the number of heads are kept the same. Point map estimation accuracy is chosen as the evaluation metric for our ablation study, as it reflects the model's joint understanding of scene geometry and camera parameters. Results in Tab. 5 demonstrate that our Alternating-Attention architecture outperforms both baseline variants by a clear margin. Additionally, our other preliminary exploratory experiments consistently showed that architectures using cross-attention generally underperform compared to those exclusively employing self-attention.

Multi-task Learning We also verify the benefit of training a single network to simultaneously learn multiple 3D quantities, even though these outputs may potentially overlap (*e.g.*, depth maps and camera parameters together can produce point maps). As shown in Tab. 6, there is a noticeable decrease in the accuracy of point map estimation when training without camera, depth, or track estimation. Notably, incorporating camera parameter estimation clearly enhances point map accuracy, whereas depth estimation contributes only marginal improvements.

4.6. Finetuning for Downstream Tasks

We now show that the VGGT pre-trained feature extractor can be reused in downstream tasks. We show this for feed-forward novel view synthesis and dynamic point tracking.

Method	Known Input Cam	Size	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [109]	✓	256	21.44	0.832	0.122
GS-LRM [153]	✓	256	29.59	0.944	0.051
LVSM [52]	✓	256	31.71	0.957	0.027
Ours-NVS*	✗	224	30.41	0.949	0.033

Table 7. **Quantitative comparisons for view synthesis on GSO [27] dataset.** Finetuning VGGT for feed-forward novel view synthesis, it demonstrates competitive performance even without knowing camera extrinsic and intrinsic parameters for the input images. Note that * indicates using a small training set (only 20%).

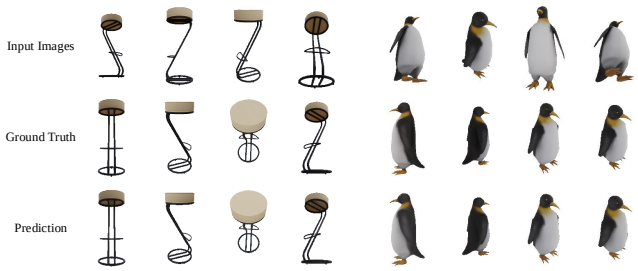

Figure 6. **Qualitative Examples of Novel View Synthesis.** The top row shows the input images, the middle row displays the ground truth images from target viewpoints, and the bottom row presents our synthesized images.

Feed-forward Novel View Synthesis is progressing rapidly [42, 48, 52, 107, 125, 139, 154]. Most existing methods take images with known camera parameters as input and predict the target image corresponding to a new camera viewpoint. Instead of relying on an explicit 3D representation, we follow LVSM [52] and modify VGGT to *directly* output the target image. Differently from them, however, we *do not* assume known camera parameters for the input frames.

We follow the training and evaluation protocol of LVSM closely, *e.g.*, using 4 input views and adopting Plücker rays to represent target viewpoints. The modification of VGGT is kept simple. The input images are converted into tokens by DINO, consistent with the original method. For target views, we apply a simple convolutional layer to encode their Plücker rays into tokens. These tokens, representing both the input images and target views, are concatenated and processed through AA transformer. Subsequently, a DPT head is employed to regress colors. It is important to note that our model does not encode Plücker rays for the input images, thus never explicitly receiving camera parameters for these input frames.

LVSM was trained on the Objaverse dataset [19], whereas we use a similar internal dataset of approximately 20% the size of Objaverese. Further training details and evaluation metrics can be found in [52]. As illustrated in Tab. 7, despite not utilizing input frame camera parameters, our model achieves competitive results on the GSO dataset [27]. We also hope to emphasize that this comparison is not strictly direct—further improvements are expected for our model if trained with a larger dataset. Our qualitative examples are shown in Fig. 6.

