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) [33]. 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 [83] 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 [87] and its evolution

MASt3R [43] 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 [79], 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 mould as large models for natural language processing and computer vision, such as GPTs [1, 18, 101], CLIP [56], DINO [6, 53], and Stable Diffusion [22]. 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 [97], MoGe [86], and LRM [34]. 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.

2. Related Work

Structure from Motion is a classic computer vision problem [33, 52, 54] 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, 24, 48, 62, 68, 92] consists of multiple stages, including image matching, triangulation, and bundle adjustment. COLMAP [62] 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 [12, 19, 76, 102] and image matching [7, 46, 60, 66] being two primary areas of focus. Recent methods [3, 67, 71, 74, 75, 78, 81, 83, 89, 108] explored end-to-end differentiable SfM, where VGGSfM [83] 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 [26, 27, 64, 88], global optimization [25, 50, 91, 100], and learning-based methods [30, 49, 55, 99, 106]. As in SfM, learning-based MVS approaches have recently seen a lot of progress. Here, DUSt3R [87] and MASt3R [43] directly estimate aligned dense point clouds from a pair of views, similar to MVS but without requiring camera parameters. Some concurrent works [73, 85, 96, 105] 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 [59] and revived by PIPs [32] during the deep learning era, aiming to track points of interest across video sequences, including dynamic motion. Given a video and some 2D query points, the task is to predict the 2D correspondences of these points across all other frames. TAP-Vid [13] proposed three benchmarks for this task, and a simple baseline method later improved in TAPIR [14]. CoTracker [37, 38] utilized correlations between different points to track through occlusions, while DOT [41] enabled dense tracking through occlusions. Recently, TAPTR [44] proposed an end-to-end transformer for this task, and LocoTrack [8] extended commonly used pointwise features to nearby regions. Here, we demonstrate that VGGT's features yield state-of-the-art tracking performance when coupled with existing point trackers.

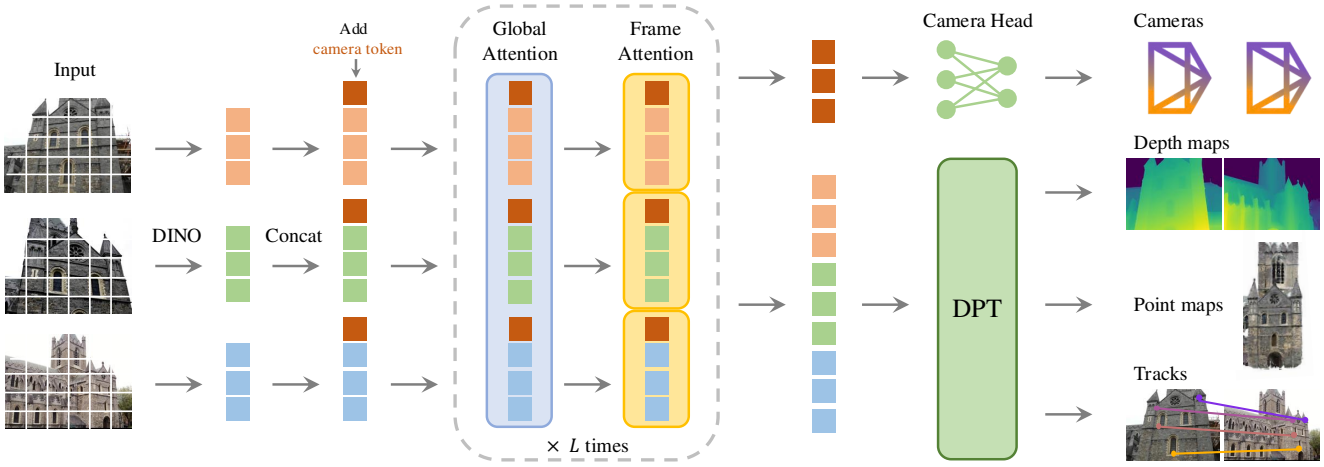

Figure 2. **Architecture Overview.** Our model first patchifies the input images into tokens by DINO, and appends camera tokens for camera prediction. It then alternates between frame-wise and global self attention layers. A camera head makes the final prediction for camera extrinsics and intrinsics, and a DPT [57] head for any dense output.

