Stereo Matching in Time: 100+ FPS Video Stereo Matching for Extended Reality

Ziang Cheng*, Jiayu Yang*, Hongdong Li[2†]
[1]Tencent XR Vision Labs, [2]Australian National University
{ziang.cheng, jiayu.yang, hongdong.li}@anu.edu.au

(a) Virtual Tour

(b) Virtual Scenes and 6-DoF Camera Trajectories

(c) Rendered Photo-realistic Images

Figure 1. **The XR-Stereo dataset:** (a) We collected high-fidelity camera movement trajectories by taking virtual tours of virtual scenes and recording 6-DoF camera poses from VR/AR HMDs. (b) Examples of the captured virtual scenes and 6-DoF camera trajectories. (c) Examples of rendered photo-realistic images.

Abstract

Real-time Stereo Matching is a cornerstone task for Extended Reality (XR) applications, such as 3D scene understanding, video pass-through, and mixed-reality games. Despite significant advancements, getting accurate depth information in real time on a low-power mobile device remains a challenge. One of the main difficulties is the lack of high-quality indoor video stereo data captured by head-mounted VR or AR glasses. To address this, we introduce a novel video stereo synthetic dataset that comprises photorealistic renderings of various indoor scenes and realistic camera motion captured by a moving VR/AR head-mounted display (HMD). Our newly proposed dataset enables one to develop a novel framework for continuous video-rate stereo matching.

As another contribution, we also propose a new video-based stereo matching approach tailored for XR applications, which achieves real-time inference at an impressive 134fps on a standard desktop computer, or 30fps on a battery-powered HMD. Our key insight is that disparity and contextual information are highly correlated and redundant between consecutive stereo frames. By unrolling an iterative cost aggregation in time (i.e. in temporal dimension), we are able to distribute and reuse the aggregated features over time. This leads to a substantial reduction in computation without sacrificing accuracy. We conducted extensive evaluations and demonstrated that our method achieves superior performance compared to the current state-of-the-art, making it a strong contender for real-time stereo matching in VR/AR applications. Our dataset is released on https://github.com/za-cheng/XR-Stereo.

1. Introduction

Extended reality (or XR in short) is a collective term that refers to immersive technologies, including Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR).

Real-time Stereo Matching is key algorithm running on a VR/AR headset, which enables a wide range of applications such as visual passthrough and 3D mapping. Efficient stereo matching is particularly important for standalone (untethered) VR/AR headsets, as these are typically low-power mobile devices with limited computing power. Therefore, any computational overhead from stereo matching algorithms can reduce the responsiveness of human in-

*Equal contribution; †ARC DP220100800.

teraction and quickly drain the headset's battery, ultimately diminishing the overall user experience. Hence, it is imperative to develop highly efficient stereo algorithms that can provide accurate estimations in real-time on such low-power devices. One promising direction is to utilize temporal information. For high-frequency video stereo matching, consecutive frames can have large overlapped region. The redundant overlapped region in the temporal dimension can be utilized to reduce the computation overhead per frame.

A significant roadblock to the development of aforementioned algorithms is the absence of high-quality indoor stereo video datasets tailored for XR scenarios. Our examination of the current stereo datasets reveals several limitations that inhibit our exploration of temporal redundancy for efficient video stereo matching in XR scenarios.

Insufficient indoor environments. Indoor scenarios present unique challenging effects, including large textureless area such as wall or floor, transparent windows and mirror reflections. Standard stereo datasets such as SceneFlow [20] were not built for indoor stereo matching.

Absence of video sequences. Among the few dataset that provide indoor scenarios, most are targeted on single-frame stereo matching or lack accurate camera poses (*e.g.* [2, 30]), which considerably diminishes the capabilities of stereo methods, hindering the exploration of temporal relations between consecutive frames.

Lacking photo-realism. For the few possible options, we find [31] suffers subpar texture and shading quality, while [25] only provide gray-scale images. The diversity of indoor environments is also limited in both datasets, making it difficult to generalize to the wider indoor XR domain.

