ViPE: Video Pose Engine for 3D Geometric Perception

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, Sanja Fidler

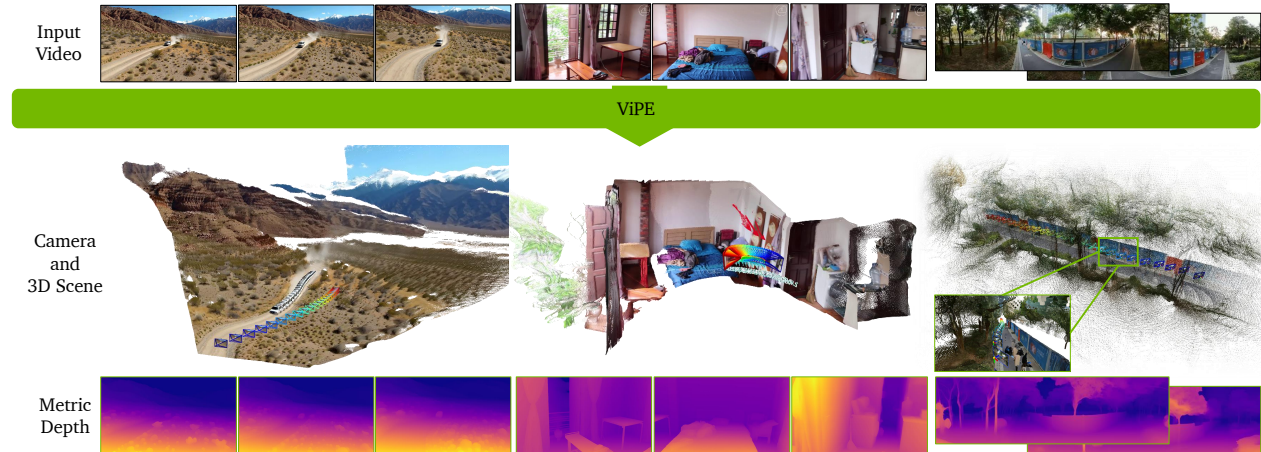NVIDIA[1]

https://research.nvidia.com/labs/toronto-ai/vipe/

Figure 1: We present **ViPE**, a powerful and versatile video annotation engine. From a casual video, ViPE outputs estimated camera motion and dense, metric-scale depth maps. ViPE robustly handles diverse camera models, including standard perspective, wide-angle, or 360° panoramic videos.

Abstract

Accurate 3D geometric perception is an important prerequisite for a wide range of spatial AI systems. While state-of-the-art methods depend on large-scale training data, acquiring consistent and precise 3D annotations from in-the-wild videos remains a key challenge. In this work, we introduce ViPE, a handy and versatile video processing engine designed to bridge this gap. ViPE efficiently estimates camera intrinsics, camera motion, and dense, near-metric depth maps from unconstrained raw videos. It is robust to diverse scenarios, including dynamic selfie videos, cinematic shots, or dashcams, and supports various camera models such as pinhole, wide-angle, and 360° panoramas. We have benchmarked ViPE on multiple benchmarks. Notably, it outperforms existing uncalibrated pose estimation baselines by 18%/50% on TUM/KITTI sequences, and runs at 3-5FPS on a single GPU for standard input resolutions. We use ViPE to annotate a large-scale collection of videos. This collection includes around 100K real-world internet videos, 1M high-quality AI-generated videos, and 2K panoramic videos, totaling approximately 96M frames – all annotated with accurate camera poses and dense depth maps. We open-source ViPE and the annotated dataset with the hope of accelerating the development of spatial AI systems.

1. Introduction

The ability to understand 3D environments is a cornerstone of spatial intelligence for applications ranging from robotics to VR/AR, and autonomous systems. The foundational task of estimating low-level geometry—camera parameters and 3D scene structure remains a critical first step to many downstream technologies such as 3D reconstruction, camera or depth-conditioned video generation models, and training robotic policies.

Traditionally, this problem has been tackled by two main classes of methods. Classical Simultaneous Localization and Mapping (SLAM) systems [17, 49] excel at estimating camera poses and sparse geometry from long video sequences, leveraging temporal consistency and loop closure [8]. However, they typically assume a static scene and known camera intrinsics, and can be less robust to dynamic objects or degenerate motions. While some systems like COLMAP [59] can refine intrinsics, jointly optimizing them with dense geometry for diverse, non-curated videos remains a challenge.

More recently, end-to-end feed-forward models [15] have emerged, trained on large datasets to directly regress camera poses and depth from images. While these methods show impressive robustness, their scalability is a significant bottleneck. Processing long videos is often intractable due to large GPU memory footprints, forcing practitioners to resort to subsampling video frames or processing short, disconnected chunks [47, 54]. A promising recent trend seeks a hybrid approach between SLAM and feed-forward approaches by integrating powerful learned front-ends like MaSt3R [36] into traditional SLAM back-ends, as demonstrated in systems like MASt3R-SLAM [50] and concurrent work such as VGGT-SLAM [47]. However, simply swapping the front-end is often insufficient in practice. As we demonstrate, these methods can still lack the accuracy and robustness required for large-scale annotation of diverse, in-the-wild videos, which motivates the need for a more tightly integrated system.

In this work, we introduce **Vi**deo **P**ose **E**ngine (shortened as **ViPE**), designed to bridge the gap between classical and learning-based approaches. It combines the scalability and precision of a dense Bundle Adjustment (BA) framework, akin to SLAM, with the robustness of modern learned components. This synergy allows ViPE to accurately and efficiently estimate camera poses, intrinsics, and dense, metric depth maps from challenging, in-the-wild videos. Compared to the closest prior work, MegaSAM [37], ViPE does not require per-frame optimization and is hence more efficient. More robust strategies are presented for handling dynamic objects, and a wider variety of camera models are supported. ViPE is also experimentally shown to provide more accurate camera estimation results. Speed-wise, ViPE can typically reach a speed of **3-5FPS** on a single GPU[2]. These advancements make our system uniquely suited for the demands of large-scale, diverse video annotation.

We put ViPE in action to annotate a large-scale annotated dataset, which we publicly release along ViPE. The dataset release comprises three distinct components: **Dynpose-100K++**, a re-annotation of approximately 100K challenging real-world internet videos with high-quality poses and dense geometry; **Wild-SDG-1M**, a large dataset of 1M high-quality, AI-generated videos sampled from video diffusion models; and **Web360**, a specialized dataset of annotated panoramic videos. Collectively, this release provides 96 million annotated frames across numerous varied sources, aiming to facilitate many downstream applications.

We summarize the key contributions:

- A robust and efficient framework, ViPE, for estimating camera parameters and dense depth from diverse, in-the-wild videos.
- A system design that integrates the strengths of classical SLAM (efficiency, scalability) and learned models (robustness), with key improvements over prior work in efficiency, dynamic object handling, and depth quality.
- A large-scale dataset of annotated videos, created using ViPE, to facilitate future research in 3D computer vision.

2. Related Works

Our work builds upon decades of research in 3D reconstruction, as well as the recent paradigm shift towards using deep learning-based methods for low-level geometric tasks. We position our contributions relative to two main lines of work: classical optimization-based systems and modern feed-forward perception models.

---

[2]Measured with the input resolution of 640×480 on NVIDIA RTX 5090 GPU.

2.1. Visual SLAM and SfM

The traditional approaches to 3D reconstruction from images can be divided into Structure-from-Motion (SfM) and Simultaneous Localization and Mapping (SLAM) techniques. SfM systems, such as the widely used COLMAP [51, 59], are typically designed to process unordered collections of images, performing global optimization via Bundle Adjustment (BA) to recover highly accurate camera poses and sparse point clouds. SLAM systems, like ORB-SLAM [49] and others [8, 17, 20, 35], are tailored for sequential video streams, processing frames incrementally to track camera motion in real-time while building a map of the environment.

Despite their success and precision, these classical methods face significant challenges with the "in-the-wild" videos, which we focus on in our work. Their reliance on hand-crafted feature matching struggles in poorly-textured regions, and they are notoriously brittle in the presence of dynamic objects or non-rigid motion, which violates their fundamental static-world assumption. While systems like GLOMAP [51] have improved scalability, the core challenges remain. More recently, a trend towards purely data-driven SfM pipelines has emerged, where methods such as ParticleSfM [90], VGGSfM [71], DATAP-SfM [84], and DiffusionSfM [89] leverage deep learning priors to replace classical components. While powerful, these approaches often focus on smaller-scale problems or specific aspects of the pipeline. ViPE, on the other hand, targets a robust, scalable, and fully-integrated system.

2.2. Feed-forward 3D Perception Models

Classical methods, which depend on geometric consistency, often fail in challenging scenarios such as textureless regions, repetitive patterns, or wide-baseline views where feature matching is ambiguous. To address this, a recent wave of research has focused on feed-forward models that leverage powerful priors learned from large-scale datasets. This paradigm began with pairwise models like DUSt3R [74] and MASt3R [36] and was quickly extended to multi-view settings that improved accuracy for general scenes [7, 44, 64, 69, 70, 73, 75, 79, 82]. However, a critical bottleneck for this entire family of models is scalability; their computational and memory requirements grow quickly with the number of input frames, making it intractable to process long videos.

This limitation directly spurred the development of hybrid systems that integrate a robust feed-forward front-end into a classical SLAM [43, 47, 50] or SfM [18, 19, 41] back-end to handle long sequences. While these hybrids represent a significant step, they often involve a "loose coupling" that does not fully resolve inconsistencies between the learned front-end and the optimization back-end.