Dynamic Point Tracking. Video point tracking has emerged as a highly competitive task in recent years [24, 43, 56, 135], and it serves as another downstream application for our learned features. Following standard practices, we report these point-tracking metrics: Occlusion Accuracy (OA), which comprises the binary accuracy of occlusion

Method	Kinetics			RGB-S			DAVIS		
	AJ	δ^{vis}_{avg}	OA	AJ	δ^{vis}_{avg}	OA	AJ	δ^{vis}_{avg}	OA
TAPTR [62]	49.0	64.4	85.2	60.8	76.2	87.0	63.0	76.1	91.1
LocoTrack [12]	52.9	66.8	85.3	69.7	83.2	89.5	62.9	75.3	87.2
BootsTAPIR [25]	54.6	68.4	86.5	70.8	83.0	89.9	61.4	73.6	88.7
CoTracker [55]	49.6	64.3	83.3	67.4	78.9	85.2	61.8	76.1	88.3
CoTracker + Ours	**56.1**	**68.9**	**88.7**	**71.5**	**83.9**	**91.4**	**64.1**	**77.2**	**90.5**

Table 8. **Dynamic Point Tracking Results on the TAP-Vid benchmarks.** Although our model was not designed for dynamic scenes, simply fine-tuning CoTracker with our pretrained weights significantly enhances performance, demonstrating the robustness and effectiveness of our learned features.

predictions; δ^{vis}_{avg}, comprising the mean proportion of visible points accurately tracked within a certain pixel threshold; and Average Jaccard (AJ), measuring tracking and occlusion prediction accuracy simultaneously.

We adapt the state-of-the-art CoTracker2 model [56] by substituting its backbone with our pretrained feature backbone. This is necessary because VGGT is trained on unordered image collections instead of sequential videos. Our backbone predicts the tracking features T_i, which replace the outputs of the feature extractor and later enter the rest of the CoTracker2 architecture, that finally predicts the tracks. We finetune the entire modified tracker on Kubric [40]. As illustrated in Tab. 8, the integration of pretrained VGGT significantly enhances CoTracker's performance on the TAP-Vid benchmark [22]. For instance, VGGT's tracking features improve the δ^{vis}_{avg} metric from 78.9 to 83.9 on the TAP-Vid RGB-S dataset. Despite the TAP-Vid benchmark's inclusion of videos featuring rapid dynamic motions from various data sources, our model's strong performance demonstrates the generalization capability of its features, even in scenarios for which it was not explicitly designed.

5. Discussions

Limitations. While our method exhibits strong generalization to diverse in-the-wild scenes, several limitations remain. First, the current model does not support fisheye or panoramic images. Additionally, reconstruction performance drops under conditions involving extreme input rotations. Moreover, although our model handles scenes with minor non-rigid motions, it fails in scenarios involving substantial non-rigid deformation.

However, an important advantage of our approach is its flexibility and ease of adaptation. Addressing these limitations can be straightforwardly achieved by fine-tuning the model on targeted datasets with minimal architectural modifications. This adaptability clearly distinguishes our method from existing approaches, which typically require extensive re-engineering during test-time optimization to accommodate such specialized scenarios.

Input Frames	1	2	4	8	10	20	50	100	200
Time (s)	0.04	0.05	0.07	0.11	0.14	0.31	1.04	3.12	8.75
Mem. (GB)	1.88	2.07	2.45	3.23	3.63	5.58	11.41	21.15	40.63

Table 9. **Runtime and peak GPU memory usage across different numbers of input frames.** Runtime is measured in seconds, and GPU memory usage is reported in gigabytes.

Runtime and Memory. As shown in Tab. 9, we evaluate inference runtime and peak GPU memory usage of the feature backbone when processing varying numbers of input frames. Measurements are conducted using a single NVIDIA H100 GPU with flash attention v3 [97]. Images have a resolution of 336×518. We focus on the cost associated with the feature backbone since users may select different branch combinations depending on their specific requirements and available resources. The camera head is lightweight, typically accounting for approximately 5% of the runtime and about 2% of the GPU memory used by the feature backbone. A DPT head uses an average of 0.03 seconds and 0.2 GB GPU memory per frame. When GPU memory is sufficient, multiple frames can be processed efficiently in a single forward pass. At the same time, in our model, inter-frame relationships are handled only within the feature backbone, and the DPT heads make independent predictions per frame. Therefore, users constrained by GPU resources may perform predictions frame by frame. We leave this trade-off to the user's discretion.

We recognize that a naive implementation of global self-attention can be highly memory-intensive with a large number of tokens. Savings or accelerations can be achieved by employing techniques used in large language model (LLM) deployments. For instance, Fast3R [140] employs Tensor Parallelism to accelerate inference with multiple GPUs, which can be directly applied to our model.