In this paper, we propose two novel contributions to facilitate the development of real-time stereo matching and other 3D vision tasks in XR scenarios. Our first contribution is a novel indoor XR video stereo dataset, which comprises photo-realistic stereo video sequences of various indoor environments rendered by a physically-based path tracing rendering engine, along with realistic and complex 6-DoF camera trajectories captured by VR/AR head-mounted displays (HMDs). Furthermore, the dataset includes various complete and accurate ground-truth labels, such as disparity, depth, optical flow, normal *etc*. Our novel video stereo dataset enables us to propose our second contribution, which is a video stereo matching algorithm that can achieve 100+fps inference speed on a standard desktop computer while maintaining comparable precision to current state-of-the-art methods. We achieve this by exploring the computational redundancy in the temporal dimension, based on the key insight that disparity and contextual information are highly correlated between consecutive stereo frames. Specifically, we assume known camera pose, and unroll an iterative cost aggregation network into the temporal dimension through temporal warping, which enables us to distribute and reuse the aggregated feature over time and leads to a substantial reduction in computation without sacrificing accuracy. Extensive experiments demonstrate that our method achieves significant improvements in inference speed compared to current state-of-the-art methods while maintaining comparable precision, making it a strong contender for real-time stereo matching in XR applications.

Our contributions are summarized as follows:

- A high-fidelity synthetic dataset for video stereo matching in indoor XR scenarios, consisting of photo-realistic stereo sequences and 6-DoF realistic camera motion captured by VR/AR head-mounted displays.
- A video stereo matching pipeline that exploits temporal redundancy of scene geometry to achieve real-time inference with high accuracy.
- Extensive experiments and comparisons with existing methods show that our method can maintain accuracy comparable to state-of-the-art methods while achieving an impressive 134fps inference speed on a desktop computer.

2. Related work

Video stereo datasets. A huge roadblock for developing efficient video stereo algorithms for indoor XR scenario is the lack of high-quality video stereo datasets in indoor environment. Specifically, we find several limitations of existing stereo datasets, including (1) Deficiency of indoor scenarios, (2) absence of video sequence and (3) lacking photo-realism, that largely blocked our way to explore the utilization of temporal information redundancy to develop a highly efficient video stereo matching algorithms dedicated for indoor XR scenario (see Table 1 for a summary). Existing stereo datasets are collected from various domains, including autonomous driving [11, 15], robotics [25, 31], movie [5] or synthesized from 3D models for general stereo matching training [20]. Indoor scenarios present unique challenging effects, including large texture-less ares such as wall or floor, transparent windows and mirror reflections. Few existing datasets [23–25, 30, 31] provide indoor scenes for training stereo matching to overcome these challenges. Among them, most are targeted on single-frame stereo matching [23, 24] and do not provide high-quality real-time video sequences, which considerably diminishes the capabilities of current stereo methods, limiting them to single-frame setups and hindering the exploration of temporal relations. For the very few stereo datasets that offer indoor stereo video sequences, [30] do not provide camera pose. Only [25, 31] provide accurate camera poses. Lack of camera data can ultimately limit the exploration of utilizing temporal relation, as camera pose is required for accurately model the relation between pixels in consecutive video frames. For these two possible op-

tions, we find [31] exhibits subpar rendering quality and lacks the photo-realistic effects present in indoor XR environments, while [25] only provide low resolution gray-scale images. Their variety of indoor environments is also limited for training algorithms dedicated to the indoor XR domain.

To facilitate the development of real-time stereo matching and other 3D vision tasks in XR scenarios, we propose a novel indoor video stereo dataset that utilizes physically-based path tracing to achieve photorealistic rendering of stereo pairs, along with *real* 6-DoF movement captured by a head-mounted display (HMD) as the camera trajectory. Our dataset is designed and implemented to a high standard to facilitate the evaluation of our approach and promote further research in Extended Reality scenarios.

Stereo Matching. Stereo Matching is a long-standing task in computer vision, involving the estimation of the disparity between a stereo image pair. Classical methods [4, 14, 24] compute hand-crafted matching costs along with local, semi-global, or global cost aggregation to achieve accurate, complete, and smooth disparity estimation. In recent years, deep learning-based stereo methods [13, 19, 29, 32, 35] have achieved superior results by utilizing learned image features to build cost volumes, learned cost aggregation networks, and learned disparity regression.