To further achieve high-quality, metric-scale dense geometry, our system also builds upon another important line of research: powerful monocular metric depth estimators [27, 52, 53] and video depth models [9, 14]. We integrate these as critical components at multiple stages: as a regularizing prior during optimization to resolve scale ambiguity, inspired by works like [76], and as a source for high-quality depth refinement.

In parallel, handling dynamic content remains a significant challenge. This has been tackled both by pure feed-forward models for pairwise image [11, 21, 33, 63, 74, 86, 87] or short video inputs [3, 31, 77, 80], and by dense reconstruction systems like CasualSAM [88] and MegaSAM [37]. Our work is most closely related to this latter category. By addressing scalability, handling dynamic objects, and reaching metric scale in a unified framework, ViPE introduces several key advantages over prior work, including a more efficient keyframe-based architecture, a more sophisticated strategy for modeling dynamics, and broader support for diverse camera models.

2.3. Downstream Applications

Large-scale datasets of videos annotated with accurate camera poses and 3D geometry are essential for a wide range of downstream applications. Such annotations serve both as valuable supervision signals during training
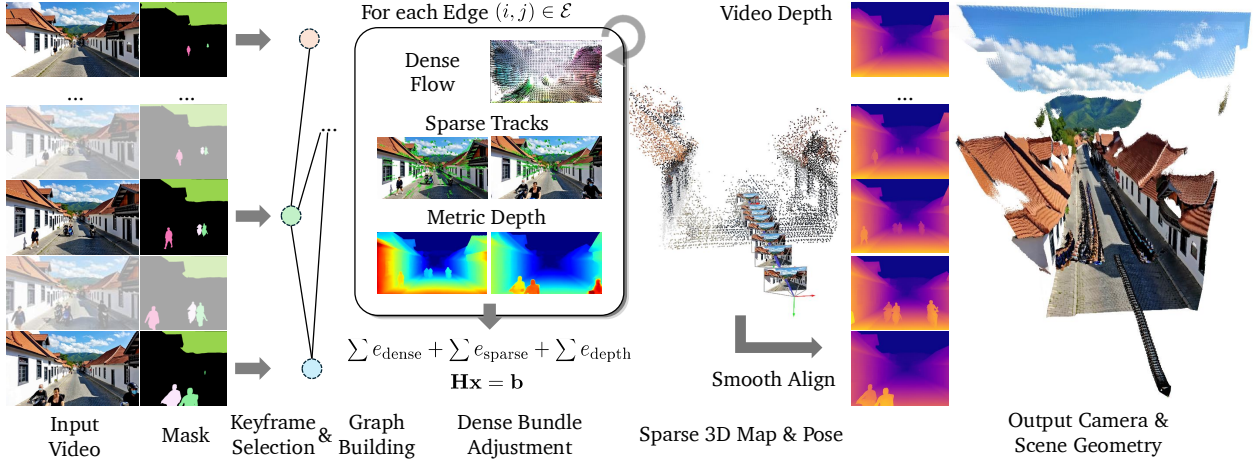
Figure 2: **Pipeline** of ViPE. The system takes a video as input and first estimates the semantic segmentation masks of the movable objects. It then estimates the camera poses, intrinsics, and depth maps from the video by solving a dense bundle adjustment problem incorporating various constraints. The final output is a dense depth map that is consistent with the camera poses and the intrinsics after the smooth depth alignment step.

and as informative inputs at test time. For example, they are currently widely used in training novel view synthesis methods, spanning diffusion-based models [22, 45, 56, 78, 91] and feed-forward reconstruction networks [32, 38]. Camera trajectory information is also demonstrated to be useful in controllable video generation [45, 55, 57, 85] tasks. Additionally, accurate 3D geometric annotations can be helpful in deep-learning-based multi-view stereo (MVS) models [30], and is used for policy evaluation in embodied AI [65] and trajectory understanding [39].

A critical requirement across these applications is that the dataset must be large-scale, diverse in scene types, and contain high-quality geometric and pose annotations. Existing datasets [4, 23, 25, 62] often fall short in these aspects: many are small in scale and limited in diversity (e.g., focused only on indoor environments or constrained by fixed camera rigs). While other real-world datasets exist [33], ours is the first to offer a combination of large-scale, diverse real-world content and high-quality annotations, making it uniquely suited for a broad spectrum of vision and robotics tasks.

3. Methodology

In this section, we present the core methodology of ViPE, with an overview of the pipeline in § 3.1, the core Bundle Adjustment (BA) formulation in § 3.2, and the details of the depth alignment stage to produce the final per-frame depth maps in § 3.3.

3.1. Overview

The pipeline of ViPE is based on a keyframe-based SLAM system for easier scalability and robustness to videos of arbitrary lengths. It generally follows the same frontend and backend design as in most keyframe-based systems, primarily DROID-SLAM [66], consisting of the following steps (as illustrated in Fig. 2):

1. **Intrinsics Initialization:** An initial estimate of the camera intrinsics is obtained by uniformly sampling 4 frames from the video and running them through GeoCalib [67].
2. **Keyframe Selection:** For each incoming frame, we predict the motion from the current frame to the previous keyframe. The motion is a combination of the weighted optical flow from the dense flow network (§ 3.2.1) and the sparse keypoint tracks (§ 3.2.2). If the motion is larger than a pre-defined threshold, we consider

this a keyframe and add it to the BA graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

3. **Frontend Tracking:** For the newly added keyframe, we build a graph $\mathcal{G}$ within a small sliding window of the last several keyframes. Frames within this window are connected by edges either if they are close in observation time step or if the co-visibility is high enough. The energy equation in § 3.2 is then constructed and optimized using a Gauss-Newton solver.

4. **Backend Optimization:** In the backend, a full BA optimization problem is solved that involves all the current keyframes, with the $\mathcal{G}$ built similarly as above. Camera intrinsic parameters are also unlocked for optimization at this stage. We empirically perform this optimization when the number of keyframes reaches 8, 16, and 64, as well as at the end of frontend tracking.

5. **Pose Infilling:** Lastly, for each of the non-keyframes, we obtain the pose by building a small local graph that connects this frame to its closest two keyframes. We add uni-directional edges from the keyframe to the non-keyframe only, hence eliminating the need to compute the metric depth for all the frames during the BA optimization. The procedure is applied in parallel to all non-keyframes.

6. **Dense Depth Estimation:** For each frame, we estimate a dense depth map in the same resolution as the input image that is consistent with the camera pose and the intrinsics. This process will be detailed in § 3.3.

3.2. Formulation

At the core of ViPE, we solve for a BA problem with the following unknown variables: frame poses $\{\mathbf{T}_i \in \mathbb{SE}(3)\}$, camera intrinsics $\boldsymbol{k} \in \mathbb{R}^K$, and a low resolution depth map $\{\mathbf{D}_i \in \mathbb{R}^{h \times w}\}$ for each keyframe i in a graph $\mathcal{G}$:

$$e_{\text{ViPE}}(\{\mathbf{T}_i\}, \{\mathbf{D}_i\}, \boldsymbol{k}) = \sum_{(i,j) \in \mathcal{E}} e_{\text{dense}}(\mathbf{T}_i, \mathbf{T}_j, \mathbf{D}_i, \boldsymbol{k}) + \sum_{(i,j) \in \mathcal{E}} e_{\text{sparse}}(\mathbf{T}_i, \mathbf{T}_j, \mathbf{D}_i, \boldsymbol{k}) + \alpha \sum_{i \in \mathcal{V}} e_{\text{depth}}(\mathbf{D}_i). \quad (1)$$

Here e_{dense} is the term that leverages dense matching between the two frames i and j, while e_{sparse} is a supplement term that leverages sparse keypoint matching information. e_{depth} is a depth regularization term to ensure consistency and robustness of the pose estimation. These terms will be described in detail in §§ 3.2.1 to 3.2.3. The above energy function is minimized with respect to the unknown variables using a Gauss-Newton solver, and since the linear system is intrinsically sparse, we can efficiently solve it with a factorized solver using COLAMD reordering [16]. Since most of the casual videos contain dynamic objects and motions, we describe in § 3.2.4 how we mask the dynamic objects in the video. In § 3.2.5, we describe how the system supports different camera models.

3.2.1. Dense Flow Constraint

The dense flow constraint is formulated as the same as in DROID-SLAM [66], where:

$$e_{\text{dense}}(\mathbf{T}_i, \mathbf{T}_j, \mathbf{D}_i, \boldsymbol{k}) = \sum_{\mathbf{u}} w[\mathbf{u}] \cdot \left\| \Pi_{\boldsymbol{k}}(\mathbf{T}_j^{-1} \mathbf{T}_i \circ \Pi_{\boldsymbol{k}}^{-1}(\mathbf{D}_i[\mathbf{u}])) - \mathbf{u} - \mathbf{F}_{ij}[\mathbf{u}] \right\|^2. \quad (2)$$

Here $\mathbf{u}$ represents the pixel coordinates in the image, and the above equation is summed over all $h \times w$ existing pixels in the depth map $\mathbf{D}$. We choose $h = H/8$ and $w = W/8$ to reduce the number of unknown variables to be optimized. An additional optical flow $\mathbf{F}_{ij}$ is estimated between the two frames i and j, which is regressed from an optical flow network from [66]. Such an optical flow network takes two images as input and outputs a flow map $\mathbf{F}_{ij} \in \mathbb{R}^{h \times w \times 2}$ in the same resolution as the depth map. Internally the network builds a cost volume with an iterative refinement module, and provides a hint to the current estimation with a prior estimated flow $\mathbf{F}_{ij}^{\text{prior}} = \Pi_{\boldsymbol{k}}(\mathbf{T}_j^{-1} \mathbf{T}_i \circ \Pi_{\boldsymbol{k}}^{-1}(\mathbf{D}_i[\mathbf{u}]))$ as the initial guidance of the cost volume. In addition to the flow, a weight map $w[\mathbf{u}]$ is also estimated (detailed in § 3.2.4) to reflect the confidence of the flow estimation as well as the probability of motion, which is less useful for pose estimation.

