Video Depth Propagation

Luigi Piccinelli*[1] Thiemo Wandel*[1] Christos Sakaridis[1] Wim Abbeloos[2] Luc Van Gool[1,3]

[1]ETH Zürich [2]Toyota Motor Europe [3]INSAIT, Sofia University St. Kliment Ohridski

Abstract

Depth estimation in videos is essential for visual perception in real-world applications. However, existing methods either rely on simple frame-by-frame monocular models, leading to temporal inconsistencies and inaccuracies, or use computationally demanding temporal modeling, unsuitable for real-time applications. These limitations significantly restrict general applicability and performance in practical settings. To address this, we propose VeloDepth, *an efficient and robust online video depth estimation pipeline that effectively leverages spatiotemporal priors from previous depth predictions and performs deep feature propagation. Our method introduces a novel Propagation Module that refines and propagates depth features and predictions using flow-based warping coupled with learned residual corrections. In addition, our design structurally enforces temporal consistency, resulting in stable depth predictions across consecutive frames with improved efficiency. Comprehensive zero-shot evaluation on multiple benchmarks demonstrates the state-of-the-art temporal consistency and competitive accuracy of VeloDepth, alongside its significantly faster inference compared to existing video-based depth estimators. VeloDepth thus provides a practical, efficient, and accurate solution for real-time depth estimation suitable for diverse perception tasks. Code and models are available at github.com/lpiccinelli-eth/velodepth.*

1. Introduction

Depth estimation is a fundamental task in computer vision, which enables a dense perception of the geometric structure of the surrounding scene that is pivotal in a vast variety of applications ranging from autonomous systems [28, 46] and robotics [7, 59] to augmented reality [6] and medicine [20]. While the basic monocular setting of this task, *i.e.* monocular depth estimation (MDE) from single images, is inherently ill-posed due to scale ambiguity and offers fewer priors to learn, its simplicity has historically led to far more attention than depth estimation from videos, especially in the deep learning era [1, 8, 32, 33, 37, 52, 55, 56].

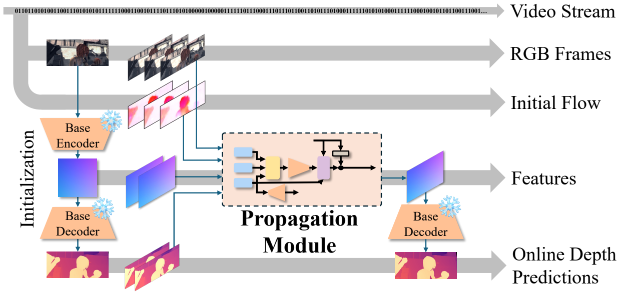

Figure 1. **VeloDepth** learns to leverage prior information contained in video data, such as previous predictions and scene dynamics. The Propagation Module refines the previous frame features, which "Base Decoder" decodes as current frame predictions. The module also propagates the features, along with the predictions, as the next frame's inputs. The prior information and the propagation lead to improved consistency and more efficient inference while maintaining the per-frame performance of the large "Base Model".

However, the video setting is better constrained since video sequences inherently provide strong priors, unlike single images, which can be leveraged to improve depth estimation. In particular, consecutive frames contain redundant visual information, allowing previous depth predictions and features to serve as informative cues for future frames. Even when estimated approximately, motion provides additional constraints on depth evolution over time. Leveraging temporal priors by propagating features and depth estimates across frames should lead to more accurate, consistent, and computationally efficient video-based depth methods. However, existing methods either ignore these priors, *i.e.* MDE, leading to temporal inconsistencies and flickering artifacts, or rely on computationally expensive solutions such as test-time optimization [15, 23], full temporal attention [3, 48], or video diffusion [12, 40], making them impractical for real-time applications. Moreover, these methods usually require future frames not only during training but also during inference, rendering them impractical for most real-world applications, which typically need to run online.

We respond to the above shortcomings in the literature by proposing VeloDepth, a video metric depth estimator that is

*Denotes equal contribution.

based on the propagation of depth-related information in a video across frames through time. The core principle of our approach is to exploit depth predictions and feature representations from previous frames, using them as informative priors to bootstrap the computation for subsequent frames. Specifically, our method employs a temporal propagation strategy where previously computed depth features and outputs are warped forward through fast but inaccurate optical flow and then refined via a learned residual correction, as depicted in Fig. 1. This design structurally enforces temporal consistency, as depth estimation at each frame inherently benefits from previously estimated features and outputs. Moreover, our approach enhances computational efficiency, since the propagation module only needs to learn the simpler residual mapping from propagated features, rather than performing a full RGB-to-depth prediction from scratch for every frame. Therefore, VeloDepth achieves comparable accuracy to computationally intensive single-image models applied frame-by-frame, while simultaneously largely increasing consistency and presenting the efficiency required for real-time applications.

We validate our approach with extensive experiments across four diverse benchmarks, demonstrating its robustness under different motion and scene conditions. Our results show that leveraging spatiotemporal priors leads to a better trade-off between temporal stability, computational efficiency, and overall depth accuracy compared to standard monocular depth models and prior video-based depth approaches.

2. Related Works

Monocular depth estimation was proposed in its end-to-end neural network formulation in [8]. However, monocular methods [8, 9, 17, 19, 31, 32, 39, 56] typically suffer from generalization issues due to limited data and the inherently ill-posed nature of monocular 2D-to-3D unprojection. Affine-invariant (relative) depth estimation mitigates this by predicting depth up to an unknown scale and shift, removing ill-posedness and improving cross-dataset performance [14, 37, 52]. However, relative depth estimation is unsuitable for physical, metric applications. More recent works strive for generalizable metric MDE incorporating camera information into the input [11, 55], internal features [33–35], or output space [1]. All MDE methods increase both data and compute to improve performance at the cost of real-time feasibility. Moreover, they are inherently trained in an image-based fashion, ignoring any temporal information and leading to inconsistencies across frames when run on videos.

Offline video depth estimation leverages all frames of the input video to enhance both temporal performance and accuracy over single-frame depth estimation. The paradigm defined by [15, 23] involves test-time optimization on initial depth estimates with either fixed or optimizable camera poses. [12, 40] have been the first to repurpose video diffusion models for video depth estimation, while Video Depth Anything [3] extends a large pre-trained affine MDE [53] by incorporating a spatiotemporal head that uses attention to correlate information across frames. However, these methods suffer from significant drawbacks, including high memory consumption and the inability to produce metric depth predictions. Moreover, their superior temporal consistency can be attributed to their offline nature, i.e. processing videos in chunks, where future frames are also included. On the other hand, VeloDepth does not require processing the entire video for each frame, which renders our method online, efficient, as well as capable of providing high consistency.