A recent approach, RAFT-Stereo [18], employs a GRU structure to iteratively lookup matching costs and refine disparity estimation. Another approach, CREStereo [17], extends this method with a cascade framework and adaptive correlation.

Real-time Stereo Matching. Few existing stereo matching methods focus on achieving real-time inference with high accuracy. Classical methods, such as PatchMatch Stereo [4], are capable of running in real-time, but their estimation accuracy is far inferior to current learning-based methods due to their hand-crafted cost metric and cost aggregation. These methods are largely outperformed by deep learning-based counterparts. Few recent deep learning-based methods have explored real-time stereo matching [9, 18]. DeepPruner [9] is based on PatchMatch Stereo [3, 4], which implements a differentiable PatchMatch layer for pruning the disparity searching space in low computational cost to alleviate the computational overhead of subsequent deep cost volume and deep cost aggregation network. StereoNet [16] propose to use hierarchical refinement to improve efficiency. HiTNet [28] use a multi-resolution initialization paired with coarse-to-fine slanted window based propagation to improve efficiency. Coex [1] propose a Guided Cost Volume Excitation to alleviate the computation of 3D convolutions. Most closely related to our method is RAFT-Stereo [18], which utilizes a multilevel recurrent field to iteratively refine disparity estimation. It iteratively looks up matching cost from a pre-built cost volume and uses an iterative cost aggregation network to gradually refine the disparity estimation. Such a strategy has shown great promise in terms of accuracy and generality, but its runtime increases linearly with the increasing number of iterations. Therefore, it has to trade-off accuracy by reducing the number of iterations to improve inference speed.

Video Stereo. Temporal information can substantially contribute to both the accuracy and efficiency of stereo matching; however, recent deep stereo methods rarely explore the utilization of temporal information. Classical methods like Patchmatch Stereo [4] utilize temporal relation by propagating previous disparity estimations to the current frame as disparity hypotheses but lack further utilization of contextual information over time. Open-World Stereo [38] builds an LSTM connecting latent features along the time dimension. However, they heavily focus on the unsupervised open-world stereo setup and provide very limited insight or evaluation on the utilization of temporal information for improving stereo matching accuracy or efficiency. DeepVideoMVS [10] adopted similar LSTM structure for temporal multi-view stereo, where the temporal matching require a static scene that is infeasible in XR scenario. Very recently, a concurrent work TemporalStereo [37] also explored the utilization of temporal information in stereo matching. They extend a single-frame coarse-to-fine estimation framework similar to cascade stereo methods [12, 34], and warp previous cost volume into the current view with hand-crafted statistical fusion. They also warp previous disparity to the current time frame for extra disparity hypothesis. Based on these simple temporal extensions, their temporal model yields incremental accuracy improvement over their single-frame baseline but is slower at runtime. Unlike TemporalStereo, our framework is specifically designed to reduce the computational cost per frame by learning temporal cost aggregation. Our model can better utilize temporal information and can achieve 3x to 5x faster inference speed with superior accuracy than our single-frame baseline.

3. XR-Stereo Dataset

We first introduce our new video stereo synthetic dataset, XR-Stereo, which contains 60K stereo images in 640x480 resolution. We design our dataset in an indoor extended reality (XR) setup, where the stereo cameras are mounted on a head mounted display (HMD). Below we introduce details of this new dataset.

3.1. Indoor Scenes

We use a set of 13 carefully crafted, photo-realistic 3D virtual indoor environments. As shown in Figure 1(b), the virtual scenes selected for this research primarily focus on household environments such as living rooms, kitchens, bedrooms, and study rooms, while also featuring a few additional environments such as office, hospital and hotel room.