3.2.2. Sparse Point Constraint

While the estimated dense optical flow $\mathbf{F}_{ij}$ is robust to various camera motions and texture-less scenes, due to its low resolution and network-inference nature, it might miss fine details visible only from the original high-resolution images that are critical for localization. With this in mind, we propose a sparse point constraint based on an off-the-shelf CUDA-based fast feature detection and tracking module from the `cuVSLAM` package [35]. Internally, the features are generated by the Shi-Tomasi corner detector [61] and tracked using the Lucas-Kanade algorithm [46]. These features are computed on the original high-resolution image, providing sub-pixel constraints relative to the network's resolution, hence providing a physically grounded set of accurate flow vectors. The sparse constraint is formulated as:

$$e_{\text{sparse}}(\mathbf{T}_i, \mathbf{T}_j, \mathbf{D}_i, \boldsymbol{k}) = \sum_{\mathbf{p}_i} \left\| \Pi_{\boldsymbol{k}}(\mathbf{T}_j^{-1}\mathbf{T}_i \circ \Pi_{\boldsymbol{k}}^{-1}(\texttt{Bilerp}(\mathbf{D}_i, \mathbf{p}_i))) - \mathbf{p}_j \right\|^2, \tag{3}$$

where $\mathbf{p}_i \in \mathbb{R}^2$ and $\mathbf{p}_j \in \mathbb{R}^2$ are the matched sparse keypoints detected in frame i and j, respectively, and $\texttt{Bilerp}(\mathbf{D}_i, \mathbf{p}_i)$ is the bilinear interpolation of the depth map $\mathbf{D}_i$ at the pixel coordinates $\mathbf{p}_i$ with the prior assumption that the optimal depth map should be smoothly interpolated.

In practice, however, the above term would lead to a semi-sparse Hessian pattern when solving the BA problem since the Jacobian of one e_{sparse} term is related to multiple (up to 4 neighbourhood) pixel locations in $\mathbf{D}$, creating numerous interactions between the depth maps in the graph $\mathcal{G}$ themselves. Although a highly efficient solver is proposed in prior works such as [28], we found it more efficient and effective to use a simpler constraint by replacing the bilinear interpolation with a bilinear splatting operation, yielding the same constraint as in Eq (2) but replacing its depth map $\mathbf{D}$ with $\texttt{Bisplat}(\{\mathbf{p}_j - \mathbf{p}_i\}, \{\mathbf{p}_i\})$, where $\texttt{Bisplat}$ is the bilinear splatting operation that assigns each pixel location $\mathbf{u}$ an accumulated value weighted by the distance to all the input locations $\mathbf{p}_i$.

3.2.3. Depth Regularization

Similarly to the flow correspondences, accurate depth map estimations are typically crucial for resolving ambiguities, especially for small (or degenerate) camera motions. We hence add a depth regularization term as:

$$e_{\text{depth}}(\mathbf{D}_i) = \sum_{\mathbf{u}} m[\mathbf{u}] \cdot \left\| \mathbf{D}_i[\mathbf{u}] - \mathbf{D}_i^{\text{prior}}[\mathbf{u}] \right\|^2, \tag{4}$$

where $\mathbf{D}_i^{\text{prior}}$ is the prior depth map estimated from a pre-trained monocular metric depth estimation network (with m being the estimation uncertainty). We allow the users to choose from different depth estimation models, including Metric3dv2 [27], UniDepthV2 [53], as well as UniK3D [52], depending on the camera models[3]. All these networks are based on single images and provide an estimate of the current scene scale, hence not only help reduce the scale drifting issue commonly seen in SLAM systems, but also provide a good estimate of the real-world metric scale.

Notably, for the metric depth estimation models, the predicted depth maps are conditioned on the camera intrinsics $\boldsymbol{k}$. We hence update the depth predictions after the intrinsics are optimized, and replace the prior depth maps $\mathbf{D}_i^{\text{prior}}$ with the newly predicted ones (for [27] this is a simple scaling operation).

3.2.4. Dynamic Object Masking

Many real-world videos usually have dynamic objects occupying a large pixel region of the video, revealing challenging ambiguities determining the static background that camera poses $\mathbf{T}$ depend on. Most recent state-of-the-art motion segmentation methods (such as [24, 29]) combine semantic priors and optical flow for robustly segmenting the dynamic objects. We hereby take a simpler approach by using pure semantic

---

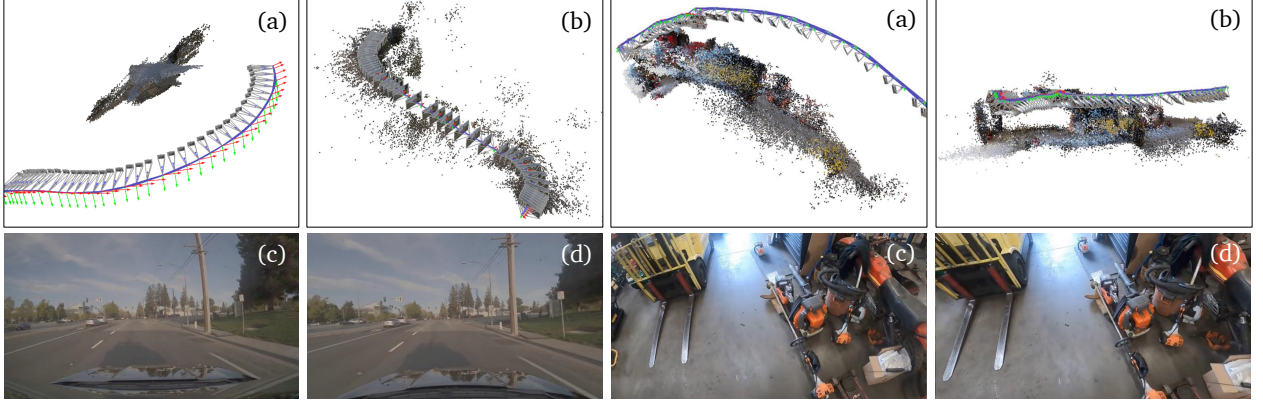[3]In all of our quantitative experiments, we choose Metric3dv2 [27] consistently.

Figure 3: Pose estimation results on **wide-angle cameras**. (a) Baseline results with the pinhole camera assumption. (b) ViPE's results using the unified camera model. (c) Sample frames from the video. (d) Rectified frames using ViPE's intrinsics estimation.

segmentation information for its robustness and efficiency. Specifically, given a list of user-specified semantic classes, we follow SAM-Track [13], and first apply GroundingDINO [42] to provide bounding box prompts to the Segment Anything [34] model, obtaining the segmentation masks of these classes. Instead of applying the above two models on each frame which is computationally expensive, we apply them at a fixed frame interval and propagate the segmentation masks through XMem [12].

The output of the semantic masks is then inverted to obtain the static background mask as $\mathbf{M}$. For Eq (2), we multiply $\mathbf{M}$ with the weight regressed by the dense flow network to obtain the weight map w. Similarly, for Eq (3), we remove all the point tracks outside the region of $\mathbf{M}$ and treat them as outliers.

3.2.5. Handling Different Camera Models

ViPE provides support for various camera models as well as optimization of their intrinsic parameters $\mathbf{k}$. Our system is based on the assumption of the following radial camera formulation:

$$\mathbf{u} = \Pi_{\mathbf{k}} \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) = \begin{bmatrix} f \cdot q_{\mathbf{k}}(\theta) \cdot \cos \phi + W/2 \\ f \cdot q_{\mathbf{k}}(\theta) \cdot \sin \phi + H/2 \end{bmatrix}, \tag{5}$$

where x, y, z are 3D coordinates in the camera local space, $\theta = \arctan \frac{\sqrt{x^2+y^2}}{z}$ is the angle between the corresponding ray and the optical axis, and $\phi = \arctan \frac{y}{x}$ is the rotation angle of the projected point on the canvas. Note that for simplicity we always assume that the principal point is fixed at the image center $(\frac{W}{2}, \frac{H}{2})$, and the focal length f is the same for both axes.

For a simple pinhole camera, we let $q_{\mathbf{k}}(\theta) = \tan \theta$, and f is the only scalar parameter to be estimated in $\mathbf{k}$. For wide-angle/fisheye camera, we follow [26] and use the unified camera model [48] where $q_{\mathbf{k}}(\theta) = \frac{\tan \theta}{1+\alpha\sqrt{\tan^2 \theta+1}}$ and $\mathbf{k} = [f, \alpha] \in \mathbb{R}^2$. Here α controls the distortion strength, with $\alpha = 0$ falling back to the pinhole camera model. Fig. 3 shows examples of the pose estimation results on videos captured by wide-angle cameras.

