

Stereo4D: Learning How Things Move in 3D from Internet Stereo Videos

Linyi Jin^{1,2} Richard Tucker¹ Zhengqi Li¹ David Fouhey³
 Noah Snavely^{1*} Aleksander Holynski^{1*}

¹Google DeepMind ²University of Michigan ³New York University *equal contribution

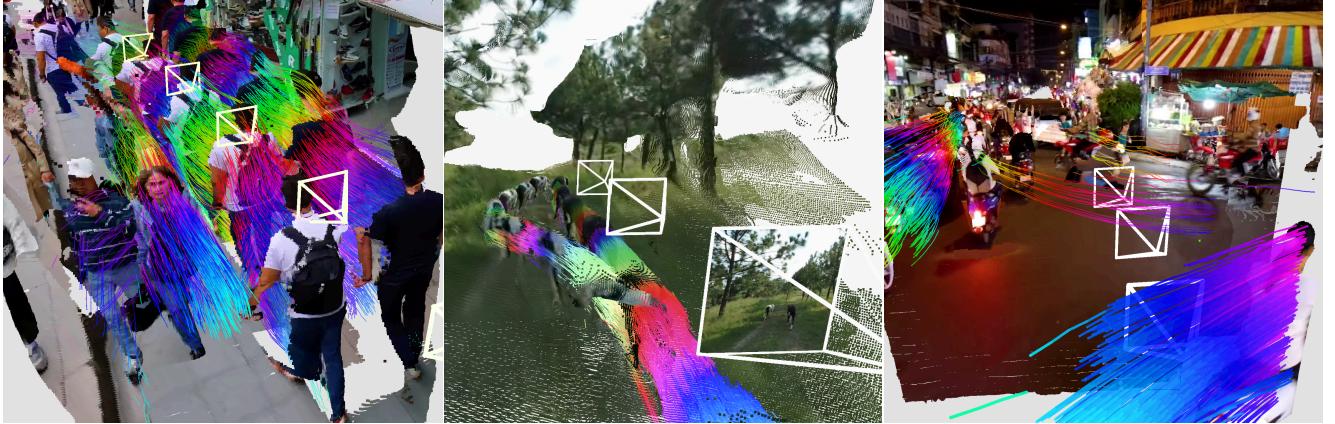


Figure 1. There is currently no scalable source of data for real-world, ground truth 3D motion paired with video. We present a framework for mining such data from existing stereoscopic videos on the Internet, in the form of 3D point clouds with long-range world-space trajectories. Our framework fuses and filters camera poses, dense depth maps, and 2D motion trajectories to produce high-quality, pseudo-metric point clouds with long-term 3D motion trajectories, pictured above, for hundreds of thousands of video clips. We show how this data is useful in learning a model that reasons about both 3D shape and motion in imagery.

Abstract

Learning to understand dynamic 3D scenes from imagery is crucial for applications ranging from robotics to scene reconstruction. Yet, unlike other problems where large-scale supervised training has enabled rapid progress, directly supervising methods for recovering 3D motion remains challenging due to the fundamental difficulty of obtaining ground truth annotations. We present a system for mining high-quality 4D reconstructions from internet stereoscopic, wide-angle videos. Our system fuses and filters the outputs of camera pose estimation, stereo depth estimation, and temporal tracking methods into high-quality dynamic 3D reconstructions. We use this method to generate large-scale data in the form of world-consistent, pseudo-metric 3D point clouds with long-term motion trajectories. We demonstrate the utility of this data by training a variant of DUSi3R to predict structure and 3D motion from real-world image pairs, showing that training on our reconstructed data enables generalization to diverse real-world scenes. Project page: <https://stereo4d.github.io>

1. Introduction

Simultaneously predicting and understanding geometry and motion—that is, dynamic 3D content—from images is a fundamental building block for computer vision, with applications ranging from robotic interaction and scene reconstruction to novel view synthesis of dynamic scenes. While recent work has made remarkable progress in predicting static 3D structure from images [5, 99, 105], modeling real-world 3D motion—people gesturing, balls bouncing, leaves rustling in the wind—remains a critical unsolved challenge for building truly general models of the visual world.

Recent breakthroughs in AI, from large language models [1, 89] to image generation [73] and static 3D reconstruction [5, 99, 105], demonstrate a consistent pattern: large amounts of high-quality, realistic training data and scalable architectures enable dramatic performance improvements. In the realm of 3D reasoning, prior works [49, 74, 75, 99, 104] have shown the value of large-scale training data for strong zero-shot generalization within single-view or two-view static scene settings. But applying this same formula to *dynamic* 3D scenes (i.e. moving 3D

structure) requires a corresponding large-scale dataset consisting of diverse visual content paired with corresponding ground-truth 3D motion trajectories. Obtaining such data presents unique challenges. While there are synthetic datasets [9, 19, 29, 115], these often fail to capture the distribution of real-world content and the nuanced patterns of real-world motion. Traditional approaches to gathering real motion data, such as motion capture systems or multi-view camera arrays [28, 35, 38, 43] are accurate, but difficult to scale and limited in the diversity of scenes they can capture.

We identify online stereoscopic fisheye videos (often referred to as VR180 videos) as an untapped source of such data. These videos, designed to capture immersive VR experiences, provide wide field-of-view stereo imagery with a standardized stereo baseline. We present a pipeline that carefully combines state-of-the-art methods for stereo depth estimation and video tracking along with structure-from-motion methods optimized for dynamic scenes. By combining our system with careful filtering and quality control, we show that we can extract over 100K video sequences, each containing high-quality 3D point clouds with per-point long-term trajectories (see Fig. 1), as well as all other intermediate quantities: depth maps, camera poses, images, and 2D correspondences. We additionally show the utility of the dataset by training *DynaDUS3R*, an extension to DUS3R that can predict high-quality 3D structure *and* motion from challenging image pairs.

Our contributions include: (1) a framework for obtaining real-world, dynamic, and pseudo-metric 4D reconstructions and camera poses at scale from existing online video; (2) DynaDUS3R, a method that takes a pair of frames from any real-world video, and predicts a pair of 3D point clouds and the corresponding 3D motion trajectories that connect them in time.

2. Related work

2D and 3D motion data. There has been tremendous progress on the task of motion estimation from images and videos, and in particular for 2D image-space correspondence estimation. Most state-of-the-art methods use neural networks trained on ground truth data to predict these correspondences directly from images. While these approaches require large training datasets, synthetic data from graphics engines [9, 19, 29, 30, 59, 83, 115] has proven surprisingly effective at generalizing to real-world data, likely because the core task, low-level textural correspondence, is similar between the two domains.

However, the same cannot be said for 3D motion estimation, since predicting both 3D structure and motion is usually more ambiguous and can require specific prior knowledge about the real world and how it moves. To help address this domain gap, a number of real-world datasets have been proposed. The KITTI [27] and Waymo [86] datasets in-

clude real-world autonomous driving sequences with stereo and motion annotations derived from LiDAR and odometry information, but only focus on the relatively closed domain of street scenes, whereas our data depicts more diverse in-the-wild scenarios. A number of annotated smaller-scale datasets, such as TAPVid [16], TAPVid3D [46], and Dycheck [25], have been proposed, primarily serving as evaluation datasets for benchmarking depth estimation, 3D reconstruction, and 3D motion estimation approaches. WSVD [97] and NVDS [100] are stereo video datasets that include disparity maps derived from optical flow. While their source content is similar, our method provides richer 3D annotations beyond time-independent disparity maps, such as 3D camera parameters and long-term 3D motion tracks.

Static and dynamic scene reconstruction The problem of reconstructing a static 3D scene has been studied for decades. Traditional 3D reconstruction methods tackle this problem by first estimating camera parameters via Structure-from-Motion (SfM) [2, 32, 71, 72, 72, 76, 82, 87] or SLAM [11, 15, 20, 60]. Dense scene geometry can then be estimated through Multi-view Stereo (MVS) [10, 22–24, 36, 77, 106, 107] followed by surface reconstruction algorithms [14, 33, 40]. More recently, deep neural network-based approaches have shown promising results in improving camera localization accuracy or scene reconstruction through intermediate representations such as depth maps [4, 54, 79, 88, 91, 93], radiance fields [21, 26, 55, 69, 80, 102], or 3D scene coordinates [7, 8, 48, 99, 111]. However, these methods assume the input images to be observations of a static environment, and therefore produce inaccurate geometry and camera poses for dynamic scenes.

Reconstructing dynamic scenes is more challenging since scene and object motions violate the multi-view constraints used to reconstruct static scenes. As a result, many prior works require RGBD input [6, 62] or only recover sparse geometry [66, 81, 96]. Several recent works tackle this problem from monocular input through video depth maps [45, 113, 114], time-varying radiance fields [25, 47, 51, 52, 56, 67, 68, 98], or generative priors [103].