Online video depth estimation aims at online and possibly real-time, temporally consistent depth estimation. Early methods relied on recurrent architectures, such as LSTM [10], to retain temporal features [4, 24, 57], while others incorporated LiDAR for multi-modal fusion [30] or introduced stabilization networks to refine external depth [48, 50]. Most of the above methods are based on recurrent networks to retain past features but suffer from drift, vanishing gradients, and a poor capacity-efficiency tradeoff, which limits their effectiveness on real-time and long sequences. Stabilization-based methods [48, 50] refine depth estimates post-prediction but introduce additional computational overhead and fail to fully leverage past information. Yasarla *et al.* [54] proposed an optical flow-based attention memory, which ignores any features from previous predictions or flow, although requiring high-quality flow, and exploits memory-intensive attention. VeloDepth avoids these pitfalls by directly incorporating the previous frame's "neck features", depth predictions, and optical flow as priors, which, combined with a strong initialization, ensures consistency while maintaining computational efficiency.

3. Method

Video-based data naturally allow the use of prior estimates and the establishment of correspondences between consecutive frames. However, single-frame monocular depth estimation makes independent per-frame predictions, overlooking these temporal cues. The inherent temporal coherence in video sequences provides valuable prior information that can be exploited to enhance depth propagation, detailed in Sec. 3.1. VeloDepth leverages this temporal information by incorporating past depth predictions, deep feature propagation, and refined optical flow in a structured multi-modal framework as depicted in Fig. 2. The previous depth prediction acts as a geometric prior, ensuring consistency over time. In the absence of motion, the model should ideally learn an identity transformation from the previous to the current depth prediction, reinforcing temporal stability. Similarly, deep features from previous frames are propagated to provide additional prior information at the feature level.

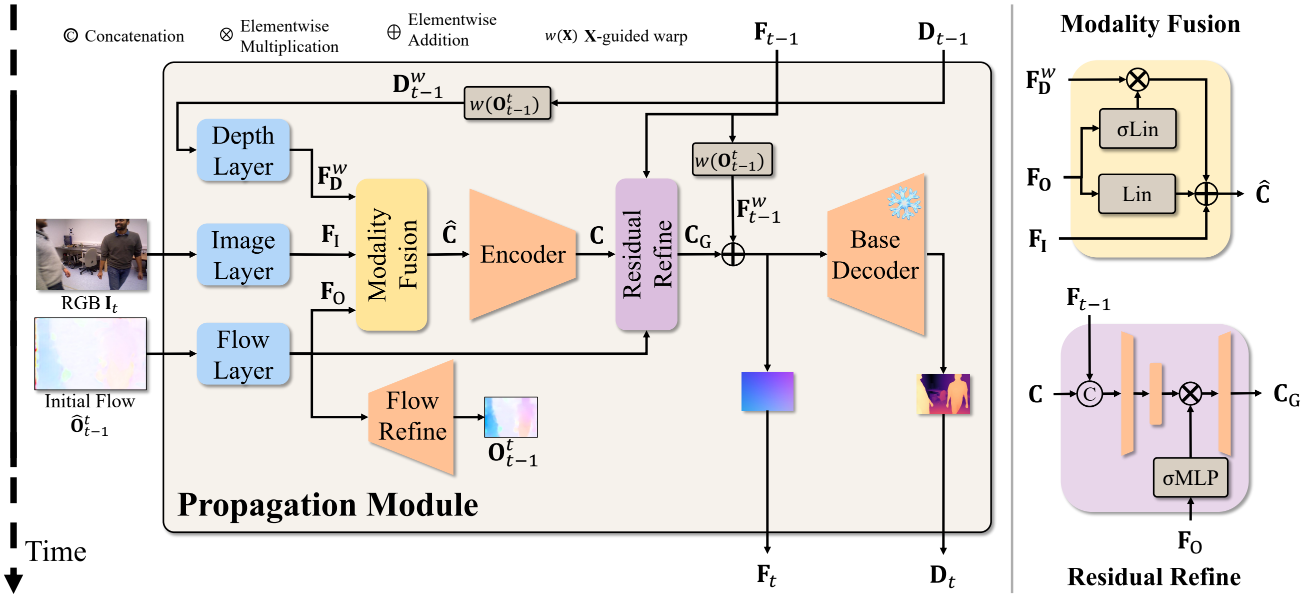

Figure 2. **Model architecture.** VeloDepth's Propagation Module is fed with the depth prediction from the previous frame ($\mathbf{D}_{t-1}$) and respective features ($\mathbf{F}_{t-1}$), the current frame ($\mathbf{I}_t$), and the backward optical flow between the current and previous frame ($\hat{\mathbf{O}}^t_{t-1}$). The goal of the Propagation Module is to produce a correction $\mathbf{C}_{\mathrm{G}}$ to the warped features of the previous frame, $\mathbf{F}^w_{t-1}$. The warping is performed with the refined optical flow $\mathbf{O}^t_{t-1}$. The initial correction tensor $\hat{\mathbf{C}}$ is the gated combination of features that are aware of current appearance, previous geometry, and scene dynamics, *i.e.* $\mathbf{F}_{\mathbf{I}}$, $\mathbf{F}^w_{\mathrm{D}}$, $\mathbf{F}_{\mathbf{O}}$. The correction tensor is processed with a lightweight encoder to produce the unrefined residuals $\mathbf{C}$, which in turn are made aware of the previous frame features $\mathbf{F}_{t-1}$ and gated based on the flow features $\mathbf{F}_{\mathbf{O}}$. For the sake of simplicity, we present the instances of $\mathbf{C}$, $\mathbf{C}_{\mathrm{G}}$, $\mathbf{F}_{t-1}$ and $\mathbf{F}^w_{t-1}$ for a single resolution, but these tensors and the block $\mathrm{ResidualRefine}$ have four instances, one for each resolution, without sharing weights.

Moreover, an additional warping-based 3D self-consistency loss is added to improve consistency over sequences in a bi-directional fashion, as described in Sec. 3.2.

3.1. Propagation Module

Our Propagation Module is inspired by the video encoding paradigm, where redundancies in video data are compressed by using motion vectors, *i.e.* rough optical flow, and residual coefficients, which capture the difference between the previous frame warped (PFW) and the current frame. However, unlike video encoding, a video depth estimator does not have access to the current frame output, as the latter is the final product of the entire model. Therefore, the Propagation Module learns a correction function that refines the PFW output in a way that the corrected version matches the actual current output. This correction is performed at the deep feature level rather than directly in the output depth space, in order to take advantage of a more expressive feature representation and the pre-trained Decoder.