ViPE's BA-based formulation naturally extends to multi-camera rigs. This is achieved by expanding the transformation $\mathbf{T}_i$ in Eq (1) to $\mathbf{T}_v \mathbf{T}_i$, where $\mathbf{T}_v$ is the transformation from the rig to the v-th camera's reference frame. In [66], the two cameras on a stereo rig are correlated by adding an additional set of edges in the graph that connects the left and right cameras at the same time step. This might become less effective in finding co-visible landmarks if the view frustums of the cameras have little overlap. We hence adaptively add cross-view edges in the graph based on the co-visibility of the cameras measured by projecting the dense depth maps to

the other cameras' views. 360 camera is a special case of such a setting where the panorama image is stitched from two or more fisheye cameras. To tackle these videos, we project the original image into 6 pinhole cameras (facing towards front, back, left, right, up, and bottom directions) roughly covering the cubical surface, and fix the relative transformation $\mathbf{T}_v$ during the optimization.

3.3. Post-processed Dense Depth Alignment

While state-of-the-art dense depth estimation networks [9, 83] can produce high-quality relative depth maps, it is challenging to recover a consistent absolute scale with the estimated camera pose $\{\mathbf{T}_i\}$ across the entire video sequence. On the other hand, the dense depth map $\mathbf{D}$ solved from the bundle adjustment Eq (1) typically has a better alignment with the camera poses, and shows consistency across the entire video sequence. They can be, however, noisy/incomplete (especially in textureless regions) and suffer from low resolutions.

To achieve the best of both worlds, we propose a smooth depth alignment strategy. We first use a video depth estimation network [9] to estimate a temporally smooth yet affine-invariant depth map for each frame i, denoted as $\mathbf{D}_i^{\text{VDA}} \in \mathbb{R}^{H \times W}$. In parallel we aggregate the point cloud unprojected with $\{\mathbf{D}_i\}$ from the BA optimization, filter the pixels that fail the consistency check with the estimated camera poses, and project them back to the image space to obtain a sparse depth map $\mathbf{D}_i^{\text{BA}} \in \mathbb{R}^{H \times W}$. We then make use of a momentum-based update strategy to find the best affine transformation parameters:

$$
\begin{aligned}
\alpha_i, \beta_i &= \underset{\alpha, \beta}{\arg\min} \sum_{\mathbf{u} \text{ is valid}} \left\| \mathbf{M} \cdot (\alpha/\mathbf{D}_i^{\text{VDA}}[\mathbf{u}] + \beta - 1/\mathbf{D}_i^{\text{BA}}[\mathbf{u}]) \right\|_2^2, \\
\hat{\alpha}_i &= m \cdot \hat{\alpha}_{i-1} + (1-m) \cdot \alpha_i, \quad \hat{\beta}_i = m \cdot \hat{\beta}_{i-1} + (1-m) \cdot \beta_i,
\end{aligned}
\tag{6}
$$

where m is the momentum factor, and the final depth map we output is $\mathbf{D}_i^{\text{HD}} = \frac{1}{\hat{\alpha}_i/\mathbf{D}_i^{\text{VDA}}+\hat{\beta}_i}$.

Notably, real-world videos can be diverse in terms of the scene distributions, and the projected depth map $\mathbf{D}_i^{\text{BA}}$ might not have enough information to constrain the above affine transformation. We hence compute the percentage of the pixels that are covered by this depth map, and apply PriorDA [76] to infill the depth map conditioned on the partial observation as well as the input image before assigning this to $\mathbf{D}_i^{\text{BA}}$ for alignment. Under very extreme and rare cases where only a few of or none of the pixels are covered, we directly assign $\mathbf{D}_i^{\text{BA}}$ to be the metric depth estimation from § 3.2.3.

4. Evaluation

To fully demonstrate the capability of ViPE, we compare ourselves to the state-of-the-art methods on the fundamental geometry estimation tasks, including camera intrinsics, poses, and depth estimation across standard benchmarks as well as *in-the-wild* casual videos.

4.1. Camera Pose Estimation

4.1.1. Evaluation on Standard Benchmarks

We first demonstrate our competitive performance by evaluating against established baselines on widely recognized datasets with readily available ground truth.

Datasets. We measure the accuracy of the estimated camera pose and intrinsics on two main scenarios: (1) Indoor scenes represented by the widely used **TUM RGB-D** dataset [62] with multiple loop closures and complicated camera trajectory with scene motions; (2) Outdoor driving scenes. For the latter, we crop the ultra-wide images from the **KITTI** odometry [23] dataset to a resolution of 512×368 and only keep the first 1024 frames for simplicity. Since all the sequences from the KITTI dataset are captured with a fixed camera

	Freiburg1 (static)				Freiburg3 (dynamic)				Run Time†
	ATE (cm) ↓	RTE (cm) ↓	RRE (°) ↓	Focal (°) ↓	ATE (cm) ↓	RTE (cm) ↓	RRE (°) ↓	Focal (°) ↓	
DROID-SLAM† [66]	4.4	0.6	0.39	4.1	2.7	1.0	0.27	4.3	∼2min
MASt3R-SLAM [50]	6.8	2.3	0.54	N/A	7.6	2.7	0.41	N/A	∼1.5min
VGGT [70]	8.4	0.8	0.44	11.1	12.9	0.5	0.30	10.1	∼4min
MegaSAM [37]	7.0	0.6	0.37	10.5	1.5	0.8	0.26	12.6	∼15min
Ours	3.6	0.7	0.39	1.8	1.5	0.8	0.27	0.6	∼3min

†: Run time is measured on a single NVIDIA RTX 5090 GPU, excluding the per-frame dense depth estimation time if possible. The dataset has ∼950 frames per sequence on average.

Table 1: Pose and intrinsics accuracy measured on **TUM-RGBD** [62] dataset.

	KITTI [23]		RDS [1]		Run Time
	ATE (m) ↓	Focal (°) ↓	ATE (m) ↓	Focal (°) ↓	
MASt3R-SLAM [50]	122.2	N/A	21.0	N/A	∼4min
MASt3R-SLAM^{KF} [50]	21.3	N/A	9.5	N/A	-
VGGT [70]	23.8	1.9	5.7	5.9	∼3min
MegaSAM [37]	25.4	2.3	9.3	47.7	∼17min
Ours	9.2	1.9	5.0	7.9	∼4.5min

Table 2: Pose and intrinsics accuracy measured on **outdoor driving** datasets.

intrinsics, to further evaluate the robustness of our method to different camera focal lengths, we supplement with an additional set **RDS**. This dataset consists of a subset of 64 sequences sampled from the original Real Driving Scene (RDS) dataset [1, 2]. We resample the input images to a pinhole with varying focal lengths ranging from 30 to 70 degrees. This further tests the robustness of ViPE on more complicated driving scenarios.

Baselines and metrics. We compare to various state-of-the-art baselines available in the literature at the time of release. Out of them, MASt3R-SLAM [50], VGGT [70] and MegaSAM [37] allow raw video input. For a fair comparison, we executed MASt3R-SLAM in its uncalibrated configuration to acquire camera poses for all frames, mirroring the approach taken by other baselines. Additionally, for comprehensive reference, we present MASt3R-SLAM's metrics, specifically computed on keyframes (MASt3R-SLAM^{KF}), in Tab. 2, adhering to the methodology outlined in its original publication. To facilitate VGGT inference on videos comprising hundreds to thousands of frames, we devised a sliding window strategy. This approach involves running VGGT on each local window, defined by N frames, with a guaranteed overlap of K frames between successive windows. Subsequently, we estimate the similarity transformation to align the adjacent window predictions. This estimation leverages point maps accumulated from the overlapping frames, where we select the top 50% of points based on their confidence scores predicted by VGGT. Our experimental observations suggest that employing larger local window sizes for VGGT generally leads to more accurate results, primarily by reducing error propagation from Sim(3) alignment on dense point maps. Through empirical evaluation, we determined that $N = 120/200$ and $K = 5$ provided the optimal performance when inference was performed on a single GPU. Due to its simplicity and efficiency, we also add DROID-SLAM [66] as a reference baseline, where the camera intrinsics is directly estimated via GeoCalib [67] using the first 2s of video (denoted as DROID-SLAM†).

We compute Absolute Trajectory Error (ATE), Relative Translation/Rotation Error (RTE, RRE), and pinhole intrinsics error (Focal). ATE, RTE, RRE are classical SLAM metrics that measure how the predicted pose deviates from the ground truth after optimal rigid alignment, reflecting the pose quality in terms of both global and local scales [49, 62]. The intrinsics error is computed as the absolute difference between prediction and ground-truth field of view angles.

Results. As quantitatively shown in Tab. 1 and 2, ViPE reaches competitive performance in both indoor and outdoor datasets. The method is also robust to dynamic scenes by properly and efficiently removing movable objects from the camera estimation. Notably, the scale of the output pose is roughly in line with the
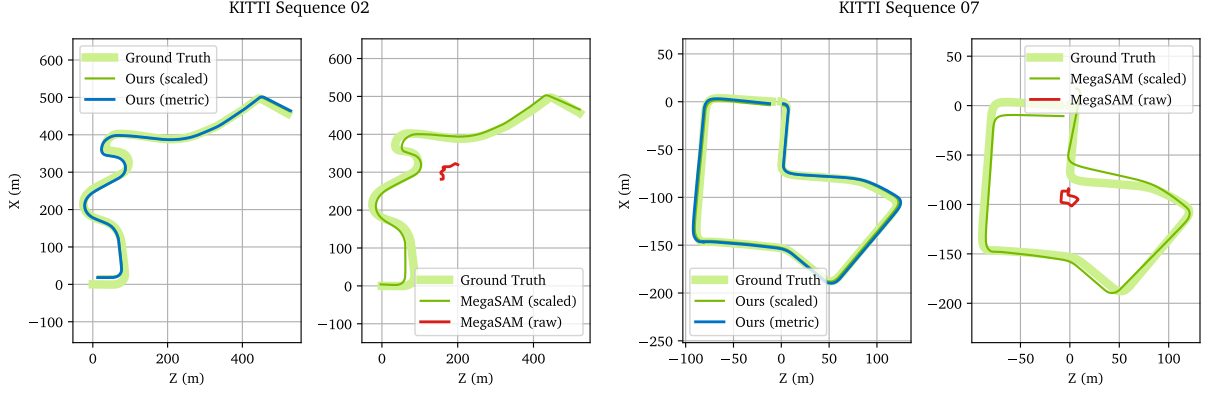
Figure 4: Qualitative results of camera pose estimation on **KITTI dataset** [23]. Output of ViPE can be used as an approximation of the metric scale in real world, while the baseline [37] is not guaranteed to be scale-consistent.

real-world scale thanks to the metric depth prediction module, as demonstrated in Fig. 4, where the baseline, *e.g.* MegaSAM, outputs pose in an indefinite scale space.