Monocular and stereo depth. Recent works on single-view depth prediction have shown strong zero-shot generalization to in-the-wild domains by training deep neural networks on diverse RGBD datasets [41, 49, 50, 70, 74, 75, 104, 105, 108, 109]. However, producing *temporally consistent* and *metric* depth from video is still challenging. To tackle this, recent works use test-time optimization [58, 114] or end-to-end learning with temporal attention [34, 45, 78, 100]. On the other hand, stereo images or videos are also popular input modalities for obtaining reliable metric depth maps, and various stereo matching algorithms have been proposed [3, 12, 31, 37, 39, 42, 44, 53, 65, 85, 94, 101, 110, 112]. Building on these advancements,

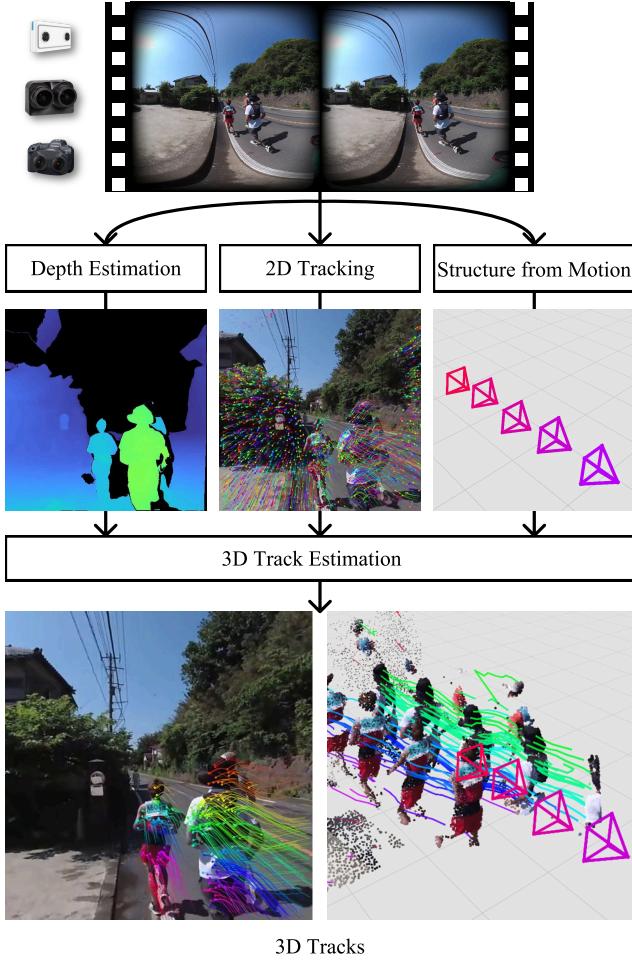


Figure 2. Data processing pipeline. Our method starts with VR180 (wide-angle, stereoscopic) videos, and estimates metric stereo depth, 2D point tracks, and camera poses. These quantities allow the tracks to be lifted to 3D where they are filtered and denoised to produce world-space, metric 3D point trajectories.

our method bridges ideas from monocular video depth estimation and stereo video processing. We use a light-weight optimization step and extend them to stereo inputs for more consistent motion estimation in metric space.

3. Creating a dataset of 4D scenes

A core contribution of this work is a pipeline for extracting high-quality, pseudo-metric, 3D data from online stereoscopic fisheye videos (known as VR180 videos). High-resolution, wide field of view VR180 videos can be found readily online. We show that this data is ideal for deriving rich dynamic 3D information that can power models for predicting geometry and motion from imagery.

Concretely, each instance of data starts as an N frame stereo video consisting of left-right image pairs \mathbf{I}_i and \mathbf{I}'_i indexed by frame index $i \in [1, N]$. We convert these

stereo pairs to a dynamic 3D point cloud with K points in a world-space coordinate frame, where each point, indexed by $j \in [1, K]$, has a time-varying position \mathbf{p}_i^j . As part of the process of generating this dynamic point cloud, we also extract a number of auxiliary quantities: (1) per-frame camera extrinsics, (the left camera’s position \mathbf{c}_i and orientation \mathbf{R}_i), (2) rig calibration for the stereo video giving the position \mathbf{c}_r and orientation \mathbf{R}_r of the right camera relative to the left camera, and (3) a per-frame disparity map \mathbf{D}_i .

3.1. Data Processing Pipeline

At a high level, our pipeline for converting a stereoscopic video into a dynamic point cloud involves estimating camera poses, stereo disparity, and 2D tracks, fusing these quantities into a consistent 3D coordinate frame, and performing several filtering operations to ensure temporal consistent, high-quality reconstructions (Fig. 2). In this section, we describe in detail the key components of this process.

SfM. We start by processing the sequence of stereo frames $\mathbf{I}_i \leftrightarrow \mathbf{I}'_i$ to produce camera pose estimates $(\mathbf{c}_i, \mathbf{R}_i)$. We first use a SLAM method to divide the video into shots, as in [116]. For each shot, we run an incremental SfM algorithm similar to COLMAP [76]. We initialize the stereo rig calibration $(\mathbf{c}_r, \mathbf{R}_r)$ to a rectified stereo pair with baseline 6.3cm, but optimize for the calibration in bundle adjustment. In practice, we found that the exact stereo pair orientation can vary significantly from its nominal configuration and that optimizing the rig was critical for good results.

Depth Estimation. We next estimate a per-frame disparity map, operating on each frame independently. In particular, we use the estimated camera rig calibration $\mathbf{c}_r, \mathbf{R}_r$ to create rectified stereo pairs from the stereo fisheye video and estimate the per-frame disparity \mathbf{D}_i with RAFT [83, 84, 90].

3D Track Estimation and Optimization. We extract long-range 2D point trajectories using BootsTAP [17]. Using the camera poses $\mathbf{c}_i, \mathbf{R}_i$ and disparity maps \mathbf{D}_i , we unproject these tracked points into 3D space, turning each 2D track j into a 3D motion trajectory $\mathbf{p}_1^j, \dots, \mathbf{p}_N^j$ across all frames. In general, each point will usually only be tracked in a subset of frames, but for simplicity, we describe the formulation as if all points are always visible. Moreover, since subsequent steps are done independently per-track, we drop the superscript j in future references.

Since stereo depth estimation is performed per-frame, the initial disparity estimates (and therefore, the 3D track positions) are likely to exhibit high-frequency temporal jitter. To compensate for potentially inconsistent disparity estimates, we formulate an optimization strategy that solves for a per-frame scalar offset $\delta_i \in \mathbb{R}$ that moves each point \mathbf{p}_i along the ray from camera location \mathbf{c}_i to \mathbf{p}_i at frame i . This ray is denoted $\mathbf{r}_i = (\mathbf{p}_i - \mathbf{c}_i)/\|\mathbf{p}_i - \mathbf{c}_i\|$, and we refer to the updated location as $\mathbf{p}'_i = \mathbf{p}_i + \delta_i \mathbf{r}_i$.

To ensure static points remain stationary while moving tracks maintain realistic, smooth motion, avoiding abrupt depth changes frame by frame, we design an optimization objective comprising three terms: a static loss $\mathcal{L}_{\text{static}}$, a dynamic loss $\mathcal{L}_{\text{dynamic}}$, and a regularization loss \mathcal{L}_{reg} . The static loss $\mathcal{L}_{\text{static}}$ minimizes jitter by encouraging points to remain close to each other in world space:

$$\mathcal{L}_{\text{static}} = \sum_{i=1}^N \sum_{j=1}^N \frac{\|\mathbf{p}'_i - \mathbf{p}'_j\|^2}{N_p'^2} \quad (1)$$

where $N_p' = \sum_{i=1}^N \|\mathbf{p}'_i\|/N$ is a normalizing factor. The dynamic loss term reduces jitter by minimizing acceleration along the camera ray through a discrete Laplacian operator:

$$\mathcal{L}_{\text{dynamic}} = \sum_{i=1}^N \sum_{\Delta \in \mathcal{W}} \left[(\mathbf{p}'_{i+\Delta} - 2\mathbf{p}'_i + \mathbf{p}'_{i-\Delta})^\top \mathbf{r}_i \right]^2 \quad (2)$$

where the acceleration along the ray is calculated over multiple window sizes $\mathcal{W} = \{1, 3, 5\}$.