Flow. Optical flow plays a crucial role in propagation, enabling warping of both previous-frame depth predictions and deep features. The model encodes flow between consecutive frames using two backbone layers, which take as input a three-channel image: red and green channels encode normalized flow values, while the blue channel captures the luma difference between frames, generating flow features $\mathbf{F}_{\mathbf{O}}$. The initial optical flow estimate $\hat{\mathbf{O}}^t_{t-1}$ is refined using a light-weight two-layer convolutional network, producing a correction term $\mathbf{H}$ that improves the flow accuracy, resulting in a corrected estimate $\mathbf{O}^t_{t-1} = \hat{\mathbf{O}}^t_{t-1} + \mathbf{H}$. This refinement step performs denoising and sharpening, particularly beneficial for low-resolution feature space warping. The initial flow estimate is computed using the CPU-based DIS algorithm [16], but motion vectors or high-quality predicted optical flow can alternatively be used; VeloDepth is agnostic and robust to this initial flow estimation.

Fusion and residuals. While feature warping improves propagation, it is prone to failure in occluded regions or when the optical flow is inaccurate. To address this, the fusion mechanism incorporates flow features $\mathbf{F}_{\mathbf{O}}$ to guide the selection of reliable propagated information. RGB and PFW depth features, denoted as $\mathbf{F}_{\mathbf{I}}$ and $\mathbf{F}^w_{\mathbf{D}}$, are first encoded with a backbone layer before being fused using a gated mechanism, which is used to prevent incorrect propagation in regions where $\mathbf{O}^t_{t-1}$ is unreliable. The fused features $\hat{\mathbf{C}}$ are obtained via:

$$\hat{\mathbf{C}} = \mathbf{F}_{\mathbf{I}} + \mathbf{G}_{\mathbf{D}} \odot \mathbf{F}^w_{\mathbf{D}} + \mathrm{Lin}_O(\mathbf{F}_{\mathbf{O}}), \qquad (1)$$

where $\mathbf{G}_{\mathbf{D}}$ is the depth gate computed as $\sigma(\mathrm{Lin}_D(\mathbf{F}_{\mathbf{O}}))$, and Lin is a linear layer. The gating mechanism ensures that erroneous flow does not degrade depth propagation. The fused features are processed through the remaining blocks of the, now shared, backbone and yield multi-resolution encoder features $\mathcal{C} = \{\mathbf{C}_i\}_{i=0}^{4}$. Finally, the warped neck features $\mathbf{F}_{i,t-1}^{w}$ are corrected using multi-resolution encoder features $\mathcal{C}$ in a residual formulation:

$$\mathbf{F}_{i,t} = \text{Conv}(\mathbf{C}_i || \mathbf{F}_{i,t-1}, \mathbf{G}_{\mathbf{F}}) + \mathbf{F}_{i,t-1}^{w}, \tag{2}$$

where $\mathbf{G}_{\mathbf{F}}$ is the feature gate controlling the correction process obtained via $\sigma(\text{MLP}_F(\mathbf{F}_{\mathbf{O}}))$, and "Conv" is a ResNet Block with gating applied in the bottleneck. The gating mechanism selectively propagates reliable corrections when needed while filtering out harmful residuals where the PFW depth features are already corrected. The concatenation of $\mathbf{C}_i || \mathbf{F}_{i,t-1}$ is utilized to make the correction from the Encoder aware of the previous frame features.

Keyframe selection. To ensure efficient propagation, VeloDepth minimizes redundant predictions. If the input remains stable, prior predictions are propagated, while VeloDepth has to re-initialize and predict from scratch when significant changes occur. In particular, we define a simple re-initialization heuristic based on optical flow via the magnitude of the flow and a warping-based difference metric. Formally, we incur a keyframe if and only if

$$\begin{aligned} \left\| w(\mathbf{1}_{H\times W}, \hat{\mathbf{O}}_{t-1}^{t}) \right\|_1 &\leq 0.2 \times 0.9^t \vee \\ \left\| \hat{\mathbf{O}}_{t-1}^{t} \right\|_1 &\geq 0.15 \times 0.9^t + 0.1, \end{aligned} \tag{3}$$

where $\mathbf{1}_{H\times W}$ is a $H \times W$ matrix of ones, $w(x, \mathbf{y})$ denotes warping x using flow $\mathbf{y}$, $\|\mathbf{X}\|_1 = \frac{1}{HW}\sum_{i,j} \|x_{ij}\|_2$, and t is the frame count since the last keyframe. The decay 0.9^t accounts for gradual degradation over time, balancing efficiency and robustness for long sequences.

3.2. Consistency

Maintaining temporal consistency is essential for online and real-time depth estimation. Ideally, the same 3D point should retain a consistent location across consecutive frames. However, traditional MDE models operate on independent images; this makes them highly sensitive to small input variations due to the absence of temporal constraints. To mitigate these issues, VeloDepth introduces a refined consistency loss formulation. A key limitation of previous methods [47, 57] is the lack of explicit camera motion compensation. Depth values propagated through warping reside in different coordinate frames, and without appropriate transformations, their direct comparison is inconsistent. To ensure equivariance against camera motion, VeloDepth applies the consistency loss on metric radial distance rather than raw depth values. Radial distance remains invariant to rotational transformations, ensuring that consistency is preserved across frames regardless of camera orientation. To address translational motion, a linear shift is computed by aligning the median-based centers of consecutive 3D point clouds:

$$\begin{gathered} \mathcal{L}_{\text{con}}(t-1,t) = \left\| w(\|\mathbf{P}_{t-1}\|_2, \mathbf{O}_{t-1}^{t}) - \|\mathbf{P}_t - \mathbf{t}\|_2 \right\|_1, \\ \mathbf{t} = \text{med}(\mathbf{P}_t) - \text{med}(\mathbf{P}_{t-1}), \end{gathered} \tag{4}$$

where $\mathbf{P} \in \mathbb{R}^{3\times H\times W}$ represents the 3D point map, $\mathbf{O}_{t-1}^{t}$ is the pseudo-ground-truth flow from [49], and $\text{med}(\cdot)$ computes the median over pixel and dimension-wise elements. Occlusions and disocclusions are masked out based on a forward-backward flow consistency check as per standard practice. This formulation enforces a pose-agnostic consistency constraint without requiring explicit extrinsic parameters, enabling robust and efficient depth propagation suitable for practical deployment. The consistency loss is applicable only for models that infer 3D points directly from RGB inputs, as it is formulated in terms of metric Euclidean distance. Additionally, the loss is computed bidirectionally, ensuring time-invariant consistency across frames: $\mathcal{L}_{\text{con}} = \mathcal{L}_{\text{con}}(t-1,t) + \mathcal{L}_{\text{con}}(t,t-1)$. Moreover, we propose to use it in conjunction with a temporal flip augmentation. This augmentation helps mitigate the forward-motion bias typically present in casual coherent videos, which would otherwise induce the network to always mimic forward ego-motion even when it is not present.