### 4.1.2. Evaluation on Unposed Videos

Camera pose consistency metrics. In addition to the common evaluation setup, we further showcase the applicability of ViPE for real-world *in-the-wild* videos, where we are not equipped with ground-truth poses for evaluation. Hence we propose two new metrics for camera pose evaluation without ground-truth annotations.

- **Shuttle Pose Error.** We feed both the video itself and a reversed version of it into the algorithm, obtaining two independent trajectories. Pose (ATE, RRE) and calibration errors (Focal) are then measured on these two sets of estimations, which we denote as S-ATE, S-RTE, and S-Focal. To make sure that the metric numbers are comparable across different baselines, we normalize the lengths of the trajectories to 1 before computing the rigid alignment between them.

- **Sampson Error.** We define the Sampson error as the first-order approximation of the distance from one interest point detected by LightGlue [40] to its corresponding epipolar line in the subsequent frame:

$$\frac{1}{N} \sum_{i=1}^{N-1} \frac{1}{K} \sum_{k=1}^{K} \frac{|\bar{\boldsymbol{y}}_{ik}^{\top} \mathbf{F} \bar{\boldsymbol{x}}_{ik}|}{\sqrt{\|\mathbf{S}\mathbf{F}\bar{\boldsymbol{x}}_{ik}\|_2^2 + \|\mathbf{S}\mathbf{F}^{\top}\bar{\boldsymbol{y}}_{ik}\|_2^2}} \ , \quad \text{where } \mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \tag{7}$$

Here N it the total number of frames and K is the number of correspondences. The correspondences themselves between frame i and frame $i+1$, *i.e.* $\bar{\boldsymbol{x}}_{ik}$ and $\bar{\boldsymbol{y}}_{ik}$, are expressed as homogeneous coordinates. The fundamental matrix $\mathbf{F}$ is computed from the output pinhole intrinsic parameters.

To demonstrate the effectiveness of the above-proposed metrics, we utilize a sub-dataset from § 4.1.1 and compute both the consistency metrics and the standard metrics. As shown in the inset, the proposed focal and pose errors are generally correlated to the standard pose errors (with $r \in (-1, 1)$ denoting the Pearson correlation coefficient).
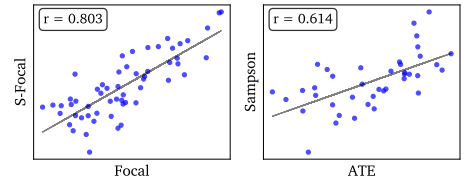
Dataset. We gather two subsets of datasets for benchmarking: (1) **OpenDV** dataset is a random subset of 50 videos from [81]. The dataset contains mainly dashcam videos mounted on a driving vehicle recording the road and various surrounding environments around the globe. These videos are demonstrated to benefit self-driving applications. (2) **VidBench** dataset is a random subset of 60 videos from the Dynpose-100K dataset [58], which contains videos gathered from the web, including various scenes recorded either by hand-held cameras
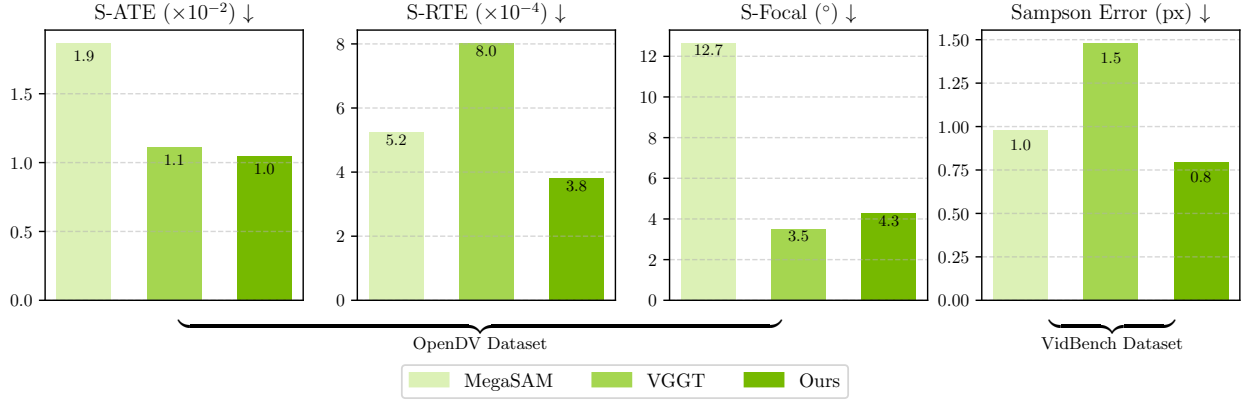
Figure 5: Qualitative results of camera pose estimation on **unposed videos** using the proposed metrics.

	SINTEL [6]			ETH3D [60]		
	RelAbs ↓	LogRMSE ↓	$\delta_{1.25}$ ↑	RelAbs ↓	LogRMSE ↓	$\delta_{1.25}$ ↑
DepthPro [5]	0.29	0.31	62.8	0.31	0.37	64.4
UniDepth [53]	0.23	0.28	79.8	0.19	0.27	72.1
VGGT [70]	0.22	0.36	75.7	0.20	0.28	69.9
MegaSAM [37]	0.29	0.33	67.9	0.23	0.28	64.9
Ours	0.21	0.27	80.8	0.16	0.22	81.7

Table 3: Depth estimation accuracy measured on **synthetic and real-world indoor** datasets.

or professional ones. The videos are diverse in terms of motions and scene distribution, covering both indoor and outdoor scenes.

Results. As shown in Fig. 5, our method reaches better consistency in the shuttle measurement and lower Sampson error, indicating that the estimated camera poses are more reliable. This demonstrates the wide applicability of ViPE in real-world scenarios.

4.2. Depth Estimation

Datasets. We evaluate ViPE for depth estimation on the two well-established benchmarks, *i.e.*, MPI-Sintel [6] (**SINTEL**) dataset and the **ETH3D** [60] dataset. For the SINTEL synthetic dataset, we manually select 6 representative sequences[4], since some contain degenerate camera motions or large sky regions whose depth ground truth is not reliable. For the ETH3D dataset, we eliminate those dark sequences (with names containing 'dark') to avoid large outliers during comparison, resulting in 50 scenes in total.

Baselines and metrics. We use the same setting as in § 4.1 for the VGGT and MegaSAM baselines. Additionally, as a reference, we add monocular metric depth estimation methods including DepthPro [5] and UniDepth [53], where the depth maps are obtained by applying the model on each frame independently. Following the practice in [73], we use the relative absolute depth error (RelAbs), log of depth RMSE error (LogRMSE), and relative depth ratio error ($\delta_{1.25}$).

Results. Quantitative results are shown in Tab. 3 on the two datasets. Empirically, we found that although the depth maps coming from the monocular depth estimation methods have better metrics, there is usually non-negligible jittering across frames. ViPE does not suffer from this issue due to the use of the video depth model. As shown in Fig. 6, after accumulating multiple unprojected point clouds into the 3D world, the

---

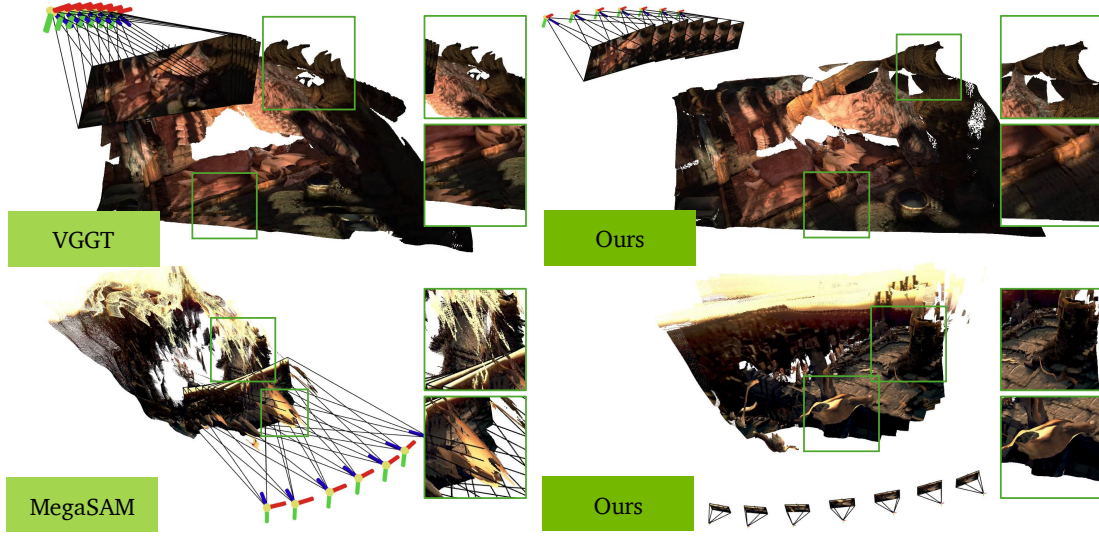[4]Sequences are `alley_2`, `bamboo_1`, `bamboo_2`, `sleeping_1`, `sleeping_2`, `temple_2`.