The two loss terms are weighted by a track-dependent function, $\sigma(m)$, where m is a measure of the motion magnitude of the track. Motion is measured in 2D rather than 3D because distant points can appear to have a larger 3D motion due to noise amplification at low disparities. Specifically, we project the 3D motion trajectory between time $i - w_o$ and the current time i into 2D image-space at time i , and calculate the track's motion magnitude m as the 90th percentile of the track's trail length across all frames. The track trail length for a frame is measured by projected 3D points along the track to the current frame as if the camera is *static* in a window of $w_o = 16$ frames,

$$m = \text{Percentile}_{i=1:N}^{90} \left[\max_{w=1:w_o} \|\pi_i(\mathbf{p}_i) - \pi_i(\mathbf{p}_{i-w})\| \right] \quad (3)$$

where $\pi_i(\cdot) \in \mathbb{R}^2$ gives the projected pixel location of a 3D point on camera \mathbf{c}_i 's image plane. The weighting function $\sigma(m)$ is defined as $\sigma(m) = \frac{1}{1+\exp(m-m_0)}$ where $m_0 = 20$. Finally, to encourage faithfulness to the originally estimated disparities, we regularize the displacements in disparity space:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \sum_{i=1}^T \left(\frac{1}{\delta_i + \|\mathbf{p}_i - \mathbf{c}_i\|} - \frac{1}{\|\mathbf{p}_i - \mathbf{c}_i\|} \right)^2, \quad (4)$$

where use of disparity space reflects the fact that the measurements themselves originate as disparities. Practically, the impact of the use of disparity is that larger deviations are tolerated at more distant points, where depth is intrinsically more uncertain.

The full objective function is

$$\min_{\{\delta_i\}_{i=1}^N} \sigma(m)\mathcal{L}_{\text{static}} + (1 - \sigma(m))\mathcal{L}_{\text{dynamic}} + \mathcal{L}_{\text{reg}}. \quad (5)$$

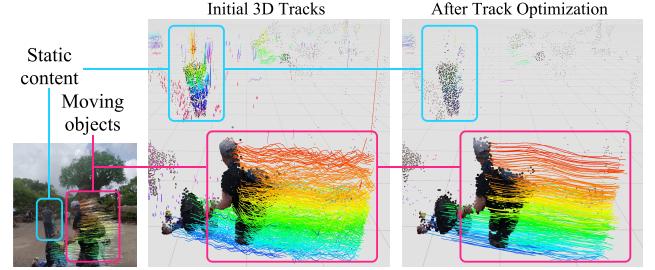


Figure 3. **Effect of track optimization.** Comparing motion trajectories before and after track optimization, we see that optimization resolves the high frequency jitter along camera rays, affecting both static and dynamic content. After optimization, static content has static tracks, and dynamic tracks are less noisy.

We set $\lambda_{\text{reg}} = 10^{-4}$ and optimize Eqn. 5 using Adam with a learning rate of 0.05 for 100 steps. The effect of track optimization is shown in Fig. 3. The optimized motion is smoother and does not contain high frequency noise.

Implementation details. *Shot-selection.* Rather than work with the full video, we break the footage into discrete, trackable shots using ORB-SLAM2’s stereo estimation mode [61] following [118]. *Field of View.* While estimating pose, we use a 140° FoV fisheye format, which we found to capture more of the (usually static) background and less of the (often dynamic) foreground, yielding more reliable camera poses. *Stereo Confidence Checks.* We discard pixels where the y -component of RAFT flow is more than 1 pixel (since rectified stereo pairs should have perfectly horizontal motion) and where the stereo cycle consistency error is more than 1 pixel (since such pixels are unreliable). *Dense 2D tracks.* To get dense tracks, we run BootsTAP with dense query points: for every 10th frame, we uniformly initialize 128×128 query points on frames of resolution 512×512 . We then prune redundant tracks that overlap on the same pixel. *Drifting tracks.* Since 2D tracks can drift on textureless regions, we discard moving 3D tracks that correspond to certain semantic categories (*e.g.*, “walls”, “building”, “road”, “earth”, “sidewalk”), detected by DeepLabv3 [13] on ADE20K classes [116, 117].

Filtering details. A fraction of the video clips that are processed may be unsuitable because they either (1) are not videos, and are entirely static images, (2) contain cross-fades, or (3) have text or other synthetic graphics. To discard text and title sequences, we avoid creating video clips from the start and ends of the source videos. We identify cross-fades by running SIFT [57] matching through the video at multiple temporal scales and discarding video clips with static camera but with fewer than 5 SIFT matches between frames that are 5 seconds apart.

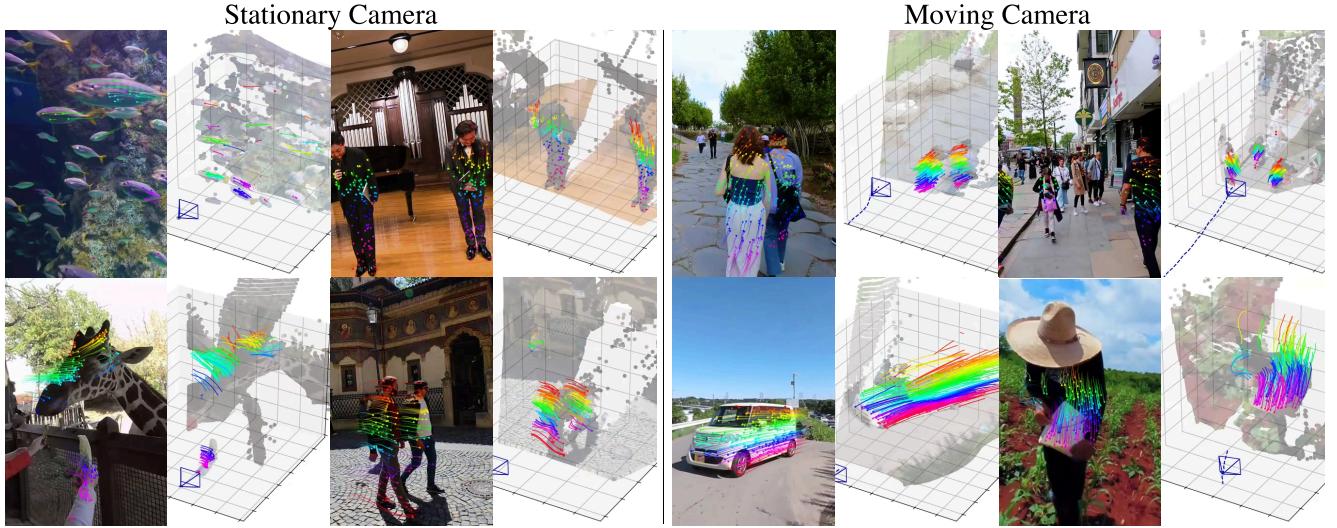


Figure 4. Diverse motion: Stereo4D captures a wide variety of types of moving objects, from swimming fish, to walking pedestrians, moving vehicles, and a farmer sowing seeds. It includes source videos captured with both stationary (left) and moving (right) cameras.



Figure 5. Diverse scene content: A word cloud of captioned frames from our dataset shows our data is diverse, including a variety of common objects seen in videos.

3.2. Stereo4D Dataset

Fig. 4 illustrates a subset of videos and reconstructions from a dataset processed with the above pipeline, encompassing more than 100K clips capturing everyday scenes and activities. To visualize the range of content, we used an automatic captioning system to generate captions for the dataset and created a word cloud (Fig. 5) highlighting the most frequently observed objects.

4. Learning a prior on how things move

We now describe our method for predicting dynamic 3D point clouds from pairs of images, and how we train it with our Stereo4D data. Our model is based on DUS3R [99], which predicts a 3D point cloud for a static scene from images. Given two input images, it uses a ViT-based architecture [18] to extract image features and uses a transformer-based decoder to cross-attend features from two images,

and then use a downstream *pointmap* decoder to output pointmaps for the two images, aligned in the first image's coordinate frame.

DynaDUST3R model. While DUST3R focuses on static scene structure, our proposed DynaDUST3R method, illustrated in Fig. 6, works with dynamic scenes by adding a *motion* head that predicts how the points move between two frames. As input, DynaDUST3R accepts two images: I_0 at time t_0 , and I_1 at time t_1 (where t_0 and t_1 may be seconds apart). It also accepts an intermediate query time $t_q \in [0, 1]$; the motion head is asked to predict 3D scene flow from the two input frames to query time t_q , as described below.