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.

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:

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

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 [83] 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 [63, 83].

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 [87], 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 [15, 39]. 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 [87], the camera parameters $\mathbf{g}$ can be inferred from the invariant point map P, for instance, by solving the Perspective-n-Point (PnP) problem [23, 42]. 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

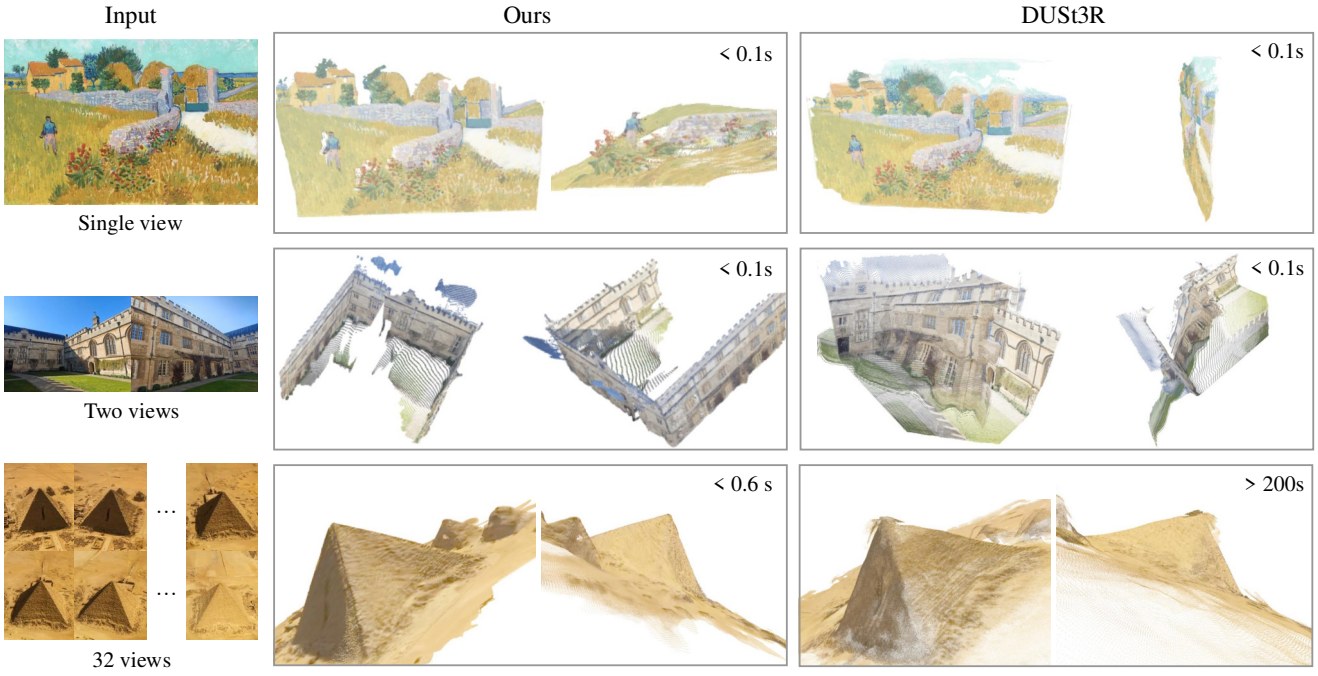

Figure 3. **In-the-wild comparison of our predicted 3D points to DUSt3R.** As shown in the top row, our method successfully predicts the geometric structure of an oil painting, while DUSt3R predicts a slightly distorted plane. In the second row, our method correctly recovers a 3D scene from two images 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 then runs out of memory.

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 [36, 87, 90], 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 [79]. 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 [53]. 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.

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

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 [10] $\mathrm{t}_i^R \in \mathbb{R}^{4 \times C'}$. 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 Predictions. 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 [57]. 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 [40, 51] $\Sigma_i^D \in \mathbb{R}_+^{H \times W}$ and $\Sigma_i^P \in \mathbb{R}_+^{H \times W}$ for each depth and point map, respectively. The uncertainty maps are used in the loss (as detailed in the supplementary) 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 [39], 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 [83], 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.