Figure 6: **Qualitative comparisons** of the method output with the baselines on the **SINTEL** dataset. We subsample the camera frames for clarity only in the visualization.

e_{dense}	e_{sparse}	e_{depth}	Masking	OpenDV [81]				VidBench [58]
				S-ATE ($\times 10^{-2}$) $\downarrow$	S-RTE ($\times 10^{-4}$) $\downarrow$	S-RRE (°) $\downarrow$	S-Focal (°) $\downarrow$	Sampson (px) $\downarrow$
✓				1.39	4.40	0.04	5.28	1.40
✓		✓		1.40	4.45	0.04	5.20	1.36
✓	✓	✓		1.35	4.21	0.04	5.00	0.84
✓		✓	✓	1.13	4.10	0.03	4.45	0.96
✓	✓	✓	✓	**1.05**	**3.80**	**0.03**	**4.26**	**0.83**

Table 4: **Ablation study** on the effectiveness of different components in ViPE.

baselines have multi-layer artifacts due to the inaccurate depth estimation or camera parameter estimation, and ViPE is generally robust under these settings.

4.3. Ablation Study

In Tab. 4, we show the effectiveness of different components introduced in § 3.2. We use the **OpenDV** and **VidBench** datasets as introduced in § 4.1.2 for evaluation since the videos are more reflective of real-world scenarios. Experimental results show that adding sparse track terms and dynamic masking improves the robustness of the estimation, while the use of the depth estimation module in the pipeline is able to further improve the accuracy.

5. Dataset Release

Overview. To address the scarcity of high-quality, diverse, and large-scale datasets for 3D geometric perception in unconstrained environments, and to facilitate future research in this field and its downstream applications, we introduce and release three new datasets annotated with ViPE's camera poses and geometric information. These datasets span a wide range of video sources and content, providing high diversity for robust visual learning. These include:

- **Dynpose-100K++:** Dynpose-100K [58] is a dataset containing ∼100K real-world videos gathered and filtered from the Internet. The videos are originally taken from the PANDA-70M [10] dataset and several

filters have been used on top to filter out sequences that are not suitable for camera pose estimation. However, the dataset has its pose annotated with a Structure-from-Motion pipeline whose camera poses are provided at a lower framerate (12FPS) than the actual video. Furthermore, no per-frame geometry is provided, making verification and evaluation of the quality challenging. We hence re-annotate the dataset (hence the name '++') using our approach, resulting in 99,501 videos with 15.7M frames spanning $\sim$150 hours in total.

- **Wild-SDG-1M**: Recently video diffusion models [1, 68] have demonstrated impressive quality given text prompts. Compared to real-world videos, the videos generated with state-of-the-art diffusion models, with well chosen prompts, are often clear and of high quality, reducing the need for further filtering. We sampled $\sim$1 million videos from the video diffusion models using our in-house curated and balanced text prompts, and annotated all the sampled frames using ViPE, resulting in $\sim$78 million frames in total.
- **Web360**: Web360 is a relatively small-scale dataset containing 360-degree panorama videos curated by the authors of [72]. The dataset contains approximately 2,000 videos in ERP format from the Internet and games. We release per-frame camera poses and distance maps for this dataset.

Impact. These newly released datasets offer a valuable resource for downstream developers – dataset scale and diversity—spanning real-world dynamic internet videos, synthetic environments, and specialized panoramic content—make them well-suited for training and evaluating 3D geometric perception models under a variety of challenging conditions. We hope that this release contributes to advancing downstream real-world applications.

Qualitative Results. To visually demonstrate the consistency and robustness of ViPE's annotations across highly diverse video types, we present a selection of qualitative results from our newly released datasets in Fig. 7 to 9. Note that in the examples we showcase the quality that ViPE consistently achieves, particularly in conditions where other methods may struggle or yield incomplete estimations.

6. Conclusion

In this work, we present ViPE, a video pose estimation engine that estimates camera poses, intrinsics, and depth maps from videos. The system is built upon a bundle adjustment framework that integrates both dense and sparse constraints, leveraging the strengths of both optical flow and keypoint tracking. We also introduce a depth alignment strategy to ensure consistent depth maps across frames. Our method is benchmarked on a variety of datasets, including both static/dynamic and indoor/outdoor scenes, and shows superior performance compared to existing methods.