Dataset	Year	Type	Scenario	Camera Pose	Camera Motion	Video	Disparity
Sintel [5]	2012	Synthetic	Movie	Virtual	Virtual Camera	50 frames	Rendered
KITTI Stereo [11]	2012	Real	Road	Estimated	Driving	N/A	LiDAR
KITTI VO [11]	2012	Real	Road	Estimated	Driving	10 Hz	LiDAR
Middlebury [23]	2014	Real	Laboratory	N/A	N/A	N/A	Structured Light
SceneFlow [20]	2016	Synthetic	Various	Virtual	Spline	N/A	Rendered
ETH3D [25]	2017	Real	Indoor/Outdoor	Estimated	Robotics	13.6 Hz	Laser Scan
Apollo [15]	2018	Real	Road	Estimated	Driving	30 Hz	LiDAR
TartanAir [31]	2020	Synthetic	Various	Virtual	Robotics	10Hz-30Hz	Rendered
IRS [30]	2021	Synthetic	Indoor	Virtual	Spline	N/A	Rendered
XR-Stereo (Ours)	2023	Synthetic	**Indoor XR**	**HMD Recorded**	**Human Head Movement**	**30 Hz**	Rendered

Table 1. Stereo Matching Datasets. Our proposed XR-Stereo dataset focus on indoor extended reality (XR) scenario and provide 6-DoF camera motion recorded by VR/AR HMD through our virtual tour pipeline.

3.2. Recorded 6-DoF Camera Trajectory

We aim to obtain accurate camera poses of realistic movement trajectories by capturing the motion of a virtual reality/augmented reality (VR/AR) head-mounted display (HMD) as its wearers walk through virtual scenes. To accomplish this, we have developed a virtual tour pipeline that enables users to explore virtual scenes while wearing a VR/AR HMD. As illustrated in Figure 1(a), the pipeline consists of a real-time stereoscopic viewport rendering engine that is connected to the VR/AR HMD using OpenXR APIs. The pipeline utilizes the 6-DoF head pose of the HMD in real-time and associates it with the virtual head location in the virtual scene. The corresponding stereoscopic viewport images of the virtual head location are then rendered in real-time and streamed back to the HMD. This allows the wearer to freely explore the virtual scene while physically walking in the real world, subject to the physical space available for walking. Figure 1(b) shows examples of captured head trajectories overlaid on virtual scene examples. To ensure a diverse set of trajectories, we collected trajectories from different users with varying movement styles for each virtual scene, for a total of 17 trajectories.

3.3. Physically-based rendered images

Existing synthetic datasets typically utilize real-time rendering engine such as Lumen [27] or Eevee [7] for its high efficiency and low computational cost. However, they are sub-optimal for training XR-oriented stereo matching networks due to the lack of challenging real-world photometric effects (such as specular reflections and transmitted lights) that are present in indoor environments. To simulate complex real-world optics, we use Blender's path tracing engine Cycles [7]. All the virtual scenes are shaded with Physics-Based Rendering materials for photometric fidelity. With this setup, we are able to synthesize a diverse range of optical effects in high fidelity, including mirror reflection, refraction, subsurface scattering and secondary reflection. Physically-based path tracing consumes significantly more computational resources over its real-time counterparts. We implement our rendering pipeline using 80 distributed computing nodes, each node containing 4 NVIDIA Tesla V100 GPUs. Rendering the entire dataset took around two weeks.

3.4. Lens and lighting effects

Our dataset draws inspiration from the SceneFlow dataset [21] and provide rendering of two sets of images, which we refer to as cleanpass and finalpass. The cleanpass set serves as a baseline, providing a clear and unadulterated view of the scene radiance, while the finalpass set contains real-world lens and lighting effects such as including motion-blur, defocus, rolling-shutter, lens glare and indoor light flickering.

3.5. Data Type and Potential Applications

We provide various types of data and ground-truth, including pixel-wise raw light intensity, RGB image, rendered disparity, rendered depth, optical flow, surface normal, visual ray vectors, surface mesh, instance labels, object bounding box, diffuse/specular/transmission layer separation etc. These data can facilitate the development of various indoor XR applications, such as indoor 3D reconstruction or Visual See-through (VST), and generic computer vision tasks such as monocular depth estimation, stereo matching, multi-view stereo, 3D reconstruction, structure from motion, visual odometry, SLAM, etc.

3.6. Limitation

The current version of our XR-Stereo dataset does not contain moving objects. In XR scenarios, various applications require modeling of moving objects in the scene, such as pets or people. Our dataset also lacks of an egocentric virtual human model, particularly the hand and body. The human model will aid in the development of hand tracking, human motion estimation, digital twin, and other related applications. The scene variety can also be improved.

4. Continuous video stereo matching

We now introduce our video stereo matching framework. Our key idea is to leverage the temporal redundancy of video within a RAFT-style cost aggregation scheme, where disparities are iteratively looked-up and refined by a GRU.