4. Experiments

This section compares our method to state-of-the-art approaches across multiple tasks to show its effectiveness. Detailed discussions on architecture, training losses, and datasets are included in the supplementary material.

Methods	Re10K *(unseen)* AUC@30 ↑	CO3Dv2 AUC@30 ↑	Time
Colmap+SPSG [60]	45.2	25.3	∼ 15s
PixSfM [45]	49.4	30.1	> 20s
PoseDiff [82]	48.0	66.5	∼ 7s
DUSt3R [87]	67.7	76.7	∼ 7s
MASt3R [43]	76.4	81.8	∼ 9s
VGGSfM v2 [83]	78.9	83.4	∼ 10s
MV-DUSt3R [73] ‡	71.3	(69.5)	∼ 0.6s
CUT3R [85] ‡	75.3	82.8	∼ 0.6s
FLARE [105] ‡	78.8	83.4	∼ 0.5s
Fast3R [96] ‡	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 [109] and CO3Dv2 [58]** with 10 random frames. All metrics the higher the better. Runtime were measured using one H100 GPU. None of the methods were trained on the Re10K dataset; () means not trained on CO3D. Methods marked with ‡ represent concurrent work.

4.1. Camera Pose Estimation

We first evaluate our method on the CO3Dv2 [58] and RealEstate10K [109] datasets for camera pose estimation, as shown in Tab. 1. Following [82], we randomly select 10 images per scene and evaluate them using the standard metric 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 (learnable) methods in Tab. 1 have been trained on Co3Dv2 and **not** on RealEstate10K. Our feed-forward model consistently 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*, requiring just 0.2 seconds on the same hardware. Compared to concurrent works [73, 85, 96, 105] (indicated by ‡), our method demonstrates significant performance advantages, with speed similar to the fastest variant Fast3R [96]. 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.

Our results also show that VGGT can be improved even further by combining it with optimization methods from visual geometry optimization like BA. Specifically, refining

Known GT camera	Method	Acc.↓	Comp.↓	Overall↓
✓	Gipuma [28]	**0.283**	0.873	0.578
✓	MVSNet [98]	0.396	0.527	0.462
✓	CIDER [94]	0.417	0.437	0.427
✓	PatchmatchNet [80]	0.427	0.377	0.417
✓	MASt3R [43]	0.403	0.344	0.374
✓	GeoMVSNet [106]	0.331	**0.259**	**0.295**
✗	DUSt3R [87]	2.677	0.805	1.741
✗	Ours	**0.389**	**0.374**	**0.382**

Table 2. **Dense MVS Estimation on the DTU [35] 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)	<u>0.901</u>	<u>0.518</u>	<u>0.709</u>	∼ 0.2s
Ours (Depth + Cam)	**0.873**	**0.482**	**0.677**	∼ 0.2s

Table 3. **Point Map Estimation on ETH3D [65].** 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.

the predicted camera poses and tracks with BA 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 in BA as done by [83], making our approach significantly faster (only around 2 seconds even with BA). Hence, while the feed-forward mode of 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 [43], we further evaluate our multi-view depth estimation results on the DTU [35] 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 cameras. Meanwhile, deep multi-view stereo 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 ground-truth cameras at test time. The significant performance gains can likely be attributed to our model's multi-image training scheme that teaches it to reason about multi-view triangulation natively instead of relying on ad hoc 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 [65] dataset. For each scene, we randomly sample 10 frames. The predicted point cloud is aligned to the ground truth using the Umeyama [77] 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 expensive 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 the predictions from our depth and camera heads (*i.e.*, unprojecting the predicted depth maps to 3D using the predicted camera parameters) yield higher accuracy. We attribute this to the benefits of decomposing a complex task (point map estimation) into simpler subproblems (depth map and camera prediction), even though the 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, with additional examples included in the supplementary material. VGGT outputs high-quality predictions and generalizes well, excelling on challenging out-of-domain examples, such as oil paintings, non-overlapping frames, and scenes with repeating or homogeneous textures like deserts.