Patchifying. As discussed in Sec. 3.2, we have explored the method of patchifying images into tokens by utilizing either a 14×14 convolutional layer or a pretrained DINOv2 model. Empirical results indicate that the DINOv2 model provides better performance; moreover, it ensures much more stable training, particularly in the initial stages. The DINOv2 model is also less sensitive to variations in hyperparameters such as learning rate or momentum. Consequently, we have chosen DINOv2 as the default method for patchifying in our model.

Differentiable BA. We also explored the idea of using differentiable bundle adjustment as in VGGSfM [124]. In small-scale preliminary experiments, differentiable BA demonstrated promising performance. However, a bottleneck is its computational cost during training. Enabling differentiable BA in PyTorch using Theseus [84] typically

makes each training step roughly 4 times slower, expensive for large-scale training. While customizing a framework to expedite training could be a potential solution, it falls outside the scope of this study. Thus, we opted not to include differentiable BA in this work, but we recognize it as a promising direction for large-scale unsupervised training, as it can serve as an effective supervision signal in scenarios lacking explicit 3D annotations.

Single-view Reconstruction. Unlike systems like DUSt3R and MASt3R that have to duplicate an image to create a pair, our model architecture inherently supports the input of a single image. In this case, global attention simply transitions to frame-wise attention. Although our model was not explicitly trained for single-view reconstruction, it demonstrates surprisingly good results. Some examples can be found in Fig. 3 and Fig. 7. We strongly encourage to try our demo for a more intuitive experience and better visualization.

Normalizing Prediction. As discussed in Sec. 3.4, our approach normalizes the ground truth using the average Euclidean distance of the 3D points. While some methods, such as DUSt3R, also apply such normalization to network predictions, our findings suggest that it is neither necessary for convergence nor advantageous for final model performance. Furthermore, it tends to introduce additional instability during the training phase.

6. Conclusions

We present Visual Geometry Grounded Transformer (VGGT), a feed-forward neural network that can directly estimate all key 3D scene properties for hundreds of input views. It achieves state-of-the-art results in multiple 3D tasks, including camera parameter estimation, multi-view depth estimation, dense point cloud reconstruction, and 3D point tracking. Our simple, neural-first approach departs from traditional visual geometry-based methods, which rely on optimization and post-processing to obtain accurate and task-specific results. The simplicity and efficiency of our approach make it well-suited for real-time applications, which is another benefit over optimization-based approaches.

Appendix

In the supplementary material, we provide the following:

- formal definitions of key terms in Appendix A.
- comprehensive implementation details, including architecture and training hyperparameters in Appendix B.
- additional experiments and discussions in Appendix C.
- qualitative examples of single-view reconstruction in Appendix D.
- an expanded review of related works in Appendix E.

A. Formal Definitions

In this section, we provide additional formal definitions that further ground the method section.

The camera extrinsics are defined in relation to the *world reference frame*, which we take to be the coordinate system of the first camera. We thus introduce two functions. The first function $\gamma(\mathbf{g}, \mathbf{p}) = \mathbf{p}'$ applies the rigid transformation encoded by $\mathbf{g}$ to a point $\mathbf{p}$ in the world reference frame to obtain the corresponding point $\mathbf{p}'$ in the camera reference frame. The second function $\pi(\mathbf{g}, \mathbf{p}) = \mathbf{y}$ further applies perspective projection, mapping the 3D point $\mathbf{p}$ to a 2D image point $\mathbf{y}$. We also denote the depth of the point as observed from the camera $\mathbf{g}$ by $\pi^{\mathrm{D}}(\mathbf{g}, \mathbf{p}) = d \in \mathbb{R}^+$.

We model the scene as a collection of regular surfaces $S_i \subset \mathbb{R}^3$. We make this a function of the i-th input image as the scene can change over time [150]. The depth at pixel location $\mathbf{y} \in \mathcal{I}(I_i)$ is defined as the minimum depth of any 3D point $\mathbf{p}$ in the scene that projects to $\mathbf{y}$, *i.e.*, $D_i(\mathbf{y}) = \min\{\pi^D(\mathbf{g}_i, \mathbf{p}) : \mathbf{p} \in S_i \ \wedge \ \pi(\mathbf{g}_i, \mathbf{p}) = \mathbf{y}\}$. The point at pixel location $\mathbf{y}$ is then given by $P_i(\mathbf{y}) = \gamma(\mathbf{g}, \mathbf{p})$, where $\mathbf{p} \in S_i$ is the 3D point that minimizes the expression above, *i.e.*, $\mathbf{p} \in S_i \ \wedge \ \pi(\mathbf{g}_i, \mathbf{p}) = \mathbf{y} \ \wedge \ \pi^D(\mathbf{g}_i, \mathbf{p}) = D_i(\mathbf{y})$.