In practice, ViPE has seen wide adoption across downstream applications with notable impact. It has been used to annotate training data and produce conditional buffers for world generation in Gen3C [57] and Cosmos [1]. The annotated datasets have also supported BTimer [38] in reconstructing 3DGS in a feed-forward manner, improving its robustness to diverse inputs. We hope our released dataset will continue to drive advances in 3D geometric perception and related fields.
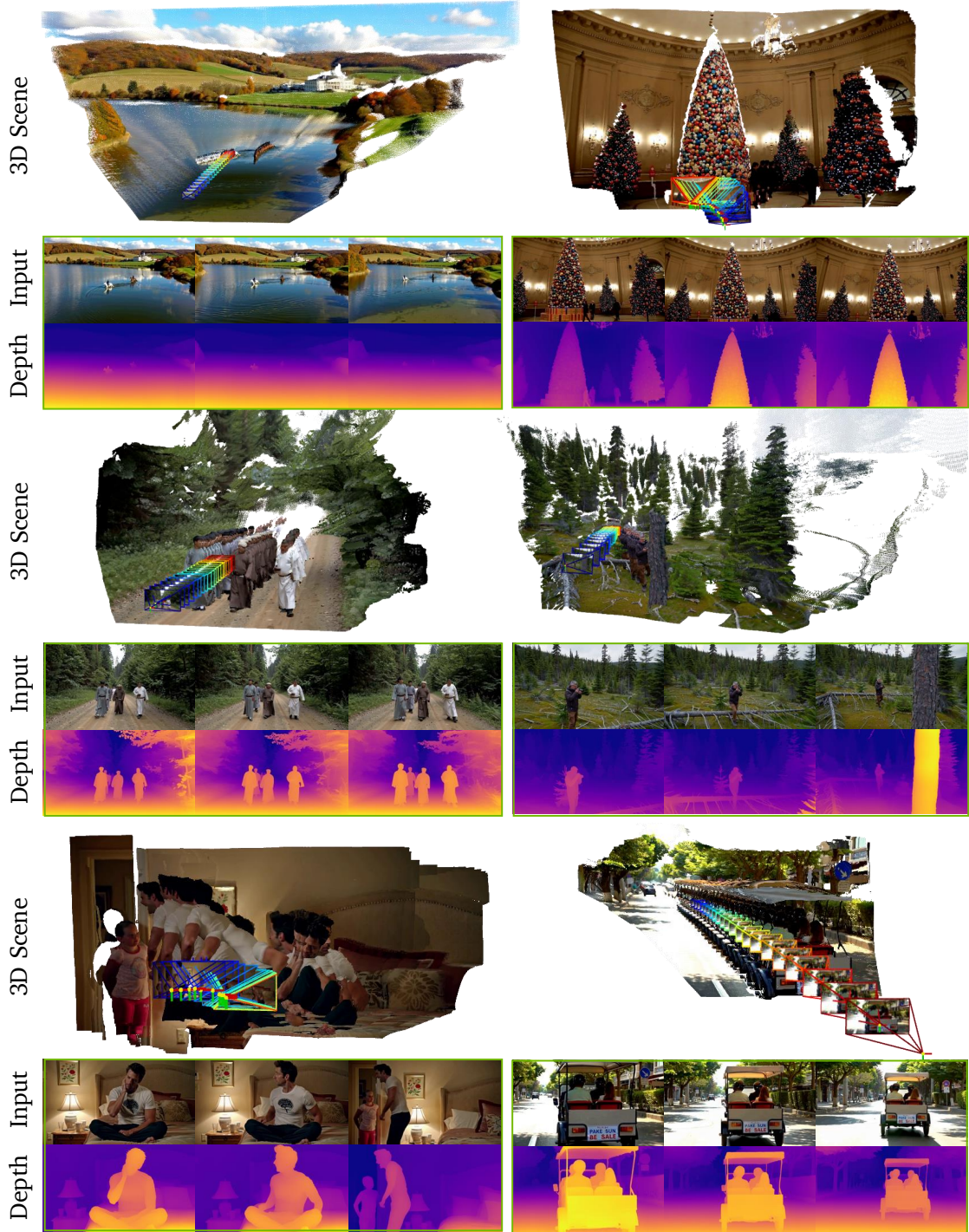
Figure 7: Qualitative demonstration of **Wild-SDG-1M** dataset annotated using ViPE. For the upper 3D scene, we accumulate the point clouds for both the static and dynamic parts of the scene, while the lower rows show the input samples from the videos and the corresponding estimated depth maps.
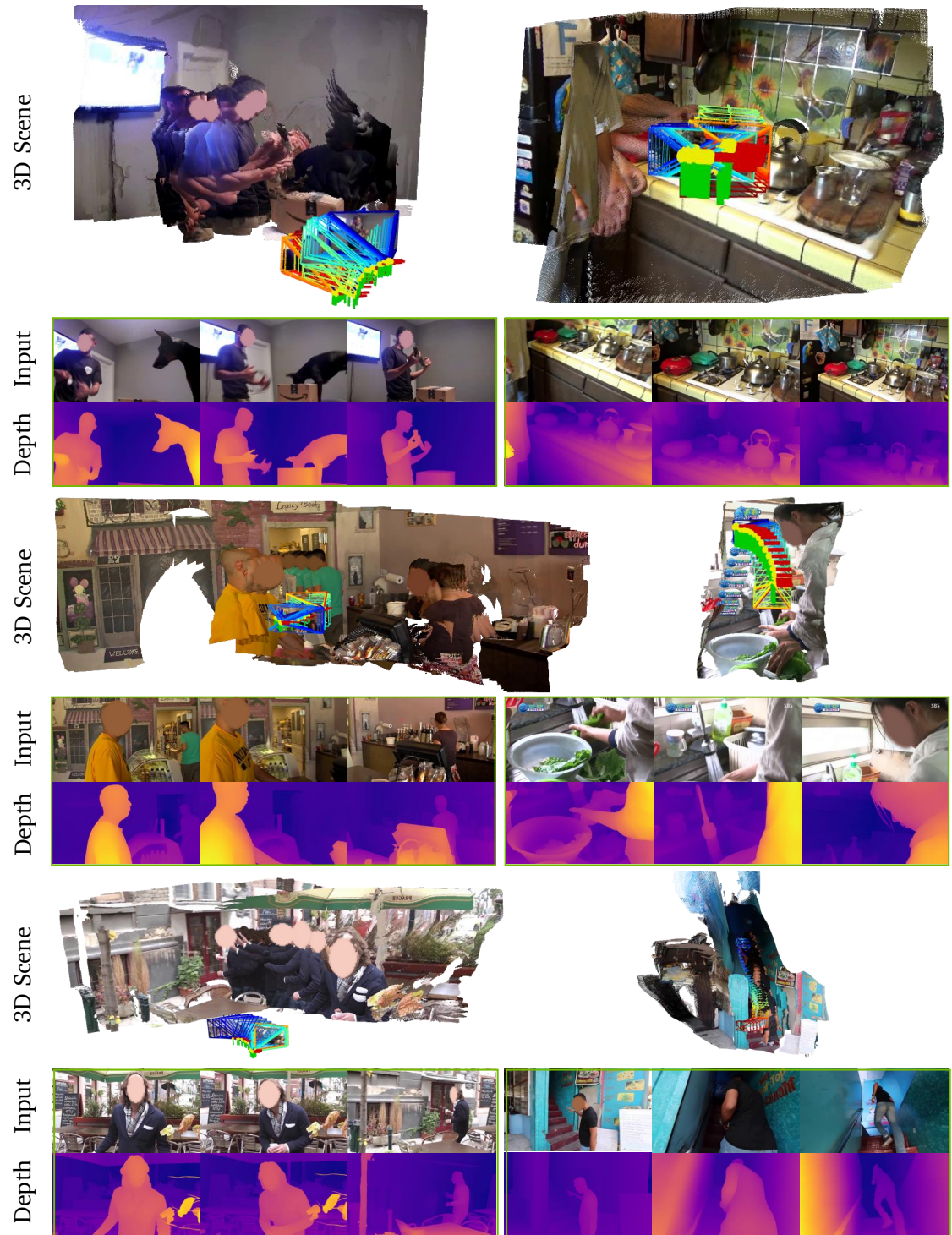
Figure 8: Qualitative demonstration of **DynPose-100K++** dataset annotated using ViPE.
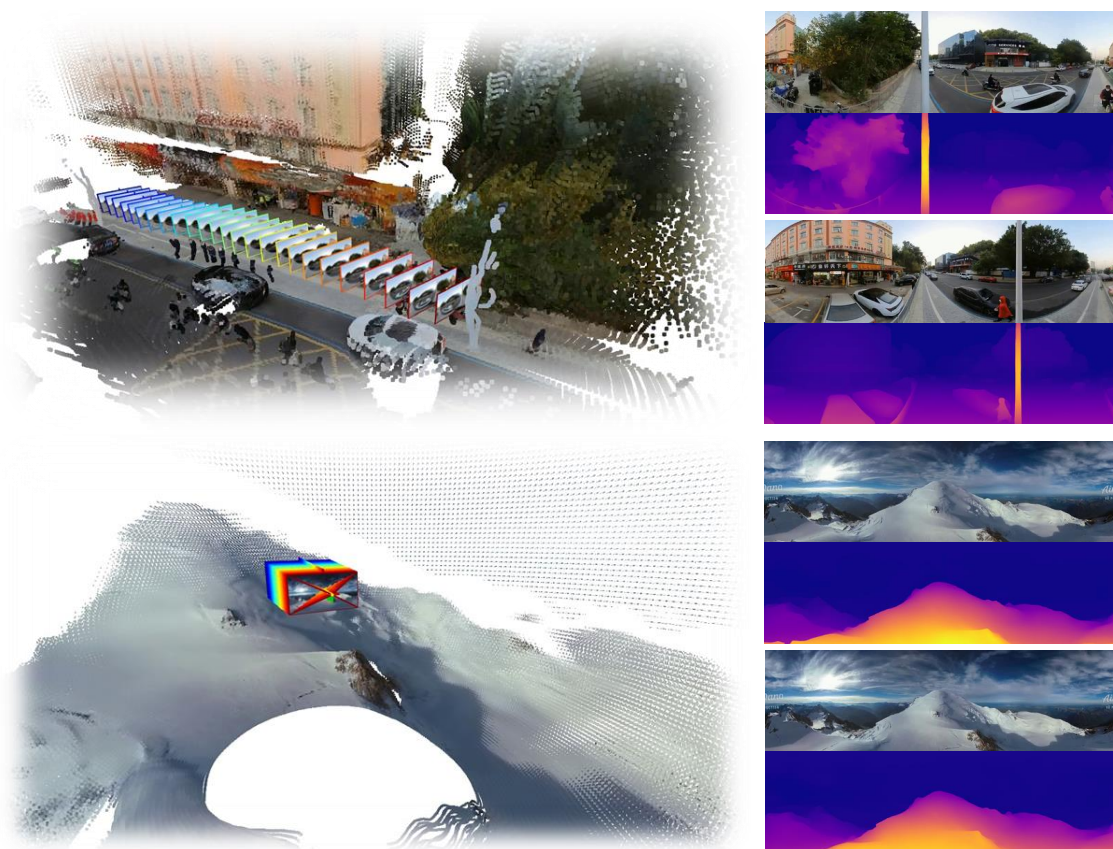
Figure 9: Qualitative demonstration of **Web360** dataset annotated using ViPE.

References

[1] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 9, 13

[2] H. A. Alhaija, J. Alvarez, M. Bala, T. Cai, T. Cao, L. Cha, J. Chen, M. Chen, F. Ferroni, S. Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025. 9

[3] A. Badki, H. Su, B. Wen, and O. Gallo. L4p: Low-level 4d vision perception unified. *arXiv preprint arXiv:2502.13078*, 2025. 3

[4] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 4

[5] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 11

[6] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012. 11

[7] Y. Cabon, L. Stoffl, L. Antsfeld, G. Csurka, B. Chidlovskii, J. Revaud, and V. Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1050–1060, 2025. 3

[8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021. 2, 3

[9] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang. Video depth anything: Consistent depth estimation for super-long videos. *arXiv preprint arXiv:2501.12375*, 2025. 3, 8

[10] T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024. 12

[11] X. Chen, Y. Chen, Y. Xiu, A. Geiger, and A. Chen. Easi3r: Estimating disentangled motion from dust3r without training. *arXiv preprint arXiv:2503.24391*, 2025. 3

[12] H. K. Cheng and A. G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 7

[13] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 7

[14] G. Chou, W. Xian, G. Yang, M. Abdelfattah, B. Hariharan, N. Snavely, N. Yu, and P. Debevec. Flashdepth: Real-time streaming video depth estimation at 2k resolution. *arXiv preprint arXiv:2504.07093*, 2025. 3

[15] W. Cong, Y. Liang, Y. Zhang, Z. Yang, Y. Wang, B. Ivanovic, M. Pavone, C. Chen, Z. Wang, and Z. Fan. E3d-bench: A benchmark for end-to-end 3d geometric foundation models. *arXiv preprint arXiv:2506.01933*, 2025. 2

[16] T. A. Davis, J. R. Gilbert, S. I. Larimore, and E. G. Ng. Algorithm 836: Colamd, a column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 30(3):377–380, 2004. 5

[17] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. 2, 3

[18] B. Duisterhof, L. Zust, P. Weinzaepfel, V. Leroy, Y. Cabon, and J. Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. *arXiv preprint arXiv:2409.19152*, 2024. 3

[19] S. Elflein, Q. Zhou, and L. Leal-Taixé. Light3r-sfm: Towards feed-forward structure-from-motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16774–16784, 2025. 3

[20] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 3

[21] H. Feng, J. Zhang, Q. Wang, Y. Ye, P. Yu, M. J. Black, T. Darrell, and A. Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. *arXiv preprint arXiv:2504.13152*, 2025. 3

[22] R. Gao, A. Holynski, P. Henzler, A. Brussee, R. Martin-Brualla, P. Srinivasan, J. T. Barron, and B. Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. 4

[23] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 4, 8, 9, 10

[24] L. Goli, S. Sabour, M. Matthews, M. Brubaker, D. Lagun, A. Jacobson, D. J. Fleet, S. Saxena, and A. Tagliasacchi. Romo: Robust motion segmentation improves structure from motion. *arXiv preprint arXiv:2411.18650*, 2024. 6