4.4. Image Matching

Two-view image matching is a widely-explored topic [47, 61, 69] 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 [21, 61] on the ScanNet dataset [9] 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 [107] 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

Method	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
SuperGlue [60]	16.2	33.8	51.8
LoFTR [69]	22.1	40.8	57.6
DKM [20]	29.4	50.7	68.3
CasMTR [5]	27.1	47.0	64.4
Roma [21]	31.8	53.4	70.9
Ours	**33.9**	**55.2**	**73.4**

Table 4. **Two-View matching comparison on ScanNet-1500 [9, 60]**. 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).

evaluation hyperparameters (*e.g.*, the number of matches, RANSAC thresholds) from Roma [21]. 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 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.

ETH3D Dataset	Acc.↓	Comp.↓	Overall↓
Cross-Attention	1.287	0.835	1.061
Global Self-Attention Only	1.032	0.621	0.827
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.

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.

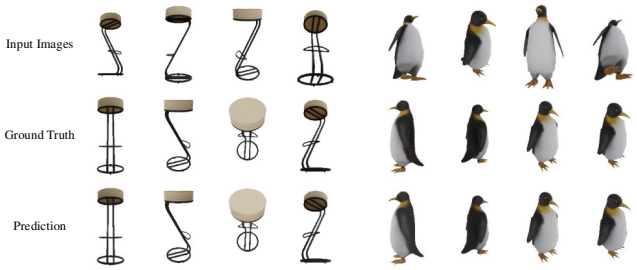

Figure 4. **Qualitative Examples of Novel View Synthesis.** The top row shows the input images, the middle row displays the ground truth images from target viewpoints, and the bottom row presents our synthesized images.

Method	Known Input Cam	Size	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [72]	✓	256	21.44	0.832	0.122
GS-LRM [103]	✓	256	29.59	0.944	0.051
LVSM [36]	✓	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 [17] 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%).

Feed-forward Novel View Synthesis is progressing rapidly [4, 31, 34, 36, 70, 84, 95, 104]. 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 [36] and modify VGGT to *directly* output the target image. 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. We make a simple modifi-

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 [38], which processes sequential inputs.

Method	Kinetics			RGB-S			DAVIS		
	AJ	$\delta^{\text{vis}}_{\text{avg}}$	OA	AJ	$\delta^{\text{vis}}_{\text{avg}}$	OA	AJ	$\delta^{\text{vis}}_{\text{avg}}$	OA
TAPTR [44]	49.0	64.4	85.2	60.8	76.2	87.0	63.0	76.1	91.1
LocoTrack [8]	52.9	66.8	85.3	69.7	83.2	89.5	62.9	75.3	87.2
BootsTAPIR [16]	54.6	68.4	86.5	70.8	83.0	89.9	61.4	73.6	88.7
CoTracker [38]	49.6	64.3	83.3	67.4	78.9	85.2	61.8	76.1	88.3
CoTracker + Ours	**57.2**	**69.0**	**88.9**	**72.1**	**84.0**	**91.6**	**64.7**	**77.5**	**91.4**

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.

cation to VGGT. As before, the input images are converted into tokens by DINO. Then, for the target views, we use a convolutional layer to encode their Plücker ray images into tokens. These tokens, representing both the input images and the target views, are concatenated and processed by the AA transformer. Subsequently, a DPT head is used to regress the RGB colors for the target views. It is important to note that we do *not* input the Plücker rays for the source images. Hence, the model is not given the camera parameters for these input frames.

LVSM was trained on the Objaverse dataset [11]. We use a similar internal dataset of approximately 20% the size of Objaverse. Further details on training and evaluation can be found in [36]. As shown in Tab. 7, despite not requiring the input camera parameters and using less training data than LVSM, our model achieves competitive results on the GSO dataset [17]. We expect that better results would be obtained using a larger training dataset. Qualitative examples are shown in Fig. 4.