B. Implementation Details

Architecture. As mentioned in the main paper, VGGT consists of 24 attention blocks, each block equipped with one frame-wise self-attention layer and one global self-attention layer. Following the ViT-L model used in DINOv2 [77], each attention layer is configured with a feature dimension of 1024 and employs 16 heads. We use the official implementation of the attention layer from PyTorch, *i.e.*, *torch.nn.MultiheadAttention*, with flash attention enabled. To stabilize training, we also use QKNorm [47] and LayerScale [114] for each attention layer. The value of LayerScale is initialized with 0.01. For image tokenization, we use DINOv2 [77] and add positional embedding. As in [142], we feed the tokens from the 4-th, 11-th, 17-th, and 23-rd block into DPT [86] for upsampling.

Training. To form a training batch, we first choose a random training dataset (each dataset has a different yet approximately similar weight, as in [128]), and from the

dataset, we then sample a random scene (uniformly). During the training phase, we select between 2 and 24 frames per scene while maintaining the constant total of 48 frames within each batch. For training, we use the respective training sets of each dataset. We exclude training sequences containing fewer than 24 frames. RGB frames, depth maps, and point maps are first isotropically resized, so the longer size has 518 pixels. Then, we crop the shorter dimension (around the principal point) to a size between 168 and 518 pixels while remaining a multiple of the 14-pixel patch size. It is worth mentioning that we apply aggressive color augmentation independently across each frame within the same scene, enhancing the model's robustness to varying lighting conditions. We build ground truth tracks following [32, 104, 124], which unprojects depth maps to 3D, reprojects points to target frames, and retains correspondences where reprojected depths match target depth maps. Frames with low similarity to the query frame are excluded during batch sampling. In rare cases with no valid correspondences, the tracking loss is omitted.

C. Additional Experiments

Camera Pose Estimation on IMC We also evaluate using the Image Matching Challenge (IMC) [53], a camera pose estimation benchmark focusing on phototourism data. Until recently, the benchmark was dominated by classical incremental SfM methods [93].

Baselines. We evaluate two flavors of our model: VGGT and VGGT + BA. VGGT directly outputs camera pose estimates, while VGGT + BA refines the estimates using an additional Bundle Adjustment stage. We compare to the classical incremental SfM methods such as [65, 93] and to recently-proposed deep methods. Specifically, recently VGGSfM [124] provided the first end-to-end trained deep method that outperformed incremental SfM on the challenging phototourism datasets.

Besides VGGSfM, we additionally compare to recently popularized DUSt3R [128] and MASt3R [61]. It is important to note that DUSt3R and MASt3R utilized a substantial portion of the MegaDepth dataset for training, only excluding scenes 0015 and 0022. The MegaDepth scenes employed in their training have some overlap with the IMC benchmark, although the images are not identical; the same scenes are present in both datasets. For instance, the MegaDepth scene 0024 corresponds to the British Museum, while the British Museum is also a scene in the IMC benchmark. For an apples-to-apples comparison, we adopt the same training split as DUSt3R and MASt3R. In the main paper, to ensure a fair comparison on ScanNet-1500, we exclude the corresponding ScanNet scenes from our training.

Results. Table 10 contains the results of our evaluation. Although phototourism data is the traditional focus of SfM methods, our VGGT's feed-forward performance is on par with the state-of-the-art VGGSfMv2 with AUC@10 of 71.26 versus 76.82, while being significantly faster (0.2 vs. 10 seconds per scene). Remarkably, VGGT outperforms both MASt3R [61] and DUSt3R [128] significantly across all accuracy thresholds while being much faster. This is because MASt3R's and DUSt3R's feed-forward predictions can only process pairs of frames and, hence, require a costly global alignment step. Additionally, with bundle adjustment, VGGT + BA further improves drastically, achieving state-of-the-art performance on IMC, raising AUC@10 from 71.26 to 84.91, and raising AUC@3 from 39.23 to 66.37. Note that our model directly predicts 3D points, which can serve as the initialization for BA. This eliminates the need for triangulation and iterative refinement of BA as in [124]. As a result, VGGT + BA is much faster than [124].