3.3. Network Design

Architecture. The proposed architecture consists of a "Base Model", specifically [35], although the former is adaptable to any metric MDE model, which comprises a "Base Encoder" and a "Base Decoder". In addition, VeloDepth involves a propagation network that integrates a multi-modal encoder, residual correction module, and optical flow refinement, as illustrated in Fig. 2. The multi-modal encoder is a convolutional network, specifically ConvNeXt-Tiny [21], with three input branches corresponding to different modalities: RGB, geometric depth, and optical flow. Each branch extracts dense features $\mathbf{F}_{\mathbf{X}} \in \mathbb{R}^{h\times w\times C\times 4}$, where $(h, w) = (\frac{H}{4}, \frac{W}{4})$, and $\mathbf{X} \in \{\mathbf{I}, \mathbf{D}, \mathbf{O}\}$. The features are processed through three shared blocks to produce the fused features $\mathbf{C}$, as described in Sec. 3.1. The processed fused features are multi-scale, producing outputs at four different resolutions, denoted as $\mathcal{C} = \{\mathbf{C}_i\}_{i=0}^{3}$. The optical flow refinement module processes $\mathbf{F}_{\mathbf{O}}$ using two convolutional layers interleaved with 2x bilinear upsampling and a leaky ReLU activation function. The residual module then corrects the neck features at each resolution, $\mathcal{F}_t = \{\mathbf{F}_{i,t}\}_{i=0}^{3}$, using the multi-modal and multi-resolution features $\mathcal{C}$, as detailed in (2). The full Base Model is applied to the first frame, which is treated as a keyframe, to generate the initial neck features, $\mathcal{F}_0$, while the base decoder processes the incoming refined features $\mathcal{F}_t$ for all subsequent frames $(t > 0)$ until the next keyframe is incurred as described in Sec. 3.1. The model outputs the pre-

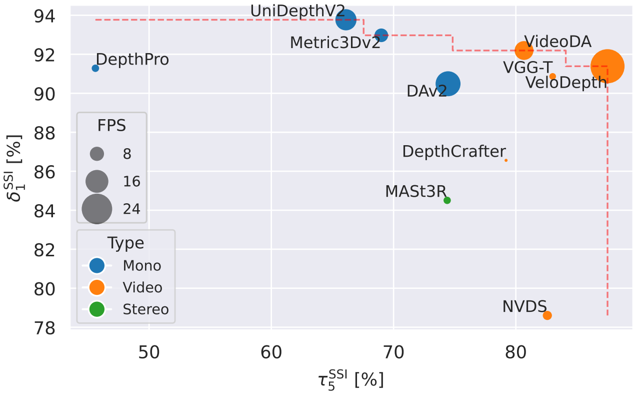

Figure 3. **Pareto optimal frontier** is evaluated in terms of combined accuracy (δ_1^{SSI}) and consistency (τ_5^{SSI}). Disk areas correspond to inference efficiency (FPS); the larger the area, the faster. VeloDepth strikes a positive tradeoff w.r.t. its Base Model (*i.e.* UniDepthV2), achieving a substantial improvement in consistency for a minor drop in accuracy.

dicted 3D point maps $\mathbf{P}_t \in \mathbb{R}^{3\times H\times W}$, thanks to intrinsics provided by the Base Model, along with the neck features $\mathcal{F}_t$, which are then propagated to the next frame $(t+1)$.

Optimization. The optimization strategy comprises five distinct loss functions targeting three main objectives: output accuracy, flow refinement, and consistency. Depth predictions are optimized using the $\mathrm{SI}_{\log}(\mathbf{D}^*, \mathbf{D})$ loss from [8], where $\mathbf{D}^*$ denotes ground-truth depth, and the $\mathrm{L}_{1,\mathrm{SSI}}(\mathbf{D}^*, \mathbf{D})$ loss from [37], computed over the entire video clip rather than per image. When GT depth is unavailable, the supervision is derived from the "Base Model" predictions. The stability and accuracy of depth predictions depend on ensuring that neck features remain sharp and do not degrade due to warping. Therefore, the corrected neck features are supervised by aligning them to the per-frame Base Model features ($\{\mathbf{F}^*_{i,t}\}_{i=0}^{3}$) using an L_1 loss: $\mathcal{L}_{\mathrm{F}}(\mathcal{F}^*_t, \mathcal{F}_t) = \sum_{i=1}^{4}\left(\frac{1}{C}\sum_{c=1}^{C}\left\|\mathbf{F}^*_{i,t,c} - \mathbf{F}_{i,t,c}\right\|_1\right)$. The refined optical flow $\mathbf{O}$ is supervised using pseudo-GT backward flow produced by SEA-RAFT [49] with an L_1 loss. Finally, the consistency between consecutive frames is enforced through the proposed bidirectional consistency loss $\mathcal{L}_{\mathrm{con}}$ described in Sec. 3.2. This formulation ensures that depth predictions remain stable over time while enabling accurate depth propagation across video sequences. The final loss is the sum.

4. Experiments

4.1. Experimental Setup

Datasets. The training dataset accounts for two different sources, in-the-wild without GT and depth datasets. The former is composed by Kinetics-700 [41], Moments-in-Time [25], and SAv2 [38], while the latter by TartanAir [45], Wild-RGBD [51], HabitatMatterport3D [36], PointOdyssey [58] and Waymo [43]. More details are given in the supplement. We evaluate the generalizability of models by testing them on 4 datasets not seen during training, in particular, ScanNet [5], Sintel [2], Bonn-RGBD [27], and TUM-RGBD [42].