Like DUS₃R, DynaDUS₃R begins by encoding the images with a shared ViT and cross-attention decoder, producing global features G^0 and G^1 for \mathbf{I}_0 and \mathbf{I}_1 , respectively. Each feature embedding can be converted into geometry using DUS₃R’s point head: e.g., for image \mathbf{I}_0 , the point head produces a pointmap $\mathbf{P}^0 \in \mathbb{R}^{H \times W \times 3}$ representing the geometry at time t_0 , as well as a point confidence map $\mathbf{C}_{\text{point}}^0 \in \mathbb{R}^{H \times W}$. Each point cloud is predicted in the coordinate frame of \mathbf{I}_0 , but *at the time of its respective image* (so, the two point clouds may differ due to scene motion).

We add a separate *motion* head in parallel to the original point head, to predict a map of 3D displacement vectors (that is, a scene flow map, which we refer to as a *motion map*) for each pointmap. The motion map should displace each input frame to an intermediate time $t_q \in [0, 1]$, where $t_q = 0$ corresponds to \mathbf{I}_0 , the time of \mathbf{I}_0 , and similarly for $t_q = 1, t_1$, and \mathbf{I}_1 . The motivation for predicting motion to an intermediate time (inclusive of the endpoints) is twofold: first, it leads to a more general prediction task where we can predict a full motion trajectory between two frames, and second, it allows us to use partial ground truth 3D trajectory

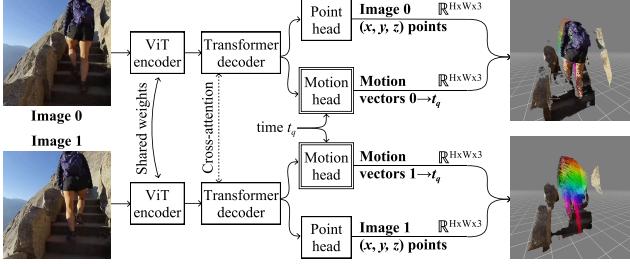


Figure 6. **DynaDUST3R architecture.** Given two images ($\mathbf{I}_0, \mathbf{I}_1$) of a dynamic scene and a desired target time t_q , the images are passed through a ViT encoder and transformer decoder. The resulting features are processed by (1) a pointmap head that predicts 3D points in the coordinate frame of \mathbf{I}_0 , and (2) a 3D motion head that predicts the motion of all points to the target time t_q . A double outline indicates a new component compared to DUST3R.

ries as supervision; not all trajectories may span all the way from t_0 to t_1 , but may span through some intermediate time.

For each image \mathbf{I}_v (with $v \in \{0, 1\}$), the network outputs a 3D motion map $\mathbf{M}^{v \rightarrow t_q}$ for the corresponding pointmap from t_v to t_q with corresponding motion confidence map $\mathbf{C}_{\text{mot}}^v \in \mathbb{R}^{H \times W}$. This prediction is based on the global feature G^v as well as an embedding of the query time $\text{emb}(t_q)$. We use positional embedding [95] to encode time t_q to a 128-D vector and inject it to the motion features in the motion head via linear projection layers.

Training objective. We use the same confidence-aware scale-invariant 3D regression loss as in DUST3R. We first normalize both the predicted and ground truth pointmaps using scale factors $z = \text{norm}(\mathbf{P}^0, \mathbf{P}^1)$ and $\bar{z} = \text{norm}(\bar{\mathbf{P}}^0, \bar{\mathbf{P}}^1)$, respectively (where a bar, e.g., $\bar{\mathbf{P}}^0$, denotes a ground truth quantity, and where ‘norm’ computes the average distance between a set of points and the world origin). We scale the motion maps with the same scales z and \bar{z} . Following DUST3R, we compute a Euclidean distance loss on the pointmap, setting $\mathcal{L}_{\text{point}}$ to

$$\sum_{v \in \{0, 1\}} \sum_{i \in \mathcal{D}^v} \mathbf{C}_{\text{point}, i}^v \left\| \frac{1}{z} \mathbf{P}_i^v - \frac{1}{\bar{z}} \bar{\mathbf{P}}_i^v \right\| - \alpha_p \log \mathbf{C}_{\text{point}, i}^v \quad (6)$$

where \mathcal{D}^v corresponds to the valid pixels where ground truth is defined and α_p is a weighting hyperparameter. We additionally compute a Euclidean distance loss on the position *after motion*, which encourages the network to learn correct displacements. This loss $\mathcal{L}_{\text{motion}}$ is defined as

$$\sum_{v \in \{0, 1\}} \sum_{i \in \mathcal{D}^v} \mathbf{C}_{\text{mot}, i}^v \left\| \frac{1}{z} \mathbf{P}_i^{v \rightarrow t_q} - \frac{1}{\bar{z}} \bar{\mathbf{P}}_i^{v \rightarrow t_q} \right\| - \alpha_m \log \mathbf{C}_{\text{mot}, i}^v, \quad (7)$$

where $\mathbf{P}_i^{v \rightarrow t_q} = \mathbf{P}_i^v + \mathbf{M}_i^{v \rightarrow t_q}$.

Training details. We initialize our network with DUST3R weights and initialize the motion head with the same weights as the point head. We finetune for 49k iterations, with batch size 64, learning rate 2.5e-5, optimized by Adam

Method	Stereo4D			ADT		
	EPE _{3D} ↓	$\delta_{3D}^{0.05}\uparrow$	$\delta_{3D}^{0.10}\uparrow$	EPE _{3D} ↓	$\delta_{3D}^{0.05}\uparrow$	$\delta_{3D}^{0.10}\uparrow$
DynaDUST3R (PointOdyssey)	0.6191	11.61	20.25	0.3126	8.56	18.03
DynaDUST3R (Stereo4D)	0.1110	65.07	75.18	0.1231	51.98	65.20

Table 1. **Synthetic vs. Real Training Data.** Compared to synthetic data (PointOdyssey [115]), training on Stereo4D improves DynaDUST3R’s ability to generalize to real-world motion.

with weight decay 0.95. During training, we randomly sample pairs of video frames that are at most 60 frames apart. The weight for the confidence loss in Eqn 6-7 is $\alpha_m = \alpha_p = 0.2$. The model is trained on tracks extracted from both 60° FoV videos for (higher quality) and 120° FoV videos for (larger coverage).

5. Experiments

We conduct a series of experiments to validate the effectiveness of our proposed data and techniques. First, we evaluate our proposed real-world Stereo4D data mined from VR180 videos on the DynaDUST3R task. In particular, we compare models that are individually trained with our real-world data and with synthetic data, and we show that our data enables model learning more accurate 3D motion priors (Sec. 5.1). Second, we show that our trained model that adapts DUST3R has strong generalization to in-the-wild images of dynamic scenes, and enables accurate predictions of underlying geometry (Sec. 5.2).

5.1. 3D motion prediction

Baselines and metrics. To evaluate the efficacy of our data paradigm on motion prediction, we primarily compare DynaDUST3R trained on Stereo4D to the same network trained on a synthetic dataset, PointOdyssey [115]. PointOdyssey contains ground truth depth maps and 3D motion tracks rendered from an animation engine; we supervise the model with this data using the same hyperparameter settings as described above. During inference, given two video frames sampled from a video of a dynamic scene, we compare 3D end-point-error (EPE) between ground truth and predicted 3D motion vectors. We also compute the fraction of 3D points that have motion within 5 cm and 10 cm compared to ground truth ($\delta_{3D}^{0.05}, \delta_{3D}^{0.10}$), following [92, 98]. Since our model outputs point clouds up to an unknown scale, we align each prediction with the ground truth through a median scale estimate. We evaluate models trained on each of these two data sources on a held-out Stereo4D test-set, as well as on Arial Digital Twin (ADT) [64] data containing scene motion, processed by the TapVid3D benchmark [46]. As test data, we randomly sample pairs of frames that are at most 30 frames apart from both Stereo4D and ADT.