Method	Test-time Opt.	AUC@3°	AUC@5°	AUC@10°	Runtime
COLMAP (SIFT+NN) [93]	✓	23.58	32.66	44.79	>10s
PixSfM (SIFT + NN) [65]	✓	25.54	34.80	46.73	>20s
PixSfM (LoFTR) [65]	✓	44.06	56.16	69.61	>20s
PixSfM (SP + SG) [65]	✓	45.19	57.22	70.47	>20s
DFSfM (LoFTR) [46]	✓	46.55	58.74	72.19	>10s
DUSt3R [128]	✓	13.46	21.24	35.62	∼ 7s
MASt3R [61]	✓	30.25	46.79	57.42	∼ 9s
VGGSfM [124]	✓	45.23	58.89	73.92	∼ 6s
VGGSfMv2 [124]	✓	<u>59.32</u>	<u>67.78</u>	<u>76.82</u>	∼ 10s
VGGT (ours)	✗	39.23	52.74	71.26	**0.2s**
VGGT + BA (ours)	✓	**66.37**	**75.16**	**84.91**	1.8s

Table 10. **Camera Pose Estimation on IMC [53].** Our method achieves state-of-the-art performance on the challenging phototropism data, outperforming VGGSfMv2 [124] which ranked first on the latest CVPR'24 IMC Challenge in camera pose (rotation and translation) estimation.

D. Qualitative Examples

We further present qualitative examples of single-view reconstruction in Fig. 7.

E. Related Work

In this section, we discuss additional related works.

Vision Transformers. The Transformer architecture was initially proposed for language processing tasks [6, 21, 119]. It was later introduced to the computer vision community by ViT [26], sparking widespread adoption. Vision Transformers and their variants have since become dominant in the design of architectures for various computer vision tasks [4, 11, 82, 136], thanks to their simplicity, high capacity, flexibility, and ability to capture long-range dependencies.

DeiT [113] demonstrated that Vision Transformers can be effectively trained on datasets like ImageNet using strong data augmentation strategies. DINO [9] revealed