Implementation details. VeloDepth is implemented in PyTorch [29] and CUDA [26]. Training uses the AdamW [22] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with an initial learning rate of 1×10^{-4}. A cosine annealing scheduler reduces the learning rate to one-tenth after 30% of total training iterations. We run 150k optimization iterations with 256 total images per iteration. The dataset sampling procedure follows a weighted sampler, where each dataset is weighted by its number of scene. We employ curriculum learning to progressively increase sequence length from 2 to 20 frames, using a linear schedule between 50k and 150k iterations. The idea behind curriculum learning is a progressive increase in sequence complexity, which stabilizes training when handling long video sequences. Since a single GPU can accommodate only 10 non-keyframe frames per iteration, initial frames of longer sequences are processed in "no grad" context. Our augmentations are both geometric and photometric, *i.e.* random resizing and cropping for the former type, and brightness, gamma, saturation, and hue shift for the latter. In addition, we employ temporal augmentation which flips the ordering of the frames in each batch with 50% probability. The Base Encoder and Decoder are frozen and initialized with UniK3D [34] weights. We randomly sample the image ratio per batch between 2:1 and 1:2 and between 0.2 and 0.4 Megapixel (MP). The training time amounts to 6 days on 8 NVIDIA RTX 4090. For the ablations, we run 80k training steps with the training pipeline as for the main experiments compressed from 150k to 80k steps.

Evaluation details. We evaluate on ScanNet following protocol from [13] and on Bonn-RGBD and TUM-RGBD following [12], while for Sintel all sequences are tested. Depth accuracy and consistency are assessed using δ_1 and τ_5 metrics, respectively. δ_1 measures the percentage of pixels whose predicted depth is within 25% of the GT depth. τ_5 measures consistency across frames by warping depth from $t-1 \rightarrow t$ using optical flow, applying ego-motion correction, and considering a pixel inlier when the difference is within 5% of the depth at t. This metric extends the accuracy evaluation used in OPW [47] and TCM [57], incorporating additional ego-motion compensation. Optical flow is either sourced from [49] or provided by the dataset itself. When per-frame depth predictions are rescaled to match GT depth for δ_1 or τ_5, we denote them as δ_1^{SSI} and τ_5^{SSI}. This rescaling enables fair comparisons with non-metric models while ignoring global scale inconsistencies in τ_5. GPU inference speed is measured on an NVIDIA RTX 3090 using synchronized timers. CPU inference speed is evaluated on an M1 Pro chip utilizing the MPS backend, as this setup closely approximates modern mobile processors such as A19 chip while keeping the testing simpler. Inference speed is mea-

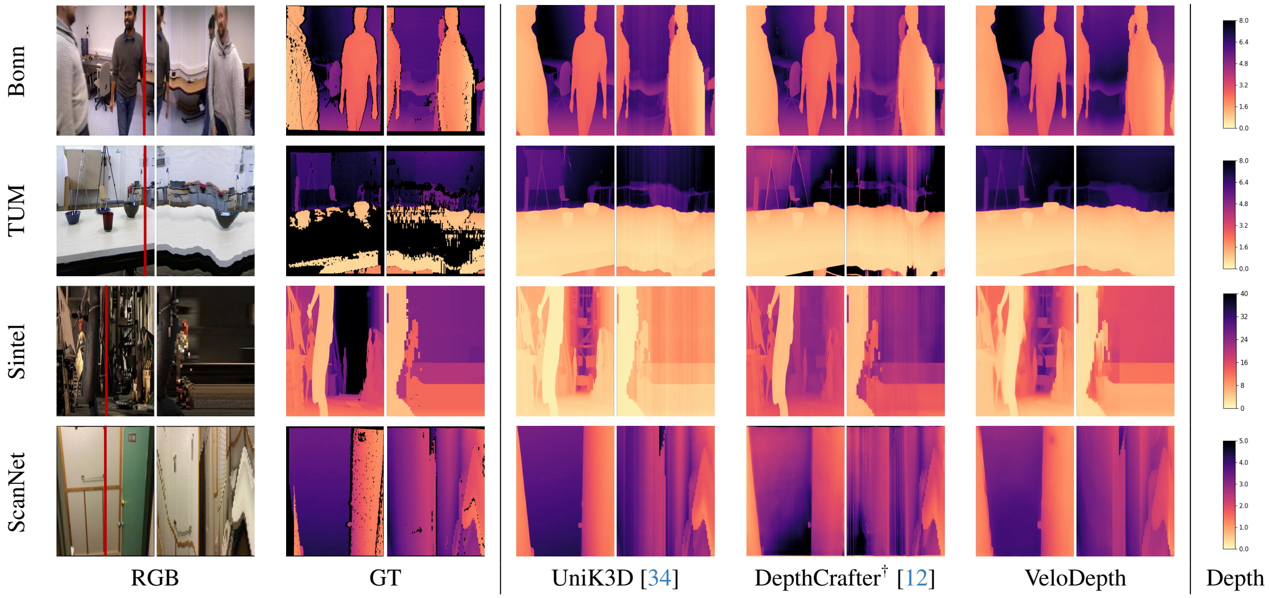

Figure 4. **Zero-shot qualitative results.** Each row corresponds to one test video sample from one domain. Each block shows the 6th frame and the video slices corresponding to the red line x-location in the first column. UniDepthV2 and VeloDepth outputs are inherently metric. No post-processing is applied. The last column represents the depth values w.r.t. "magma" colormap. (†): affine transformed to match GT. Best viewed on a screen and zoomed in.

sured over a 60-frame sequence and averaged per frame on 0.5 MP images. Both GPU and CPU benchmarks employ mixed precision. For the ablations, we evaluate VeloDepth by running it on the first 32 frames of each sequence and initializing on the first frame with the Base Model. All methods are evaluated in an online fashion: at frame t, each model has access to only frames $\leq t$. Direct comparisons to models operating offline would be misleading, as the latter exploit future information, which is not possible in causal settings. We provide offline evaluation in the supplements.

4.2. Comparison with The State of The Art

Table 1 presents a comprehensive evaluation of VeloDepth against state-of-the-art monocular, stereo, and video depth estimation methods across four distinct domains. It is worth noting that we report mainly scale- and shift-invariant metrics to increase the extensiveness of our comparison, but VeloDepth outputs metric predictions, which are evaluated more extensively in the supplements. Our method clearly demonstrates superior temporal consistency and computational efficiency compared to all competitors. In particular, when compared to a model with a similar runtime, such as [57], VeloDepth achieves significantly higher accuracy (+70.6%) and consistency (+18.5%). Furthermore, compared to the closest competitor in terms of consistency, [48], our approach not only improves accuracy by 12.8%, but also provides a 6.7× improvement in inference speed. However, VeloDepth can produce metric output, in contrast to most video-based methods, and in the metric-case, it ranks 1st and 2nd for consistency and accuracy, respectively.