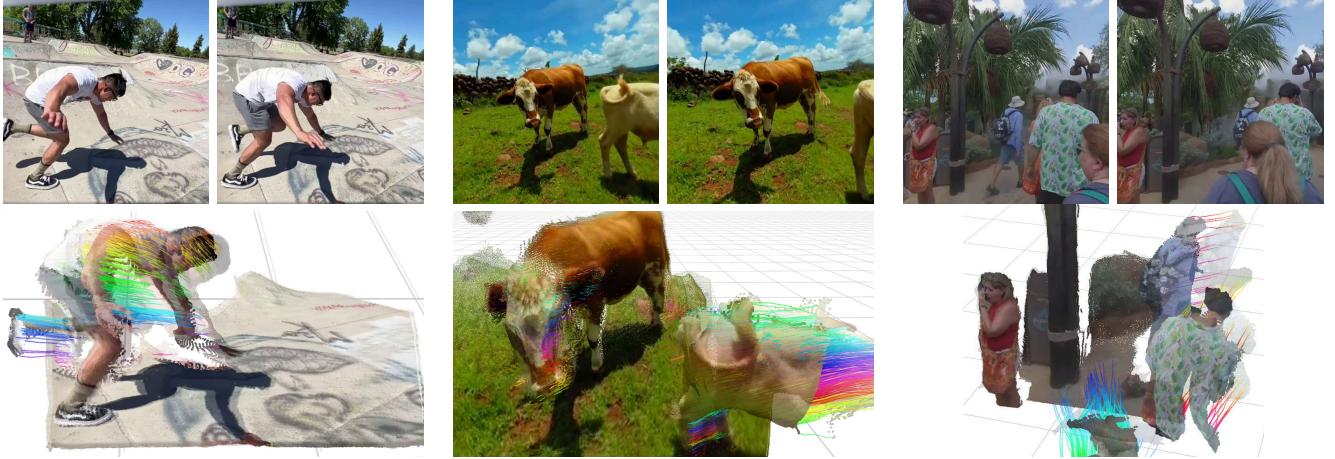


Figure 7. **Testing on held out examples from Stereo4D.** We visualize image pairs and corresponding dynamic 3D point clouds predicted by DynaDUS3R. It recovers accurate 3D shape and complex scene motion for objects such as people breakdancing and cows walking.



Figure 8. **Qualitative comparisons, 3D motion on the Stereo4D.** We compare variants of DynaDUS3R trained on different data sources. The PointOdyssey-trained model incorrectly predicts significant 3D motion on static elements such as the building wall and the banners near the streetlight, while the Stereo4D-trained model correctly predicts these elements as stationary. The Stereo4D model also makes more precise motion predictions for dynamic objects, such as humans with large movements (bottom row).

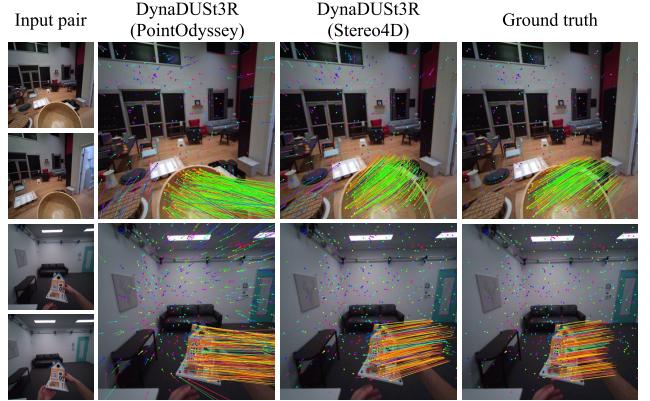


Figure 9. **Qualitative comparisons of predicted 3D motion on ADT [64].** DynaDUS3R trained on Stereo4D produces more accurate 3D motion compared to training on PointOdyssey.

Quantitative results. We show numerical results for two-frame 3D motion prediction in Tab. 1. DynaDUS3R trained on real-world data achieves better generalization and outperforms the baseline trained on PointOdyssey significantly across all metrics. This suggests the potential of our data for more effective learning of real-world 3D motion priors.

Qualitative results. Fig. 7 shows example results for three dynamic scenes in our Stereo4D test set, including visualizations of 3D point clouds and motion tracks. DynaDUS3R produces accurate 3D shape and motion tracks over the timespan defined by the two input images. Despite the inputs being two sparse images, our architecture enables querying intermediate motion states, resulting in continuous and potentially non-linear motion trajectories.

We also qualitatively compare predicted 3D motion

tracks between DynaDUS3R networks trained on Stereo4D and on PointOdyssey, by projecting their predicted 3D motion vectors into 2D image space. Fig. 8 and Fig. 9 show comparisons on the Stereo4D and ADT test set respectively. DynaDUS3R trained on Stereo4D produces more accurate 3D motion estimates for both static and moving objects. For instance, DynaDUS3R trained on PointOdyssey produces non-zero motion for the stationary street banner and erroneous motions for the walking people in Fig. 8.

5.2. Structure prediction

Baseline and metrics. We evaluate the quality of predicted 3D structure by comparing depth maps predicted by DUS3R [99], MonST3R [111], and DynaDUS3R trained on Stereo4D or PointOdyssey. DUS3R is designed to predict aligned point clouds from two input images of a static scene. MonST3R, a concurrent approach, extends DUS3R to handle dynamic scenes by predicting time-varying point

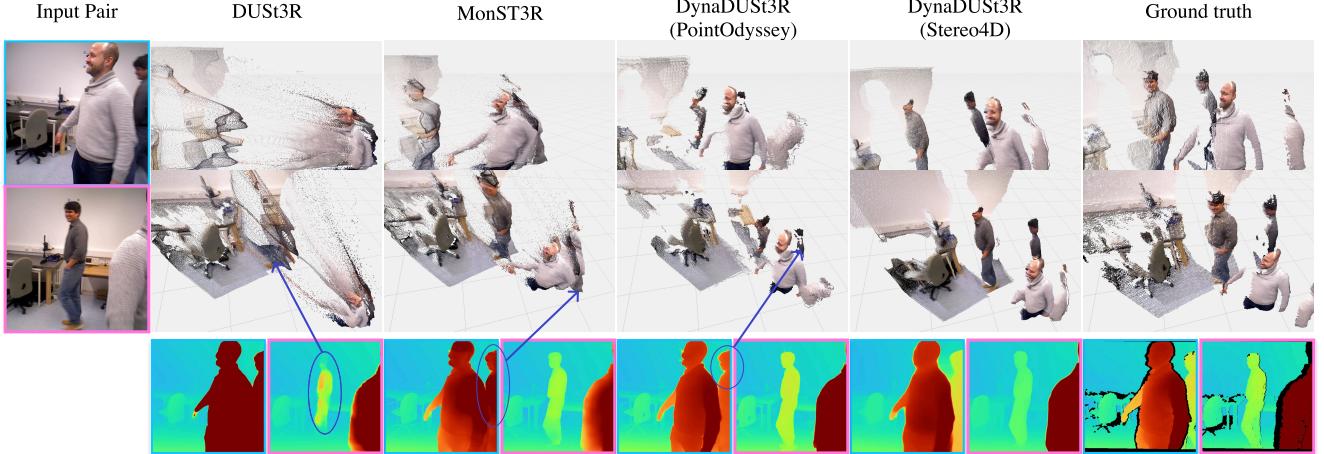


Figure 10. **Qualitative comparison, 3D structure on Bonn [63].** From left to right, we show an input image pair, predictions from different methods, and the ground truth geometry. The top two rows show 3D point clouds from two viewpoints, where we show the union of the pointmaps for the two input time steps. The bottom row shows the corresponding disparity for two input images. Compared to all the other methods, DynaDUST3R trained on Stereo4D achieves better 3D structure predictions with finer details.

clouds without modeling motion.

We evaluate predicted depth accuracy on the Bonn [63] dataset and our held-out test set, where we sample two views that are 30 frames apart from a video. Since we focus on the two-frame case, we do not apply any post-optimization to the network outputs. In addition, since all methods predict 3D point clouds in the coordinate frame of the first image, we include the two points clouds predicted from both input frames in our evaluation. We use standard depth metrics, including absolute relative error (Abs Rel) and percentage of inlier points $\delta < 1.25$, following prior work [100, 111]. We use the same median alignment as before to align the predicted depth map with the ground truth.

Quantitative comparisons. We show quantitative comparisons of depth predicted by different methods in Tab. 2. DynaDUST3R trained on Stereo4D outperforms all other baselines by a large margin. In particular, we demonstrate improved depth prediction on the unseen Bonn dataset.

Qualitative comparisons. We provide additional visual comparisons in Fig. 10, where we visualize ground truth 3D point clouds and predictions from our approach and the other three baselines at different input time steps. DuST3R predicts inaccurate depth relationships for the two moving people, while MonST3R and DynaDUST3R trained on PointOdyssey both predict distorted scene geometry. In contrast, our model trained on Stereo4D produces 3D structure that most closely resembles the ground truth.

6. Discussion and Conclusion

Limitations. Our data curation pipeline and trained model have limitations. The quality of the long-range 3D motion

Method	Stereo4D		Bonn [63]	
	Abs Rel \downarrow	$\delta < 1.25\uparrow$	Abs Rel \downarrow	$\delta < 1.25\uparrow$
DUST3R [99]	0.2696	67.77	0.1098	84.93
MonST3R [111]	0.1939	72.56	0.0721	92.60
Ours (PtOdyssey)	0.3858	61.87	0.0691	95.94
Ours (Stereo4D)	0.1032	87.93	0.0653	96.02

Table 2. **Quantitative comparison, depth maps.** DynaDUST3R trained on our Stereo4D data surpasses the performance of the model trained on PointOdyssey [115], as well as DUST3R and MonST3R under challenging sparse view settings.

tracks depends on the accuracy of optical flow and 2D point tracking and may degrade for distant background regions or objects occluded for long periods. Additionally, DynaDUST3R is a non-generative model that only operates on two-frame inputs. Extending our model to video input by adopting an extra global optimization [111] or integrating generative priors for modeling ambiguous motion content is a promising future direction.