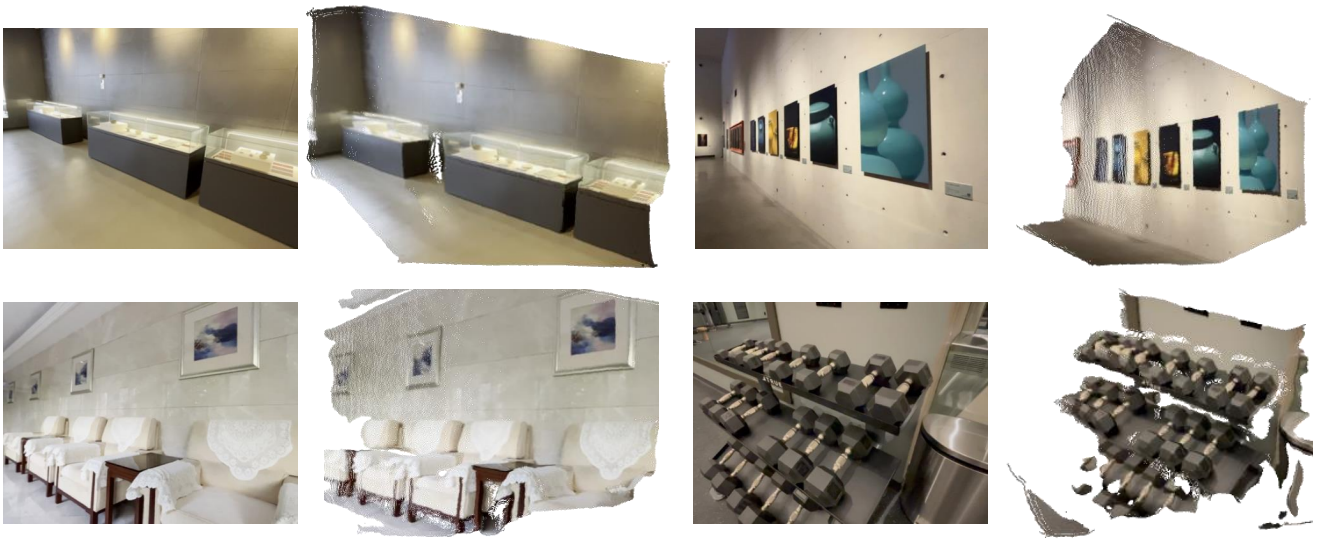

Figure 7. **Single-view Reconstruction by Point Map Estimation.** Unlike DUSt3R, which requires duplicating an image into a pair, our model can predict the point map from a single input image.

intriguing properties of features learned by Vision Transformers in a self-supervised manner. CaiT [114] introduced layer scaling to address the challenges of training deeper Vision Transformers, effectively mitigating gradient-related issues. Further, techniques such as QKNorm [47, 149] have been proposed to stabilize the training process. Additionally, [137] also explores the dynamics between frame-wise and global attention modules in object tracking, though using cross-attention.

Camera Pose Estimation. Estimating camera poses from multi-view images is a crucial problem in 3D computer vision. Over the last decades, Structure from Motion (SfM) has emerged as the dominant approach [45], whether incremental [2, 35, 93, 102, 133] or global [3, 13–16, 51, 72, 78, 80, 89, 105]. Recently, a set of methods treat camera pose estimation as a regression problem [64, 99, 108, 111, 112, 117, 121, 122, 130, 151, 152, 159], which show promising results under the sparse-view setting. AceZero [5] further proposes to regress 3D scene coordinates and FlowMap [100] focuses on depth maps, as intermediates for camera prediction. Instead, VGGSfM [124] simplifies the classical SfM pipeline to a differentiable framework, demonstrating exceptional performance, particularly with phototourism datasets. At the same time, DUSt3R [61, 128] introduces an approach to learn pixel-aligned point map, and hence camera poses can be recovered by simple alignment. This paradigm shift has garnered considerable interest as the point map, an over-parameterized representation, offers seamless integration with various downstream applications, such as 3D Gaussian splatting.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 2, 13

[3] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *2012 Second international conference on 3D imaging, modeling, processing, visualization & transmission*, pages 81–88. IEEE, 2012. 13

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 12

[5] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024. 2, 13

[6] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 12

[7] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 6

[8] Chenjie Cao and Yanwei Fu. Improving transformer-based image matching by cascaded capturing spatially informative keypoints. In *Proceedings of the IEEE/CVF Inter-*

national Conference on Computer Vision (ICCV), pages 12129–12139, 2023. 7

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021. 2, 12

[10] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6301–6310, 2021. 2

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 12

[12] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. *Proc. ECCV*, 2024. 3, 10

[13] David J Crandall, Andrew Owens, Noah Snavely, and Daniel P Huttenlocher. Sfm with mrfs: Discrete-continuous optimization for large-scale structure from motion. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2841–2853, 2012. 13

[14] Hainan Cui, Xiang Gao, Shuhan Shen, and Zhanyi Hu. Hsfm: Hybrid structure-from-motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1212–1221, 2017.

[15] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 864–872, 2015.

[16] Zhaopeng Cui, Nianjuan Jiang, Chengzhou Tang, and Ping Tan. Linear global translation estimation with feature tracks. *arXiv preprint arXiv:1503.01832*, 2015. 13

[17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 6, 7, 8

[18] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 5

[19] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 6, 9

[20] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. 12

[22] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *arXiv*, 2022. 2, 10

[23] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. *arXiv*, 2306.08637, 2023. 2

[24] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: tracking any point with per-frame initialization and temporal refinement. In *Proc. CVPR*, 2023. 3, 9

[25] Carl Doersch, Yi Yang, Dilara Gokay, Pauline Luc, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ross Goroshin, João Carreira, and Andrew Zisserman. Bootstap: Bootstrapped training for tracking-any-point. *arXiv preprint arXiv:2402.00847*, 2024. 10

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16×16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2021. 12

[27] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 9

[28] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani,

Kate Plawiak, Ke Li, Kenneth Heafield, and Kevin Stone. The Llama 3 herd of models. *arXiv*, 2407.21783, 2024. 2