While monocular depth estimators generally yield higher absolute accuracy, this accuracy typically comes at the expense of temporal consistency. For instance, VeloDepth notably surpasses the consistency of its monocular base model (UniK3D) by 13.3% and 21.4% for metric and affine-invariant evaluation, respectively, highlighting the strength of our propagation-based approach. Moreover, as illustrated in Fig. 4, despite being a metric depth estimator susceptible to global scale jitter, unlike relative depth estimators, ourmodel still maintains remarkably high consistency. Traditional monocular and repurposed-online depth methods exhibit substantial frame-to-frame jitter as color jumps, indicative of inconsistent predictions, whereas VeloDepth effectively mitigates this issue through its feature propagation mechanism. It is important to note that VeloDepth inherits occasional inconsistencies from keyframe predictions produced by the monocular base model, especially when significant scene changes trigger the computation of a new keyframe. Despite this, the propagated intermediate predictions remain highly stable.

Finally, as depicted in Fig. 3, VeloDepth establishes a Pareto-optimal frontier, clearly demonstrating the best available trade-off between consistency, accuracy, and computational efficiency among current depth estimation methods.

4.3. Ablations

We conduct an extensive ablation study to evaluate the impact of key architectural and optimization components. This includes analyzing input modalities in Table 2, gating mechanisms in Table 3, loss functions in Table 4, the choice of optical flow in Table 5, and the proposed inductive biases in

Table 1. **Comparison on zero-shot evaluation.** All methods are evaluated in an online fashion. The “Type” column indicates the original task tackled, **M**: monocular, **S**: stereo, **V**: video or multi-view. Profiling is run on 60 frames of 0.5MP averaged per frame. $\mathrm{FPS}_{\mathrm{GPU}}$ is measured on an RTX 3090 and $\mathrm{FPS}_{\mathrm{CPU}}$ on an M1 chip, both with half precision. ‡: camera GT at inference time.

Method	**Type**	TUM-RGBD		ScanNet		Sintel		Bonn-RGBD		**Aggregate**		**Efficiency**			
		$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$	$\mathrm{FPS}_{\mathrm{GPU}} \uparrow$	$\mathrm{FPS}_{\mathrm{CPU}} \uparrow$	Params[M] ↓	FLOP[T] ↓
DAv2 [53]	M	94.0	79.9	98.1	72.8	73.0	55.0	98.5	95.1	90.5	74.5	16.9	1.1	335.3	2.0
Metric3Dv2‡ [11]	M	96.1	74.0	99.0	81.3	77.2	31.3	**99.1**	86.0	93.0	69.0	7.1	0.7	411.9	3.5
DepthPro [1]	M	94.3	58.2	97.2	43.9	73.7	17.6	99.0	63.2	91.3	45.6	2.8	0.2	952.0	4.8
UniDepthV2 [35]	M	**96.6**	71.0	98.5	74.8	**80.7**	34.5	99.0	82.0	**93.8**	66.1	13.4	1.0	353.8	2.2
UniK3D [34]	M	**96.6**	69.4	98.5	76.1	80.5	32.2	99.0	79.7	93.6	64.4	12.1	1.0	375.3	2.6
MASt3R [18]	S	88.1	77.4	96.7	77.9	59.3	55.6	93.9	86.6	84.5	74.4	2.7	0.7	688.6	3.2
CS-LSTM [57]	V	24.7	81.6	26.0	59.7	8.0	71.1	27.7	73.0	20.8	69.0	25.3	2.5	15.0	**0.4**
NVDS [48]	V	76.6	83.9	85.1	88.7	67.6	65.8	88.0	**97.3**	78.6	82.6	3.9	0.5	432.9	2.2
ChronoDepth [40]	V	58.5	76.6	74.1	72.4	26.5	34.2	61.8	68.8	55.2	63.0	0.3	OOM	1522.3	19.6
DepthCrafter [12]	V	83.5	80.0	94.4	83.2	74.3	64.9	98.7	96.3	86.6	79.2	0.2	OOM	1524.6	27.1
VideoDA [3]	V	94.9	80.1	98.1	87.5	76.8	64.2	99.0	96.0	92.2	80.7	11.3	OOM	384.4	3.9
VGG-T [44]	V	90.2	84.4	**99.3**	86.5	74.1	68.9	99.0	96.6	90.9	83.0	2.3	OOM	1261.0	8.8
VeloDepth	V	94.8	**89.1**	96.2	**89.1**	76.1	**75.4**	98.4	96.3	91.4	**87.5**	**26.2**	**2.7**	409.4	0.7

Table 2. **Input modalities.** $\mathbf{D}_{t-1}^{w}$ indicates the previous frame warped depth and $\hat{\mathbf{O}}_{t-1}^{t}$ the initial optical flow. Current RGB is always used as input.

	$\mathbf{D}_{t-1}^{w}$	$\hat{\mathbf{O}}_{t-1}^{t}$	$\delta_1 \uparrow$	$\tau_5 \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$
1	✗	✗	63.1	78.0	76.8	81.1
2	✓	✗	62.3	88.3	78.4	90.6
3	✗	✓	69.3	89.7	82.6	90.3
4	✓	✓	71.3	91.8	82.9	92.4

Table 3. **Flow-based gating.** $\mathbf{F}_{t-1}^{w}$ corresponds to previous frame warped decoder features, $\mathbf{F}_{\mathbf{D}}^{w}$ to depth features, and $\mathbf{F}_{\mathbf{I}}$ image features after the first respective layers. σ indicates if the element-wise sigmoid-based gating is applied. “$1-\sigma$” represents inverse gating w.r.t. $\mathbf{F}_{\mathbf{D}}^{w}$ one.

	$\mathbf{F}_{t-1}^{w}$	$\mathbf{F}_{\mathbf{D}}^{w}$	$\mathbf{F}_{\mathbf{I}}$	$\delta_1 \uparrow$	$\tau_5 \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$
1	✗	✗	✗	71.3	91.8	82.9	92.4
2	σ	✗	✗	73.1	93.0	84.3	93.9
3	σ	σ	✗	74.1	92.7	84.7	93.5
4	σ	σ	$1-\sigma$	72.8	91.7	83.9	92.5

Table 4. **Optimization.** Flip refers to using the temporal flipping augmentation. $\mathcal{L}_{\mathrm{con}}$ and $\mathrm{out}_{\mathcal{L}_{\mathrm{con}}}$ indicate if the proposed consistency loss is employed and on which output, respectively, with **D** referring to depth and **R** to euclidean distance.