Dynamic Point Tracking has emerged as a highly competitive task in recent years [15, 32, 39, 93], 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 predictions; $\delta^{\text{vis}}_{\text{avg}}$, comprising the mean proportion of visible points accurately tracked within a certain pixel threshold; and Average Jaccard (AJ), measuring tracking and occlusion prediction accuracy together.

We adapt the state-of-the-art CoTracker2 model [39] by substituting its backbone with our pretrained feature backbone. This is necessary because VGGT is trained on unordered image collections instead of sequential dynamic 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, which finally predicts the tracks. We finetune the entire modified tracker on Kubric [29]. As illustrated in Tab. 8, the integration of pretrained VGGT significantly enhances CoTracker's performance on the TAP-Vid benchmark [13]. For instance, VGGT's tracking features improve the $\delta^{\text{vis}}_{\text{avg}}$ metric from 78.9 to 84.0 on the TAP-Vid RGB-S dataset. Despite the inclusion of videos featuring rapid dynamic motions in TAP-Vid, our model's strong performance demonstrates the generalization capability of its features.

5. 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 post-optimization 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.

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

[3] 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

[4] Ang Cao, Justin Johnson, Andrea Vedaldi, and David Novotny. Lightplane: Highly-scalable components for neural 3Dfields. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2025. 7

[5] Chenjie Cao and Yanwei Fu. Improving transformer-based image matching by cascaded capturing spatially informative keypoints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12129–12139, 2023. 7

[6] 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

[7] 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

[8] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. *Proc. ECCV*, 2024. 2, 8

[9] 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

[10] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 4

[11] 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. 8

[12] 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

[13] 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, 8

[14] 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

[15] 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, 8

[16] 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. 8

[17] 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. 7, 8

[18] 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

[19] 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

[20] 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

[21] 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. 6, 7

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

[23] 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

[24] 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

[25] 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

[26] 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

[27] 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

[28] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015. 6

[29] 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. 8

[30] 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

[31] 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. 7

[32] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *Proc. ECCV*, 2022. 2, 8

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

[34] 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, 7

[35] 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. 6

[36] 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. 4, 7, 8

[37] 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

[38] 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

[39] 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, 8

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

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

[42] 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

[43] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 2, 5, 6

[44] 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, 8

[45] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. *arXiv.cs*, abs/2108.08291, 2021. 5

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

[47] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: local feature matching at light speed. In *Proc. ICCV*, 2023. 6

[48] 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

[49] 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

[50] 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

[51] 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

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

[53] 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, 4

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

[55] 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

[56] 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

[57] 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

[58] 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. 5

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

[60] 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, 5, 7

[61] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: learning feature matching with graph neural networks. In *Proc. CVPR*, 2020. 6

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

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

[64] 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

[65] 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. 6

[66] 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

[67] 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

[68] 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

[69] 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. 6, 7

[70] 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. 7

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

[72] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content cre-

ation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 7

[73] 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, 5

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

[75] 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

[76] 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

[77] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4), 1991. 6

[78] 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

[79] 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, 4

[80] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, pages 14194–14203, 2021. 6

[81] 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

[82] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: solving pose estimation via diffusion-aided bundle adjustment. In *Proc. ICCV*, 2023. 5

[83] 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, 6

[84] 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. 7

[85] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state, 2025. 2, 5

[86] 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

[87] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proc. CVPR*, 2024. 1, 2, 3, 4, 5, 6

[88] 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

[89] 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

[90] 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. 4

[91] 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

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

[93] 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. 8

[94] Qingshan Xu and Wenbing Tao. Learning inverse depth regression for multi-view stereo with correlation cost volume. In *AAAI*, 2020. 6

[95] 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. 7

[96] 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, 5

[97] 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

[98] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 6

[99] 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

[100] 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

[101] 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

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

[103] 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. 7

[104] 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. 7

[105] 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, 5

[106] Zhe Zhang, Rui Peng, Yuxi Hu, and Ronggang Wang. Geomvsnet: Learning multi-view stereo with geometry perception. In *CVPR*, 2023. 2, 6

[107] 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. 6

[108] 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

[109] 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. 5