Conclusion. We presented a pipeline for mining high-quality 4D data from Internet stereoscopic videos. Our framework automatically annotates each real-world video sequence with camera parameters, 3D point clouds, and long-range 3D motion trajectories by consolidating different noisy structure and motion estimates derived from videos. Furthermore, we show that training a variant of DUST3R on our real-world 4D data enables more accurate learning of 3D structure and motion in dynamic scenes, outperforming other baselines.

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. [1](#)
- [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*, 2011. [2](#)
- [3] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *IJCV*, 1999. [2](#)
- [4] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. [2](#)
- [5] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. [1](#)
- [6] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgbd reconstruction with semi-supervised data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. [2](#)
- [7] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. [2](#)
- [8] 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 *Eur. Conf. Comput. Vis.*, 2024. [2](#)
- [9] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. [2](#)
- [10] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Eur. Conf. Comput. Vis.*, 2008. [2](#)
- [11] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Transactions on Robotics*, 2021. [2](#)
- [12] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. [2](#)
- [13] Liang-Chieh Chen. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. [4](#)
- [14] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996. [2](#)
- [15] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007. [2](#)
- [16] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Rebecasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *NeurIPS*, 2022. [2](#)
- [17] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped training for tracking any point. *ICCV*, 2024. [3, 2](#)
- [18] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. [5](#)
- [19] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Int. Conf. Comput. Vis.*, 2015. [2](#)
- [20] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017. [2](#)
- [21] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. [2](#)
- [22] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009. [2](#)
- [23] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2010.
- [24] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Int. Conf. Comput. Vis.*, 2015. [2](#)
- [25] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *NeurIPS*, 2022. [2](#)
- [26] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *NeurIPS*, 2024. [2](#)
- [27] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. [2](#)
- [28] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jin Xu Zhang, Angela Castillo, Changan Chen, Xin Zhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo

- Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumini Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C.V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. [2](#)
- [29] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanaprasgam, 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. [2](#)
- [30] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *Eur. Conf. Comput. Vis.*, 2022. [2](#)
- [31] Heiko Hirschmüller, Peter R Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 2002. [2](#)
- [32] Aleksander Holynski, David Geraghty, Jan-Michael Frahm, Chris Sweeney, and Richard Szeliski. Reducing drift in structure from motion using extended features. In *3DV*, 2020. [2](#)
- [33] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH*, 1992. [2](#)
- [34] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024. [2](#)
- [35] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 2023. [2](#)
- [36] Michal Jancosek and Tomas Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011. [2](#)
- [37] Junpeng Jing, Ye Mao, and Krystian Mikolajczyk. Match-stereo-videos: Bidirectional alignment for consistent dynamic stereo matching. In *ECCV*, 2025. [2](#)
- [38] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Int. Conf. Comput. Vis.*, 2015. [2](#)
- [39] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. [2](#)
- [40] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. [2](#)
- [41] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. [2](#)
- [42] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Int. Conf. Comput. Vis.*, 2017. [2](#)
- [43] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 2023. [2](#)
- [44] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006. [2](#)
- [45] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. [2](#)
- [46] Skanda Koppula, Ignacio Rocco, Yi Yang, Joe Heyward, João Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. Tapvid-3d: A benchmark for tracking any point in 3d. In *NeurIPS*, 2024. [2](#) [6](#)
- [47] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. [2](#)
- [48] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. [2](#)
- [49] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. [1](#) [2](#)
- [50] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. [2](#)
- [51] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. [2](#)
- [52] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. [2](#)

- [53] Zhaoshuo Li, Wei Ye, Dilin Wang, Francis X Creighton, Russell H Taylor, Ganesh Venkatesh, and Mathias Unberath. Temporally consistent online depth estimation in dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. [2](#)
- [54] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *arXiv preprint arXiv:2412.04463*, 2024. [2](#)
- [55] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. [2](#)
- [56] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. [2](#)
- [57] G Lowe. Sift-the scale invariant feature transform. *Int. J. Computer Vision*, 2004. [4](#)
- [58] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 2020. [2](#)
- [59] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. [2](#)
- [60] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 2015. [2](#)
- [61] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. on Robotics*, 2015. [4, 2](#)
- [62] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. [2](#)
- [63] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *IROS*, 2019. [8](#)
- [64] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *ICCV*, 2023. [6, 7](#)
- [65] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proc. CVPR Workshops*, 2017. [2](#)
- [66] Hyun Soo Park, Takaaki Shiratori, Iain Matthews, and Yaser Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *Eur. Conf. Comput. Vis.*, 2010. [2](#)
- [67] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. [2](#)
- [68] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. [2](#)
- [69] Keunhong Park, Philipp Henzler, Ben Mildenhall, Jonathan T Barron, and Ricardo Martin-Brualla. Camp: Camera preconditioning for neural radiance fields. *ACM Transactions on Graphics (TOG)*, 2023. [2](#)
- [70] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. [2](#)
- [71] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *IJCV*, 2004. [2](#)
- [72] Marc Pollefeys, David Nistér, J-M Frahm, Amir Abarzadeh, Philippas Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, Paul Merrell, et al. Detailed real-time urban 3d reconstruction from video. *IJCV*, 2008. [2](#)
- [73] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. [1](#)
- [74] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. [1, 2](#)
- [75] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. [1, 2](#)
- [76] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. [2, 3](#)
- [77] Johannes L Schonberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. [2](#)
- [78] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. *arXiv preprint arXiv:2406.01493*, 2024. [2](#)
- [79] Shihao Shen, Yilin Cai, Wenshan Wang, and Sebastian Scherer. Dytavno: Joint refinement of visual odometry and motion segmentation in dynamic environments. In *ICRA*, 2023. [2](#)
- [80] Meng-Li Shih, Wei-Chiu Ma, Lorenzo Boyice, Aleksander Holynski, Forrester Cole, Brian Curless, and Janne Kontkanen. Extranerf: Visibility-aware view extrapolation of neural radiance fields with diffusion models. In *CVPR*, 2024. [2](#)
- [81] Tomas Simon, Jack Valmadre, Iain Matthews, and Yaser Sheikh. Kronecker-markov prior for dynamic 3d reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016. [2](#)

- [82] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, 2006. 2
- [83] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 2, 3
- [84] Deqing Sun, Charles Herrmann, Fitzsum Reda, Michael Rubinstein, David J Fleet, and William T Freeman. Disentangling architecture and training for optical flow. In *ECCV*, 2022. 3, 2
- [85] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003. 2
- [86] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2
- [87] Chris Sweeney, Aleksander Holynski, Brian Curless, and Steve M Seitz. Structure from motion for panorama-style videos. *arXiv preprint arXiv:1906.03539*, 2019. 2
- [88] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2
- [89] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1
- [90] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 3, 2
- [91] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *NeurIPS*, 2021. 2
- [92] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 6
- [93] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *NeurIPS*, 2024. 2
- [94] Geert Van Meerbergen, Maarten Vergauwen, Marc Pollefeys, and Luc Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *Int. J. Comput. Vis.*, 2002. 2
- [95] A Vaswani. Attention is all you need. *NeurIPS*, 2017. 6
- [96] Minh Vo, Srinivasa G Narasimhan, and Yaser Sheikh. Spatiotemporal bundle adjustment for dynamic 3d reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2
- [97] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes, 2023. 2
- [98] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 2, 6
- [99] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 1, 2, 5, 7, 8
- [100] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Int. Conf. Comput. Vis.*, 2023. 2, 8
- [101] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *ECCV*, 2025. 2
- [102] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. Nerfiller: Completing scenes via generative 3d inpainting. In *CVPR*, 2024. 2
- [103] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. *arXiv preprint arXiv:2411.18613*, 2024. 2
- [104] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 1, 2
- [105] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 1, 2
- [106] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Eur. Conf. Comput. Vis.*, 2018. 2
- [107] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2
- [108] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 2
- [109] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 2
- [110] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2
- [111] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 2, 7, 8
- [112] Youmin Zhang, Matteo Poggi, and Stefano Mattoccia. Temporalstereo: Efficient spatial-temporal stereo matching network. In *IROS*, 2023. 2
- [113] Zhoutong Zhang, Forrester Cole, Richard Tucker, William T Freeman, and Tali Dekel. Consistent depth of moving objects in video. *ACM Transactions on Graphics (ToG)*, 2021. 2
- [114] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure

- and motion from casual videos. In *Eur. Conf. Comput. Vis.*, 2022. [2](#)
- [115] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Int. Conf. Comput. Vis.*, 2023. [2](#), [6](#), [8](#)
- [116] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. [3](#), [4](#)
- [117] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019. [4](#)
- [118] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. [4](#), [2](#)

Stereo4D: Learning How Things Move in 3D from Internet Stereo Videos

Supplementary Material

7. Stereo4D Statistics

We collected around 110k clips from 6,493 Internet VR180 videos. Fig. 11 shows the camera translation distribution between the first and last frame of each clip. In Fig. 12, we measure the motion in terms of pixel displacement projected onto the image frame. Measuring motion in pixel-space emphasizes motion that occurs closer to the camera, since such motion yields larger pixel displacements, while naturally de-emphasizing motion further from the camera.