	Flip	$\mathcal{L}_{\mathrm{con}}$	$\mathrm{out}_{\mathcal{L}_{\mathrm{con}}}$	$\delta_1 \uparrow$	$\tau_5 \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$
1	✗	✗	n/a	68.3	89.9	81.6	90.4
2	✓	✗	n/a	72.5	90.6	83.8	92.2
3	✓	✓	**D**	71.5	88.8	82.6	92.1
4	✓	✓	**R**	74.1	92.7	84.7	93.5

the Propagation Module in Table 6 and keyframe selection in Fig. 5. Each table underlines a row, which corresponds to the (partial) configuration used in the final VeloDepth.

Input Modalities. The results in Table 2 highlight the role of different input modalities in VeloDepth. Adding the PFW depth prediction (row 2) significantly enhances depth consistency, demonstrating the importance of propagating prior depth estimates. However, this addition does not directly improve depth accuracy, suggesting that the network primarily learns to preserve existing structures, *i.e.* zero residuals $\mathbf{C}_{\mathrm{G}}$ rather than actively refining depth estimates. Integrating optical flow further improves both consistency and accuracy by allowing the model to identify incorrect PFW features. This enables targeted feature corrections while maintaining stability by setting the residual to zero in regions where depth estimates are already reliable. The best performance (row 4) is achieved when both PFW depth and optical flow are included, as VeloDepth gains a comprehensive understanding of prior depth information, its motion dynamics, and where to trust these estimates. The RGB image is always included as a reference modality in all experiments.

Gating Mechanisms. Table 3 presents an analysis of the gating mechanisms applied at different stages in the network. The most significant impact is observed when gating is applied to PFW neck features (row 2), as it directly regulates whether residual corrections from the Propagation Module influence the next frame. This prevents the network from being implicitly biased to predicting always zeros, since for large parts of images correction is typically null and supervised to be null. Rows 3 and 4 evaluate gating during modality fusion, where gating depth slightly improves accuracy. However, this effect is partially redundant, as the PFW feature gating (row 2) already ensures that residuals are only applied when necessary, effectively preventing unnecessary corrections. In row 4, an inverse gate is introduced on RGB features, enforcing a convex combination of depth and RGB information. However, this leads to a detrimental effect on performance. We speculate that this occurs because RGB features are never warped, meaning that applying a gating function to them results in information loss rather than selective refinement.

Loss. Table 4 presents the ablation results on the training pipeline, focusing on flip augmentation (row 2) and the proposed loss functions (rows 3 and 4). The results indicate that flip augmentation enhances accuracy by mitigating the forward motion mimicking bias, as discussed in Sec. 3.2. The proposed consistency loss, introduced in Sec. 3.2, significantly improves both depth consistency and accuracy. By en-

Table 5. **Flow.** $\mathbf{O}_{t-1}^{t}$ is the flow used to perform warping. MV refers to using MPEG-4 motion vectors, DIS utilizes [16] and RAFT [49]. The subscript R stands for usage of the corresponding optical flow refined via FlowRefine.

	$\mathbf{O}_{t-1}^{t}$	$\delta_1 \uparrow$	$\tau_5 \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$
1	MV	69.9	89.9	79.4	90.2
2	DIS	70.2	90.8	81.8	91.7
3	MV_{R}	72.9	92.1	83.8	92.7
4	$\mathrm{DIS}_{\mathrm{R}}$	74.1	92.7	84.7	93.5
5	RAFT	74.3	93.2	85.0	93.8

Table 6. **Propagation.** Prop refers to usage of propagation via flow-based warping, while Init to the Base Model initialization. $\mathrm{Enc}_{\mathrm{Fast}}$ indicates which encoder is used for fast-frames, with fusion and refinement when applicable: "Base (no prior)" means the Base Encoder is used but no prior information is passed to.

	Prop	Init	$\mathrm{Enc}_{\mathrm{Fast}}$	$\delta_1 \uparrow$	$\tau_5 \uparrow$	$\delta_1^{\mathrm{SSI}} \uparrow$	$\tau_5^{\mathrm{SSI}} \uparrow$
1	✗	✗	Ours	54.6	74.5	70.6	82.2
2	✓	✗	Ours	62.7	90.7	75.8	92.4
3	✓	–	Base (no prior)	78.1	77.5	86.3	79.1
4	✓	✓	Ours	74.1	92.7	84.7	93.5

forcing similarity between matching locations in consecutive frames, up to a translation, the loss provides an additional supervision signal that reinforces temporal stability. Conversely, applying the consistency loss directly to depth values instead of Euclidean distances leads to a performance drop, as shown in row 3. This result suggests that enforcing consistency in depth space alone introduces incorrect supervision signals, leading to inconsistencies in depth predictions.

Flow. The effect of different optical flow methods used for warping is examined in Table 5. The tested approaches include motion vectors (MV) extracted from MPEG-4 video encoding, DIS [16] flow, and SEA-RAFT [49] flow. Both MV and DIS flow can be used directly or refined via the ”Flow Refine” convolutional layers described in Sec. 3.3 and illustrated in Fig. 2, leading to refined versions MV_{R} and $\mathrm{DIS}_{\mathrm{R}}$. The results exhibit a diminishing return effect when increasing the quality of the optical flow $\mathbf{O}_{t-1}^{t}$, indicating that beyond a certain threshold, further improvements in flow estimation yield smaller gains. Comparing row 1 to row 3 and row 2 to row 4, we observe that flow refinement, despite its relatively low capacity, improves both accuracy and consistency. This suggests that the refinement step effectively denoises the warping flow, leading to better propagation.

Propagation. Table 6 evaluates the role of initialization and propagation strategies. Row 1 represents a standard image-based MDE model, where the PFW depth $\mathbf{D}_{t+1}^{w}$ and features $\mathbf{F}_{t+1}^{w}$ are not utilized, thus we do not predict a residual but the full neck features $\mathbf{F}_{t}$ every frame. Row 2 corresponds to VeloDepth without keyframe initialization from the Base Model, Row 3 represents a model without any prior input modality but RGB (processed by the Base Encoder) and with flow-based propagation of previous neck features. Comparing row 1 and row 2 highlights the importance of prior knowledge, framing the problem as a propagation rather than a prediction significantly improves both accuracy and consistency. The comparison between row 2 and row 4 further emphasizes that a high-capacity initialization is highly beneficial for accuracy. Additionally, the results in row 3 *vs*. row 4 show that while a high-capacity model enhances accuracy, it does not necessarily improve consistency. Instead, the prior information from previous frames plays a crucial role in ensuring stable predictions. This confirms that consistency is driven primarily by leveraging prior information rather than by increasing the capacity of the propagation mechanism alone.