[29] Bardienus Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3R-SfM: a fully-integrated solution for unconstrained structure-from-motion. *arXiv*, 2409.19152, 2024. 6

[30] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 8092–8101, 2019. 2

[31] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 7

[32] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024. 7, 8, 12

[33] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proc. CVPR*, 2021. 2

[34] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[35] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 368–381. Springer, 2010. 2, 13

[36] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022. 2

[37] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 2

[38] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2015. 2

[39] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015. 7

[40] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. In *Proc. CVPR*, 2022. 6, 10

[41] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020. 2

[42] Junlin Han, Jianyuan Wang, Andrea Vedaldi, Philip Torr, and Filippos Kokkinos. Flex3d: Feed-forward 3d generation with flexible reconstruction model and input view curation. *arXiv preprint arXiv:2410.00890*, 2024. 9

[43] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *Proc. ECCV*, 2022. 2, 9

[44] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1, 2

[45] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004. 13

[46] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *arxiv*, 2023. 12

[47] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020. 11, 13

[48] 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. In *Proc. ICLR*, 2024. 2, 9

[49] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[50] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 7

[51] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *Proceedings of the IEEE international conference on computer vision*, pages 481–488, 2013. 13

[52] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: a large view synthesis model with minimal 3D inductive bias. *arXiv*, 2410.17242, 2024. 5, 9

[53] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2): 517–547, 2021. 12

[54] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 2

[55] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *Proc. ECCV*, 2024. 2, 8, 10

[56] Nikita Karaev, Ignacio Rocco, Ben Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3, 5, 6, 9, 10

[57] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proc. ICRA*. IEEE, 2016. 5

[58] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Proc. NeurIPS*, 2017. 6

[59] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: Connecting the dots. In *CVPR*, 2024. 2

[60] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009. 3

[61] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 2, 7, 12, 13

[62] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. Taptr: Tracking any point with transformers as detection. *arXiv preprint arXiv:2403.13042*, 2024. 2, 10

[63] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 6

[64] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. *arXiv preprint arXiv:2305.04926*, 2023. 13

[65] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. *arXiv.cs*, abs/2108.08291, 2021. 7, 12

[66] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. *arXiv preprint arXiv:2306.13643*, 2023. 2

[67] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: local feature matching at light speed. In *Proc. ICCV*, 2023. 8

[68] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 6

[69] Shaohui Liu, Yidan Gao, Tianyi Zhang, Rémi Pautrat, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Robust incremental structure-from-motion with hybrid features. In *European Conference on Computer Vision*, pages 249–269. Springer, 2025. 2

[70] Manuel Lopez-Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota BulÃ², Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6

[71] Zeyu Ma, Zachary Teed, and Jia Deng. Multiview stereo with cascaded epipolar raft. In *European Conference on Computer Vision*, pages 734–750. Springer, 2022. 2

[72] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proceedings of the IEEE international conference on computer vision*, pages 3248–3255, 2013. 13

[73] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3504–3515, 2020. 2

[74] David Novotný, Diane Larlus, and Andrea Vedaldi. Learning 3D object categories by looking around them. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017. 6

[75] David Novotný, Diane Larlus, and Andrea Vedaldi. Capturing the geometry of object categories from video supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 5

[76] John Oliensis. A critique of structure-from-motion algorithms. *Computer Vision and Image Understanding*, 80(2): 172–214, 2000. 2

[77] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 2, 5, 11

[78] Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015. 13

[79] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 2

[80] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024. 13

[81] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng (Carl) Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20133–20143, 2023. 6

[82] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 12

[83] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8645–8654, 2022. 2

[84] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, et al. Theseus: A library for differentiable nonlinear optimization. *Advances in Neural Information Processing Systems*, 35:3801–3818, 2022. 10

[85] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, pages 8748–8763, 2021. 2

[86] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 3, 5, 11

[87] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proc. ICCV*, 2021. 6, 7

[88] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision (ICCV) 2021*, 2021. 6

[89] Rother. Linear multiview reconstruction of points, lines, planes and cameras using a reference plane. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1210–1217. IEEE, 2003. 13

[90] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80, 2008. 2

[91] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 7

[92] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: learning feature matching with graph neural networks. In *Proc. CVPR*, 2020. 8