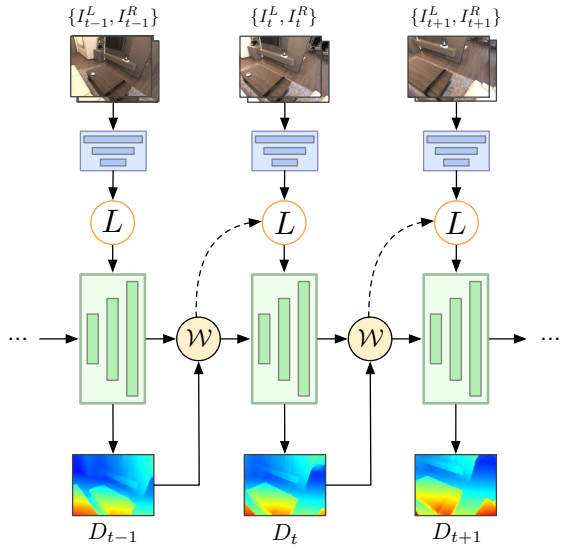

Figure 2. Real-time video stereo matching framework. We unroll an iterative cost aggregation network into temporal dimension, which enables us to distribute and reuse the aggregated feature over time and leads to substantial reduction in computation without sacrificing accuracy. In this example, we perform a single GRU iteration per frame. "L" denotes the disparity look-up.

Instead of running multiple iterations per frame in a frame-by-frame manner, we warp the previous frame disparity hypothesis to current frame to warm-start the GRU. As illustrated in Fig 2, this strategy significantly reduces GRU iterations per frame.

For time step t with new stereo image frames inputs $\{I^L_t, I^R_t\}$, their relevant world-to-camera pose inputs $\{P^L_t, P^R_t\}$, their camera intrinsic parameters $\{K^L, K^R\}$ and stereo baseline B, our method estimate disparity D_t for the left stereo image. We first extract matching feature and contextual feature from input images (Sec. 4.1). We then warp previous disparity estimation and previous hidden state into current camera frame (Sec. 4.3). Based on the warped disparity, we perform disparity lookup and compute matching cost on-the-fly. The matching cost is used along with current contextual feature in a recurrent network to estimate disparity for current time frame (Sec. 4.4)). We train this network in supervised manner (Sec. 4.5).

4.1. Feature Extraction

For any time step t with stereo image inputs $\{I^L_t, I^R_t\}$, we firstly extract matching feature $\{F^L_t, F^R_t\}$ and context feature $\{C^L_t, C^R_t\}$ using a shared-weight feature extraction network $\phi_f : I \to \{F, C\}$. We adopt the feature extraction network from [18] for fair comparison.

4.2. State Initialization

If the given stereo inputs on time step t is the very first frame, we perform a disparity and hidden state initialization. We use a small network ϕ_0 to estimate an initial disparity taking current left context feature as input. We initialize hidden state as all zeros. Pose of previous frame is set to be identical to current frame.

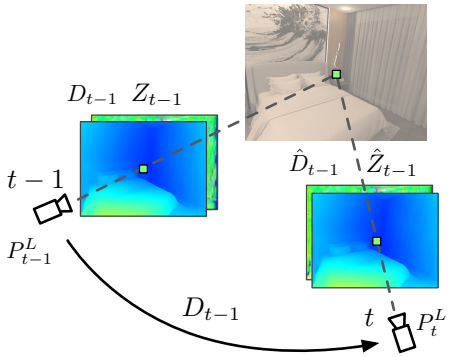

Figure 3. Temporal Warping. We reuse previous disparity estimation and hidden state feature by warping them to current frame based on relative camera pose.

4.3. Temporal Warping

We warp previous disparity estimation D_{t-1} and GRU hidden state Z_{t-1} into current camera frame, termed as $\hat{D}_{t-1}, \hat{Z}_{t-1}$ accordingly. As illustrated in Fig 3, this warping address the view point change caused by camera movement based on multi-view geometry. Specifically, we compute a transformation matrix $T^{geo}_{t-1,t}$ that maps stereo pixels (u, v, d) from previous left camera coordinate to current left camera coordinate.

$$T^{geo}_{t-1,t} = Q[RT]_{t-1,t}Q^{-1}, \tag{1}$$

where $[RT]_{t-1,t}$ is the relative camera pose and Q is the stereo transformation from stereo coordinates (u, v, d) to camera coordinates (x, y, z).