8. More qualitative comparisons

8.1. More results on held-out Stereo4D examples

Fig. 13 shows additional DynaDUS3R predictions on the Stereo4D held-out test set, extending Fig. 7 from the main paper. Fig. 14 shows additional qualitative examples of motion comparisons on Stereo4D test set, extending Fig. 8 from the main paper. Fig. 14 compares variants of DynaDUS3R trained on different data sources. The model trained on PointOdyssey incorrectly predicts large 3D motions, while the model trained on Stereo4D makes more accurate motion predictions, closer to ground truth.

8.2. More qualitative examples on track optimization

In Fig. 16, we illustrate estimated tracks for a video sequence featuring a forward-moving camera and vehicles driving towards the camera. Our initial 3D tracks derived directly from RAFT depth, BootsTAP 2D tracks, and SfM camera pose, show significant jitter for both dynamic (vehicle) and static (ground) points. However, after applying our track optimization, the ground points produce stable, static tracks, and vehicle tracks become smooth and coherent.

9. Dataset curation details

9.1. Equirectangular videos

The raw videos that we collect (see examples in Fig. 15) are natively stored in a cropped equirectangular format, which differs from a full 360° equirectangular projection as the horizontal field of view of the cropped format typically spans 180° —half of a full sphere. These videos often contain metadata specifying the horizontal and vertical field of view. For instance, metadata for a typical video might specify $\text{start}_{\text{yaw}} = -90.0^\circ$, $\text{end}_{\text{yaw}} = 90.0^\circ$, $\text{start}_{\text{tilt}} = -90.0^\circ$, $\text{end}_{\text{tilt}} = 90.0^\circ$; Since many VR180 videos are designed for an immersive VR experience, they are typically viewed with headsets. Hence, the baseline between the left and right

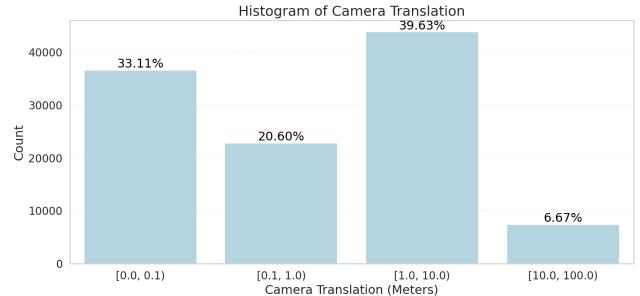


Figure 11. Camera statistics from Stereo4D. We measure the difference (in meters) of camera poses between the start and end frame of each video clip as calculated by SfM.

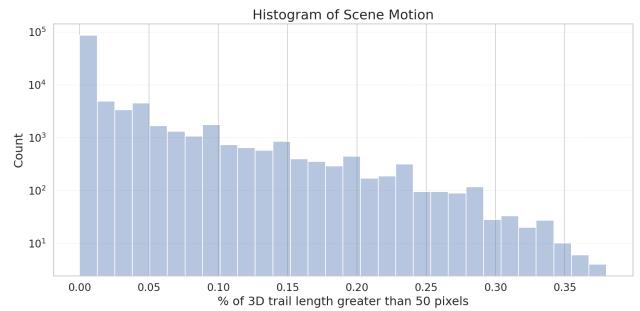


Figure 12. Scene motion statistics from Stereo4D. We measure the motion in terms of pixel displacement projected onto the image frame. For each video, we measure the percentage of tracks that have 3D trail length greater than 50 pixels. The 3D trail length is measured by Eqn. 3.

cameras typically closely matches the average human eye distance of 6.3 cm.

9.2. SfM

For ease of processing with standard 3D computer vision pipelines, and to benefit from the wide FoV of the input videos, we convert the videos from their native format (equirectangular projections) to a fisheye format for camera pose estimation. We use a 140° field of view for these fisheye-projected videos, because many equirectangular videos have a black fade-out/feathering/vignetting effect applied at the boundary, as shown in Fig. 15. We found that using wider FoV frames significantly improves camera pose estimation in dynamic scenes. When using narrow FoV projections, dynamic objects are more likely to occupy a large fraction of the frame; when these dynamic foreground objects are rich in features, they can confuse camera tracking algorithms, leading to inaccurate camera poses that track the dynamic object rather than producing



Figure 13. **More qualitative results on Stereo4D test set.** Extending Fig. 7, we visualize image pairs and corresponding dynamic 3D point clouds predicted by DynaDUS3R trained on Stereo4D. Our method recovers accurate 3D shape and complex scene motion.

true camera motion with respect to the environment. In contrast, wide-angle fisheye videos capture more background regions, which tend to have stable features for tracking, yielding more reliable camera poses.

We first use ORB-SLAM2’s stereo estimation mode [61] to identify trackable sequences within the videos, utilizing the method devised by Zhou *et al.* to divide videos into discrete, trackable shots [118]. For each given shot, consisting of frames (I_1, \dots, I_n), we estimate camera poses and rig calibration via an incremental global bundle adjustment algorithm similar to COLMAP [76]. We initialize the stereo rig calibration to be that of a rectified stereo pair with baseline 6.3 cm, but optimize for the calibration as part of the bundle adjustment process, as in practice the stereo rig can vary significantly from its nominal configuration. This process yields a camera position \mathbf{c}_i and orientation \mathbf{R}_i for each frame i (defined as the pose of the left camera), and a position \mathbf{c}_r and orientation \mathbf{R}_r for the right camera relative to the left (assumed to be constant throughout the shot).

9.3. Depth estimation

Depth estimation is first performed on a per-frame basis, with disparity maps computed independently for each frame.

We use the estimated camera rig calibration $\mathbf{c}_r, \mathbf{R}_r$ to rectify the original high resolution equirectangular video frames, ensuring that (1) the left and right views have centered principal points, (2) are oriented perpendicular to the baseline, and (3) pointing in a parallel direction. We then convert the equirectangular videos to perspective projections for downstream predictions.

Disparity is estimated from optical flow [84, 90] between the rectified left and right frames. The x -component of the optical flow is used as disparity, which is converted to met-

ric depth using:

$$\text{Depth} = \frac{\text{baseline} \times f}{\text{disparity}}. \quad (8)$$

Here baseline = 0.063m, and f is the frame’s focal length.

Outlier Rejection. Several criteria are applied to filter out unreliable pixels: *Inconsistency between left and right eyes*: Disparity is rejected if the optical flow fails a cycle-consistency check with an error exceeding one pixel. *Depth values exceeding 20 meters* are considered invalid. Estimating accurate depth beyond a certain range requires sub-pixel disparity estimation, and therefore the resulting depths are usually very noisy. *Negative flow values* that shouldn’t occur, but can, often due to errors in textureless regions. *Large vertical flow*: pixels with a y -component of flow exceeding one pixel are removed (as in our rectified stereo pairs correspondences should have the same y -value, and violating that epipolar constraint indicates uncertain matches). *Occlusion boundaries*: Depth gradients exceeding a threshold (threshold = 0.3) indicate occlusion boundaries and are rejected. For a pixel location (x, y) , depth gradients are computed as:

$$\text{grad}_x = |\text{Depth}(x+1, y) - \text{Depth}(x-1, y)|,$$

$$\text{grad}_y = |\text{Depth}(x, y+1) - \text{Depth}(x, y-1)|.$$

Pixels are rejected if $\text{grad}_x > \text{threshold} \times \text{Depth}(x, y)$ or $\text{grad}_y > \text{threshold} \times \text{Depth}(x, y)$.

9.4. 2D tracks

We extract long-range 2D point trajectories using BootS-TAP [17]. We run tracking on the left-eye video only. For every 10 frames, we uniformly initialize query points on image with stride 4. We then remove duplicated queries if earlier tracks fall within 1 pixel of a query point.