[93] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 12, 13

[94] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 3, 6

[95] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 2

[96] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017. 7, 8

[97] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37: 68658–68685, 2024. 10

[98] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12517–12526, 2022. 2

[99] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. Sparsepose: Sparse-view camera pose regression and refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21349–21359, 2023. 13

[100] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. *arXiv preprint arXiv:2404.15259*, 2024. 13

[101] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. FlowMap: high-quality camera poses, intrinsics, and depth via gradient descent. *arXiv*, 2404.15259, 2024. 2

[102] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 2, 13

[103] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6

[104] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 7, 8, 12

[105] Chris Sweeney, Torsten Sattler, Tobias Hollerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *Proceedings of the IEEE international conference on computer vision*, pages 801–809, 2015. 13

[106] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets,

Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 6

[107] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10208–10217, 2024. 9

[108] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2, 13

[109] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 9

[110] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *arXiv preprint arXiv:2412.06974*, 2024. 2, 7

[111] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 2, 13

[112] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 2, 13

[113] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 12

[114] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 32–42, 2021. 11, 13

[115] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 2

[116] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4), 1991. 8

[117] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 2, 13

[118] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017. 2, 5

[119] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 12

[120] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, pages 14194–14203, 2021. 7

[121] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2021. 2, 13

[122] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023. 13

[123] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: solving pose estimation via diffusion-aided bundle adjustment. In *Proc. ICCV*, 2023. 6, 7

[124] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: visual geometry grounded deep structure from motion. In *Proc. CVPR*, 2024. 1, 2, 3, 5, 7, 10, 12, 13

[125] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. PF-LRM: pose-free large reconstruction model for joint pose and shape prediction. *arXiv.cs*, abs/2311.12024, 2023. 9

[126] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state, 2025. 2, 7

[127] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv*, 2410.19115, 2024. 2

[128] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proc. CVPR*, 2024. 1, 2, 3, 5, 6, 7, 11, 12, 13

[129] Yuesong Wang, Zhaojie Zeng, Tao Guan, Wei Yang, Zhuo Chen, Wenkai Liu, Luoyuan Xu, and Yawei Luo. Adaptive patch deformation for textureless-resilient multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1621–1630, 2023. 2

[130] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 230–247. Springer, 2020. 2, 13

[131] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. MeshLRM: large reconstruction model for high-quality mesh. *arXiv*, 2404.12385, 2024. 5

[132] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5610–5619, 2021. 2

[133] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 2, 13

[134] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos, 2024. 6

[135] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024. 9

[136] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021. 12

[137] Fei Xie, Chunyu Wang, Guangting Wang, Yue Cao, Wankou Yang, and Wenjun Zeng. Correlation-aware deep tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8751–8760, 2022. 13

[138] Qingshan Xu and Wenbing Tao. Learning inverse depth regression for multi-view stereo with correlation cost volume. In *AAAI*, 2020. 7

[139] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. GRM: Large gaussian reconstruction model for efficient 3D reconstruction and generation. *arXiv*, 2403.14621, 2024. 9

[140] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. *arXiv preprint arXiv:2501.13928*, 2025. 2, 7, 10

[141] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proc. CVPR*, 2024. 2

[142] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 11

[143] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 7

[144] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 2

[145] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020. 6

[146] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 2

[147] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G., Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H. Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *arXiv.cs*, abs/2305.10435, 2023. 2

[148] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned Invariant Feature Transform. In *Proc. ECCV*, 2016. 2

[149] Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pages 40770–40803. PMLR, 2023. 13

[150] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: a simple approach for estimating geometry in the presence of motion. *arXiv*, 2410.03825, 2024. 11

[151] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, pages 592–611. Springer, 2022. 13

[152] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 13

[153] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2024. 9

[154] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: large reconstruction model for 3D Gaussian splatting. *arXiv*, 2404.19702, 2024. 9

[155] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views, 2025. 2, 7

[156] Zhe Zhang, Rui Peng, Yuxi Hu, and Ronggang Wang. Geomvsnet: Learning multi-view stereo with geometry perception. In *CVPR*, 2023. 2, 7

[157] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable

transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023. 8

[158] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 6

[159] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 2, 13

[160] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 6, 7