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & 1/b & 0 \end{bmatrix} \tag{2}$$

We warp previous disparity and hidden state to current frame using this transformation $T^{geo}_{t-1,t}$, resulting in $\hat{D}^{geo}_{t-1}, \hat{Z}^{geo}_{t-1}$ respectively. To handle occlusion during forward warping, we use Softmax Splatting [22] with weight proportional to current disparity, so that when multiple locations from previous view are mapped to the same location at current view, the nearest one takes priority. In case of disocclusion (holes), all missing values are assigned zero. We implement forward warping as a non-parametric layer in our network that is differentiable w.r.t. hidden state Z_{t-1}.

4.4. Disparity look-up and cost aggregation

Upon each new input frame at timestamp t, the cost aggregation GRU takes as initial input the warped hidden state

of previous frame $\hat{Z}_{t-1}$ and warped disparity $\hat{D}_{t-1}$, and performs iterative cost look-ups and disparity updates conditioned on context features. Same as RAFT-Stereo [18], we use feature correlation as cost metric to compute photometric matching cost $M_t \in \mathbb{R}^{h \times w \times K}$ as

$$M_t(u, v, d) = F_t^L(u, v) \cdot F_t^R(u + d, v). \quad (3)$$

Contrast to RAFT-Stereo [18], we are able to significantly reduce the number of GRU iterations. Even with a single GRU iteration per frame, our method performs comparably to RAFT-Stereo [18] using 20 GRU iterations.

4.5. Supervised Training

We train our network in a supervised manner. Following common practice in stereo matching methods [13, 18], we use the l_1 distance between estimated disparity and final disparity as disparity loss $\mathcal{L}_d$.

$$\mathcal{L}_d = ||D - D_{gt}||_1 \quad (4)$$

Unlike RAFT-Stereo [18] that applies an exponential loss weight over iteration steps, we treat the disparity output of each time step equally important and do not weigh down outputs on early time steps.

5. Experiments

In this section, we demonstrate the performance of our approach with a comprehensive set of experiments. Below, we describe the datasets and benchmarks, the implementation details, presentation and analysis of our results.

5.1. Datasets

We conduct extensive experiments on our proposed XR-Stereo dataset and also verify the performance of our method on the real-world KITTI VO dataset [11].

XR-Stereo dataset is our newly proposed synthetic indoor video stereo dataset. It consists the rendering of 14 virtual indoor scenes, which forms more than 60k photo-realistic stereo image pairs of indoor scenario. We use 640×480 resolution and 30Hz video frame rate. We split the dataset into 2 validation scenes and 2 testing scenes, and the rest are used for training.

KITTI Visual Odometry (VO) dataset [11] consists of image sequences acquired from a car driving in urban scenarios, captured with a calibrated stereo camera system and a high-precision GPS/IMU localization system. It cover a wide range of scenarios, such as residential areas, highways, and urban centers. It provides synchronized RGB stereo images and LiDAR measurements, IMU measurements, and GPS localization data. The dataset is widely used for evaluating visual odometry, stereo matching, and SLAM algorithms. We train our model from scratch and split the dataset into 20 training videos and 2 test videos.

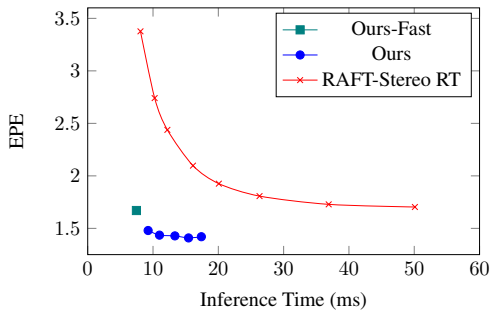

Figure 4. **XR-Stereo dataset.** Inference time and accuracy comparison between our method and our single-frame baseline RAFT-Stereo [18] real-time model. Our model outperform RAFT-Stereo real-time model on both accuracy and inference speed.