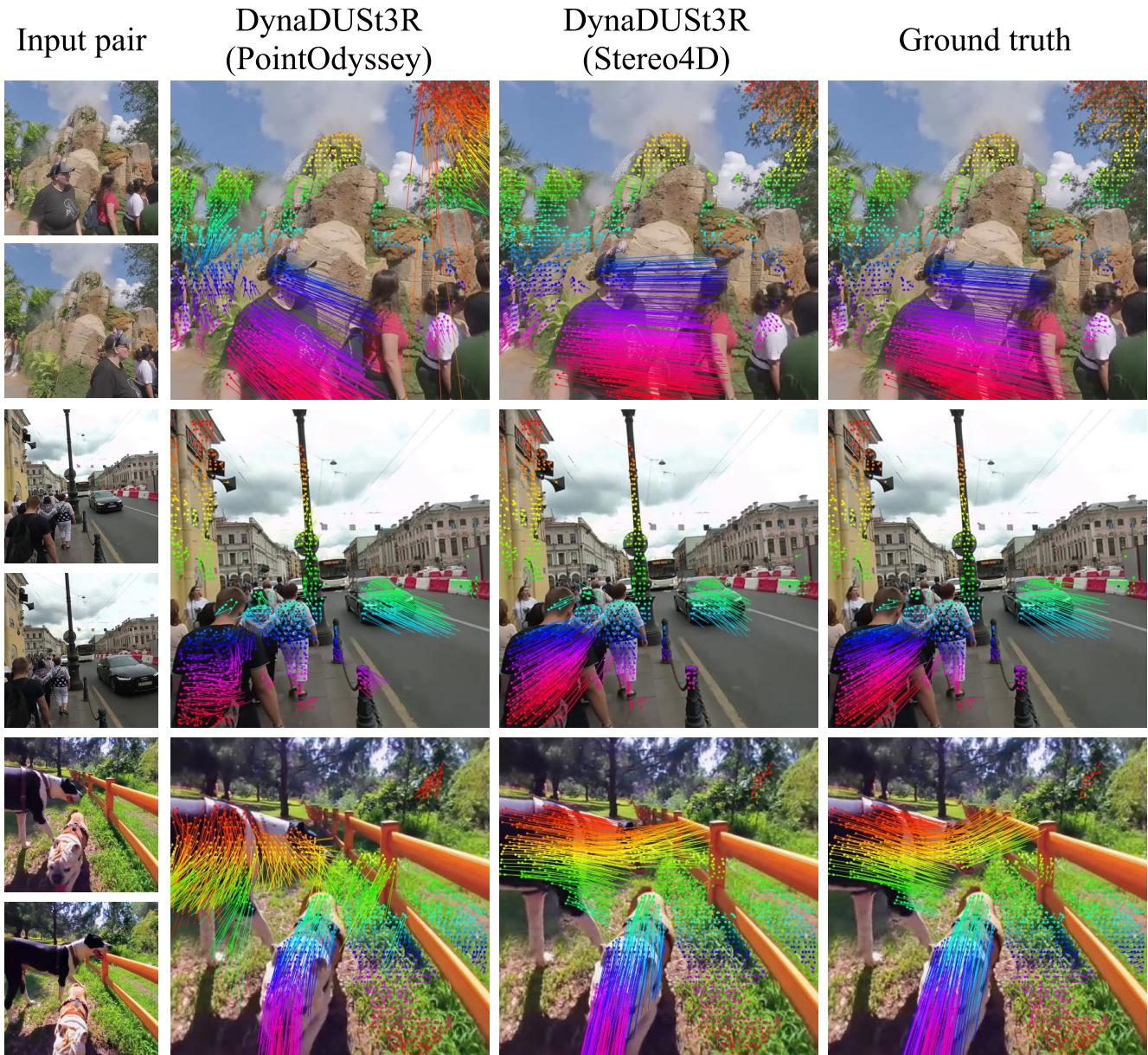


Figure 14. **More qualitative comparisons of 3D motion in the Stereo4D test set.** Extending Fig. 8, we compare variants of DynaDUST3R trained on different data sources. The Stereo4D-trained model also makes more precise motion predictions than the PointOdyssey-trained model.



Figure 15. Example equirectangular stereo videos collected from the internet.

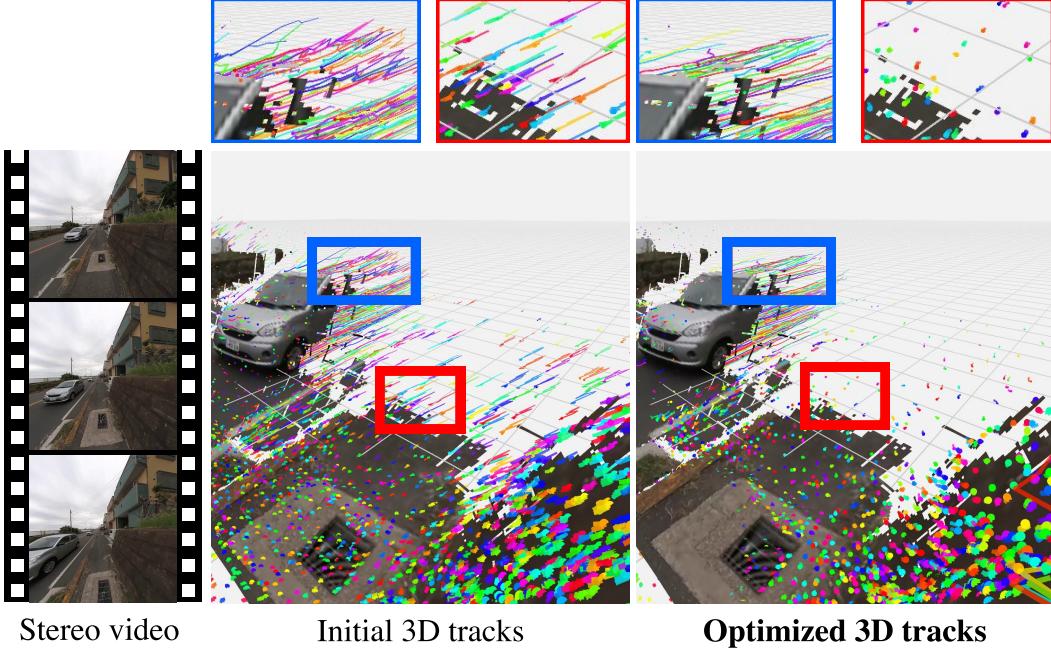


Figure 16. Effect of Track Optimization. We compare 3D tracks on a challenging walking tour video sequence. In this clip (left), the camera moves forward while vehicles drive toward the camera. We visualize the results across 16 frames, showing 3D trails left by both dynamic and static points. **Middle:** Our initial 3D tracks, created directly from RAFT, BootsTAP and SfM camera pose, also exhibit significant jitter for both dynamic (vehicle) and static (ground) points. **Right:** After applying our track optimization, the ground points yield stable, static tracks, and vehicle tracks become smooth and coherent.

9.5. Choice of FoV and resolution for perspective projection.

When converting the equirectangular videos to perspective projections, we use two FoVs: 60° and 120° . Both perspective videos are set to a resolution of 512×512 , the maximum supported by BootsTAP. The 60° projection offers a higher sampling rate in scene units, which improves the accuracy of depth estimation and 2D tracks when measured in meters. Additionally, it has smaller perspective distortion near the image boundaries. In contrast, the 120° projection provides wider coverage, ensuring longer 2D tracks across the videos. This trade-off allows us to balance data quality with spatial coverage for downstream tasks, e.g. DynaDUST3R. We take the union of the 3D tracks derived from each of these videos for DynaDUST3R training supervision.

10. DynaDUST3R training details.

Dataloader. During training, we randomly sample two frames from the training videos that are at most 60 frames apart, at times t_0 and t_1 , ($t_0 < t_1$). Additionally, we also sample one auxiliary frame in between, at time t_{aux} , $t_0 < t_{\text{aux}} < t_1$, for additional track supervision between the two input frames. During training, we add data augmentation by applying random crops and color jitter to the input images

and cropping the ground truth pointmap and motionmap accordingly.

Training. The network takes input the two RGB images as well as query times $t_q = \{0, 1, \frac{t_{\text{aux}} - t_0}{t_1 - t_0}\}$ and predicts the pointmaps for the two input views and motionmaps for each query t_q . We supervise the network with losses defined in Eqn. 6 and 7. We initialize our network with the DUST3R weights and initialize the motion head with the same weights as the point head. We finetune for 49k iterations with batch size 64, learning rate 2.5×10^{-5} , and optimized by Adam with weight decay 0.95.