[25] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022. 4

[26] A. Hagemann, M. Knorr, and C. Stiller. Deep geometry-aware camera self-calibration from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3438–3448, 2023. 7

[27] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3, 6

[28] J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, and F. Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 6

[29] N. Huang, W. Zheng, C. Xu, K. Keutzer, S. Zhang, A. Kanazawa, and Q. Wang. Segment any motion in videos. *arXiv preprint arXiv:2503.22268*, 2025. 6

[30] S. Izquierdo, M. Sayed, M. Firman, G. Garcia-Hernando, D. Turmukhambetov, J. Civera, O. Mac Aodha, G. Brostow, and J. Watson. Mvsanywhere: Zero-shot multi-view stereo. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11493–11504, 2025. 4

[31] Z. Jiang, C. Zheng, I. Laina, D. Larlus, and A. Vedaldi. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. *arXiv preprint arXiv:2504.07961*, 2025. 3

[32] H. Jin, H. Jiang, H. Tan, K. Zhang, S. Bi, T. Zhang, F. Luan, N. Snavely, and Z. Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024. 4

[33] L. Jin, R. Tucker, Z. Li, D. Fouhey, N. Snavely, and A. Holynski. Stereo4d: Learning how things move in 3d from internet stereo videos. *arXiv preprint arXiv:2412.09621*, 2024. 3, 4

[34] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 7

[35] A. Korovko, D. Slepichev, A. Efitorov, A. Dzhumamuratova, V. Kuznetsov, H. Rabeti, and J. Biswas. cuvslam: Cuda accelerated visual odometry. *arXiv preprint arXiv:2506.04359*, 2025. 3, 6

[36] V. Leroy, Y. Cabon, and J. Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 2, 3

[37] Z. Li, R. Tucker, F. Cole, Q. Wang, L. Jin, V. Ye, A. Kanazawa, A. Holynski, and N. Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10486–10496, 2025. 2, 3, 9, 10, 11

[38] H. Liang, J. Ren, A. Mirzaei, A. Torralba, Z. Liu, I. Gilitschenski, S. Fidler, C. Oztireli, H. Ling, Z. Gojcic, et al. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. *arXiv preprint arXiv:2412.03526*, 2024. 4, 13

[39] Z. Lin, S. Cen, D. Jiang, J. Karhade, H. Wang, C. Mitra, T. Ling, Y. Huang, S. Liu, M. Chen, et al. Towards understanding camera motions in any video. *arXiv preprint arXiv:2504.15376*, 2025. 4

[40] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 10

[41] S. Liu, W. Li, P. Qiao, and Y. Dou. Regist3r: Incremental registration with stereo foundation model. *arXiv preprint arXiv:2504.12356*, 2025. 3

[42] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. 7

[43] Y. Liu, S. Dong, S. Wang, Y. Yin, Y. Yang, Q. Fan, and B. Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16651–16662, 2025. 3

[44] J. Lu, T. Huang, P. Li, Z. Dou, C. Lin, Z. Cui, Z. Dong, S.-K. Yeung, W. Wang, and Y. Liu. Align3r: Aligned monocular depth estimation for dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22820–22830, 2025. 3

[45] Y. Lu, X. Ren, J. Yang, T. Shen, Z. Wu, J. Gao, Y. Wang, S. Chen, M. Chen, S. Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. *arXiv preprint arXiv:2412.03934*, 2024. 4

[46] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981. 6

[47] D. Maggio, H. Lim, and L. Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025. 2, 3

[48] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3945–3950. IEEE, 2007. 7

[49] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2, 3, 9

[50] R. Murai, E. Dexheimer, and A. J. Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 2, 3, 9

[51] L. Pan, D. Baráth, M. Pollefeys, and J. L. Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024. 3

[52] L. Piccinelli, C. Sakaridis, M. Segu, Y.-H. Yang, S. Li, W. Abbeloos, and L. Van Gool. Unik3d: Universal camera monocular 3d estimation. *arXiv preprint arXiv:2503.16591*, 2025. 3, 6

[53] L. Piccinelli, C. Sakaridis, Y.-H. Yang, M. Segu, S. Li, W. Abbeloos, and L. Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025. 3, 6, 11

[54] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. Torr, and D. W. Murray. Infinitam v3: A framework for large-scale 3d reconstruction with loop closure. *arXiv preprint arXiv:1708.00783*, 2017. 2

[55] X. Ren, Y. Lu, T. Cao, R. Gao, S. Huang, A. Sabour, T. Shen, T. Pfaff, J. Z. Wu, R. Chen, et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025. 4

[56] X. Ren, Y. Lu, H. Liang, Z. Wu, H. Ling, M. Chen, S. Fidler, F. Williams, and J. Huang. Scube: Instant large-scale scene reconstruction using voxsplats. *Advances in Neural Information Processing Systems*, 37:97670–97698, 2024. 4

[57] X. Ren, T. Shen, J. Huang, H. Ling, Y. Lu, M. Nimier-David, T. Müller, A. Keller, S. Fidler, and J. Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6121–6132, 2025. 4, 13

[58] C. Rockwell, J. Tung, T.-Y. Lin, M.-Y. Liu, D. F. Fouhey, and C.-H. Lin. Dynamic camera poses and where to find them. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12444–12455, 2025. 10, 12

[59] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[60] T. Schops, T. Sattler, and M. Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019. 11

[61] J. Shi et al. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994. 6

[62] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 4, 8, 9

[63] E. Sucar, Z. Lai, E. Insafutdinov, and A. Vedaldi. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. *arXiv preprint arXiv:2503.16318*, 2025. 3

[64] Z. Tang, Y. Fan, D. Wang, H. Xu, R. Ranjan, A. Schwing, and Z. Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5283–5293, 2025. 3

[65] A. Team, H. Zhu, Y. Wang, J. Zhou, W. Chang, Y. Zhou, Z. Li, J. Chen, C. Shen, J. Pang, et al. Aether: Geometric-aware unified world modeling. *arXiv preprint arXiv:2503.18945*, 2025. 4

[66] Z. Teed and J. Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 4, 5, 7, 9

[67] A. Veicht, P.-E. Sarlin, P. Lindenberger, and M. Pollefeys. Geocalib: Learning single-image calibration with geometric optimization. In *European Conference on Computer Vision*, pages 1–20. Springer, 2024. 4, 9

[68] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 13

[69] H. Wang and L. Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024. 3

[70] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 3, 9, 11

[71] J. Wang, N. Karaev, C. Rupprecht, and D. Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024. 3

[72] Q. Wang, W. Li, C. Mou, X. Cheng, and J. Zhang. 360dvd: Controllable panorama video generation with 360-degree video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6913–6923, 2024. 13

[73] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025. 3, 11

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

[75] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He. pi3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 3

[76] Z. Wang, S. Chen, L. Yang, J. Wang, Z. Zhang, H. Zhao, and Z. Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025. 3, 8

[77] F. Wimbauer, W. Chen, D. Muhle, C. Rupprecht, and D. Cremers. Anycam: Learning to recover camera poses and intrinsics from casual videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16717–16727, 2025. 3

[78] R. Wu, R. Gao, B. Poole, A. Trevithick, C. Zheng, J. T. Barron, and A. Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26057–26068, 2025. 4

[79] Y. Xiao, J. Wang, N. Xue, N. Karaev, Y. Makarov, B. Kang, X. Zhu, H. Bao, Y. Shen, and X. Zhou. Spatialtrackerv2: 3d point tracking made easy. *arXiv preprint arXiv:2507.12462*, 2025. 3

[80] T.-X. Xu, X. Gao, W. Hu, X. Li, S.-H. Zhang, and Y. Shan. Geometrycrafter: Consistent geometry estimation for open-world videos with diffusion priors. *arXiv preprint arXiv:2504.01016*, 2025. 3

[81] J. Yang, S. Gao, Y. Qiu, L. Chen, T. Li, B. Dai, K. Chitta, P. Wu, J. Zeng, P. Luo, et al. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14662–14672, 2024. 10, 12

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

[83] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 8

[84] W. Ye, X. Chen, R. Zhan, D. Huang, X. Huang, H. Zhu, H. Bao, W. Ouyang, T. He, and G. Zhang. Datap-sfm: Dynamic-aware tracking any point for robust structure from motion in the wild. *arXiv preprint arXiv:2411.13291*, 2024. 3

[85] M. YU, W. Hu, J. Xing, and Y. Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. *arXiv preprint arXiv:2503.05638*, 2025. 4

[86] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 3

[87] S. Zhang, Y. Ge, J. Tian, G. Xu, H. Chen, C. Lv, and C. Shen. Pomato: Marrying pointmap matching with temporal motion for dynamic 3d reconstruction. *arXiv preprint arXiv:2504.05692*, 2025. 3

[88] Z. Zhang, F. Cole, Z. Li, M. Rubinstein, N. Snavely, and W. T. Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022. 3

[89] Q. Zhao, A. Lin, J. Tan, J. Y. Zhang, D. Ramanan, and S. Tulsiani. Diffusionsfm: Predicting structure and motion via ray origin and endpoint diffusion. *arXiv preprint arXiv:2505.05473*, 2025. 3

[90] W. Zhao, S. Liu, H. Guo, W. Wang, and Y.-J. Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *European Conference on Computer Vision*, pages 523–542. Springer, 2022. 3

[91] J. J. Zhou, H. Gao, V. Voleti, A. Vasishta, C.-H. Yao, M. Boss, P. Torr, C. Rupprecht, and V. Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. 4