5.2. Metrics

We use standard evaluation metrics to assess our results. These metrics include the average end-point error (EPE), as well as the percentage of pixels with disparity error greater than 1 pixel (D1), 3 pixels (D3), and 5 pixels (D5). Furthermore, we also compute the bad 1%, bad 3%, and bad 5% EPE on the KITTI VO dataset, which correspond to the top 1%, 3%, and 5% percentiles of EPE, respectively.

5.3. Implementation Details

We implement two variants of our method, a full model (Ours) and a fast model (Ours-Fast). For the full model, we include results from our model with 1, 2 and 5 GRU iterations per frame. The fast model is introduced to maximize inference speed, for which we run GRU once per frame and disable the temporal warping. The fast model differs from our full model in two ways: (a) we remove the temporal warping by assuming camera motion between consecutive frames is small and continuous (b) we half intermediate feature channels within the feature encoder. We train both full model and fast model using Adam optimizer for 250K iterations using batch size 8 and sequence length 16. We use eight NVIDIA V100 32G GPUs for training. For all evaluations, we use a PC with one NVIDIA RTX 3090 TI GPU. To deploy our trained model on HMD, we convert the fast model via ONNX to a device-friendly float-point format running on Qualcomm XR2 chip without quantization.

5.4. Comparison with Existing Methods

We first compare our method with related methods on our XR-Stereo dataset. All methods are trained on our dataset unless otherwise stated, and are grouped into single-frame (Single-frame, or S in short) or video stereo methods (Video). For single frame methods we mainly compare to our real-time baseline RAFT-Stereo [18]. For video stereo methods we only compare to DeepVideoMVS [10] as other methods [37, 38] did not release code. Results are listed

	Method	EPE	D1	D3	D5	FPS
Single-frame	HiTNet* [28]	4.51	36.4	24.1	11.34	19.48
	GA-Net-deep [36]	2.11	22.56	11.05	8.46	1.24
	GWC-Net [13]	1.87	20.45	10.01	7.99	8.10
	PCW-Net [26]	1.77	19.43	9.96	7.84	6.53
	ACV-Net [33]	1.69	19.31	8.94	6.57	13.89
	RAFT-Stereo [18] (RT @ 7 iters)	1.93	20.93	9.85	7.03	49.83
	RAFT-Stereo [18] (RT @ 20 iters)	1.70	18.18	8.37	6.03	19.97
Video	DeepVideoMVS*	2.20	20.2	9.52	7.10	13.22
	Ours-Fast	1.67	19.40	8.86	6.25	**134.05**
	Ours (1 iter)	1.48	18.64	8.55	5.95	108.09
	Ours (2 iters)	1.44	16.80	7.89	5.50	90.97
	Ours (5 iters)	**1.42**	**16.31**	**7.69**	**5.36**	57.46

Table 2. **XR-Stereo dataset.** Our method outperforms all existing methods in terms of both accuracy and runtime fps. For accuracy similar to RAFT-Stereo [18] real-time model (RT @ 20 iters), Our fast model achieved an impressive 134 fps on inference.