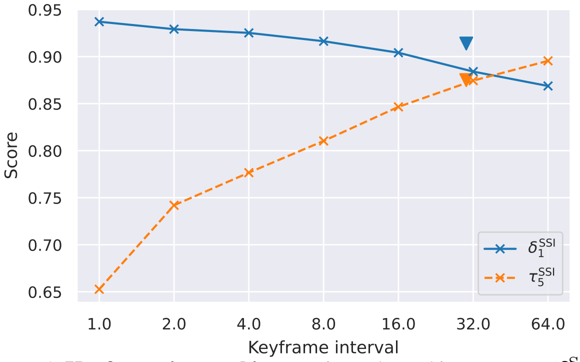

Figure 5. **Keyframe interval impact** is evaluated in accuracy (δ_1^{SSI}) and consistency (τ_5^{SSI}). Plot lines refer to fixed keyframe intervals on the x-axis. ▼ refers to our keyframe selection mechanism, accounting for an average keyframe interval of 30.

Keyframe Selection. Fig. 5 illustrates the impact of different keyframe selection strategies. We compare selecting keyframes at fixed intervals against our proposed heuristic described in Sec. 3.1. Despite its simplicity and minimal tuning, our heuristic effectively maintains temporal consistency without sacrificing accuracy. We note that increasing the distance between keyframes enhances consistency but negatively impacts accuracy, as the Propagation Module tends to produce overly smoothed results in the long run.

5. Conclusion and Limitations

We introduced VeloDepth, a novel online video depth estimation approach that leverages temporal priors to improve consistency, efficiency, and accuracy. Our Propagation Module refines and propagates depth across frames using optical flow and residual corrections, achieving strong temporal stability without relying on computationally expensive recurrent architectures. However, the method is sensitive to keyframe quality, as errors may be propagated over time. While performance depends on the quality of the optical flow input, empirical results demonstrate notable robustness and flexibility. Extensive zero-shot evaluations further show that VeloDepth delivers superior temporal consistency and a strong trade-off between accuracy, stability, and runtime efficiency, making it well suited for real-world applications.

Acknowledgment. This work is funded by Toyota Motor Europe via the research project TRACE-Zürich.

References

[1] 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, 2, 7

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012. 5

[3] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos, 2025. 1, 2, 7

[4] Arun CS Kumar, Suchendra M Bhandarkar, and Mukta Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 283–291, 2018. 2

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[6] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 1

[7] Xingshuai Dong, Matthew A Garratt, Sreenatha G Anavatti, and Hussein A Abbass. Towards real-time monocular depth estimation for robotics: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):16940–16961, 2022. 1

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2366–2374. Neural information processing systems foundation, 2014. 1, 2, 5

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2002–2011, 2018. 2

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, page 1735–1780, 1997. 2

[11] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 2, 7

[12] 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. 1, 2, 5, 6, 7

[13] Bingxin Ke, Dominik Narnhofer, Shengyu Huang, Lei Ke, Torben Peters, Katerina Fragkiadaki, Anton Obukhov, and Konrad Schindler. Video depth without video models. *arXiv preprint arXiv:2411.19189*, 2024. 5

[14] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9492–9502, 2024. 2

[15] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[16] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3, 8

[17] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *Proceedings of the International Conference on 3D Vision (3DV)*, pages 239–248, 2016. 2

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

[19] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5162–5170, 2015. 2

[20] X. Liu, A. Sinha, M. Ishii, G. D. Hager, A. Reiter, R. H. Taylor, and M. Unberath. Dense depth estimation in monocular endoscopy with self-supervised learning methods. *IEEE Transactions on Medical Imaging (TMI)*, 39(5):1438–1447, 2020. 1

[21] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. 4

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*, 2017. 5

[23] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. In *ACM Transactions on Graphics (SIGGRAPH)*. ACM, 2020. 1, 2

[24] Michele Mancini, Gabriele Costante, Paolo Valigi, Thomas A. Ciarfuglia, Jeffrey Delmerico, and Davide Scaramuzza. Toward domain independence for learning-based monocular depth estimation. *IEEE Robotics and Automation Letters*, 2 (3):1778–1785, 2017. 2

[25] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, pages 1–8, 2019. 5

[26] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53, 2008. 5

[27] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. 5

[28] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[30] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5:6813–6820, 2020. 2

[31] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3Depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1600–1611. IEEE, 2022. 2

[32] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. iDisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[33] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10106–10116, 2024. 1, 2

[34] Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniK3D: Universal camera monocular 3d estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 5, 6, 7

[35] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, (01):1–14, 5555. 2, 4, 7

[36] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Advances in Neural Information Processing Systems (NIPS)*, 2021. 5

[37] 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 Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 44(3):1623–1637, 2020. 1, 2, 5

[38] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5

[39] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5506–5514, 2016. 2

[40] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Vitor Guizilini, Yue Wang, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors, 2024. 1, 2, 7

[41] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset. *ArXiv*, abs/2010.10864, 2020. 5

[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012. 5

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 5

[44] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 7

[45] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 5

[46] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 1

[47] Yiran Wang, Zhiyu Pan, Xingyi Li, Zhiguo Cao, Ke Xian, and Jianming Zhang. Less is more: Consistent video depth estimation with masked frames modeling. In *Proceedings of the 30th ACM International Conference on Multimedia*, page 6347–6358. Association for Computing Machinery, 2022. 4, 5

[48] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9466–9476, 2023. 1, 2, 6, 7

[49] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. *arXiv preprint arXiv:2405.14793*, 2024. 4, 5, 8

[50] Yiran Wang, Min Shi, Jiaqi Li, Chaoyi Hong, Zihao Huang, Juewen Peng, Zhiguo Cao, Jianming Zhang, Ke Xian, and

Guosheng Lin. Nvds$^+$: Towards efficient and versatile neural stabilizer for video depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2024. 2

[51] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22378–22389, 2024. 5

[52] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10371–10381, 2024. 1, 2

[53] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 2, 7

[54] Rajeev Yasarla, Hong Cai, Jisoo Jeong, Yunxiao Shi, Risheek Garrepalli, and Fatih Porikli. Mamo: Leveraging memory and attention for monocular video depth estimation, 2025. 2

[55] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9043–9053, 2023. 1, 2

[56] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3906–3915. IEEE, 2022. 1, 2

[57] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 4, 5, 6, 7

[58] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19855–19865, 2023. 5

[59] Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does computer vision matter for action? *Science Robotics*, 4, 2019. 1