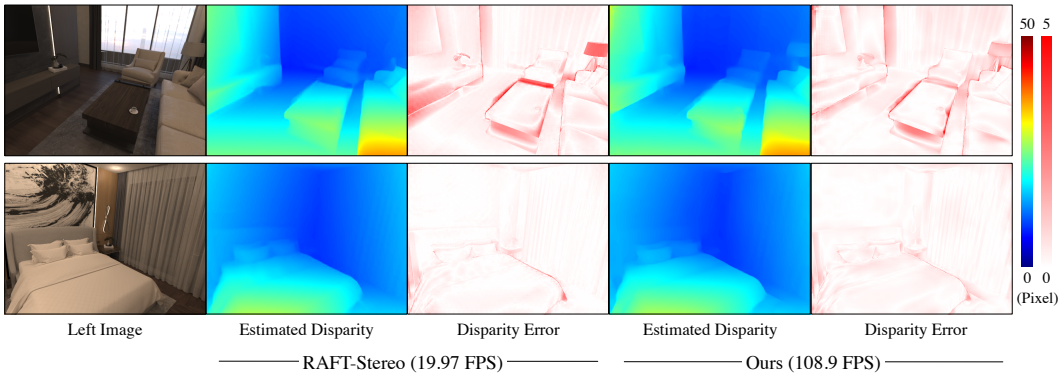

Figure 5. **XR-Stereo dataset.** Qualitative results

in Table 2. HiTNet [28] did not release training code, so we can only use its released pre-trained model for evaluation. Disparity from DeepVideoMVS [10] were obtained by $d = \frac{b \times f}{Z}$. DeepVideoMVS runs much slower than reported in original paper due to our dataset has 4X higher image resolution than ScanNet [8]. Our fast model, which lacks the temporal warping, already outperforms all existing methods in EPE and achieved an impressive 134 FPS inference speed. Our full model that runs a single GRU iteration maintains a high inference speed at 100+ FPS. By further adding GRU iterations, our method consistently outperformed all competing methods on all metrics.

5.5. Efficiency versus accuracy

Apart from the general inference fps provided in Table 5, we also specifically compared the runtime efficiency and accuracy of our model to the single-frame baseline RAFT-Stereo. For both models, we plot the runtime-EPE curve with varying number of GRU iterations. The results are shown in Figure 4. Our model achieves significantly better accuracy compared to our single-frame counterpart while operating within a fraction of its time.

5.6. Generalization Ability

We evaluate the generalization ability of our model on KITTI Visual Odometry (VO) dataset with low frame rate and sparse depth supervision. Results are shown in Table 3. Our model outperforms the single-frame baseline using significantly less GRU iterations, which proves its generalization ability to real-world applications.

5.7. Ablation study

We provide ablation experiments on the XR-Stereo dataset to evaluate the contribution of the proposed modules and also provide detailed analysis of our method.

Temporal warping In this ablation we evaluate the importance of correct geometric alignment by temporal warping. A baseline model is trained without warping and compared to our full model running one GRU iteration per frame. Results are show in Table 4. The performance of baseline model deteriorates as camera motion increases when the full model is robust to large motions. Similar observations are made in Figure 8, where the fast model is vulnerable to large motions due to lack of temporal warping.

Pose noise We analyze the robustness of our model against noisy camera pose. We manually add a random noise into to the input pose before using it in our temporal warping mod-

	Method	EPE	D1	D3	D5	Bad 1%	Bad 3%	Bad 5%
S	RAFT-Stereo [18] (RT @ 7 iters)	1.82	26.47	10.26	6.71	26.93	13.69	8.27
	RAFT-Stereo [18] (RT @ 20 iters)	1.77	25.76	9.79	6.39	27.01	13.30	7.85
Video	Ours-Fast	1.91	30.58	11.92	7.51	25.15	13.55	8.64
	Ours (1 iters)	1.84	30.33	11.14	6.93	24.87	12.82	7.94
	Ours (2 iters)	1.72	26.71	10.11	6.47	**24.71**	12.59	7.55
	Ours (5 iters)	**1.66**	**24.52**	**9.57**	**6.22**	24.87	**12.40**	**7.32**

Table 3. **KITTI VO dataset.** Performance of our method compared with RAFT-Stereo [18] in real-world scenario.

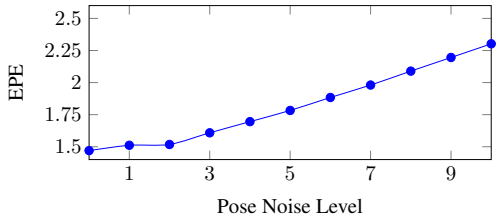

Figure 6. **XR-Stereo dataset.** EPE versus Pose Noise. Each noise level adds an increment of 0.3 degrees maximum rotation noise and 1mm maximum translation noise.

Temporal Warping	1X speed		6X speed	
	EPE	D1	EPE	D1
	1.70	18.29	3.18	32.05
✓	1.48	18.64	1.68	21.16

Table 4. **XR-Stereo dataset.** Contribution of temporal warping.

ules. Specifically, we add a random rotation and translation noise drawn from uniform distributions. Results are show in Figure 6, where each noise level adds an increment of 0.3 degrees maximum rotation noise and 1mm maximum translation noise. Our model is affected by extreme pose noise. However, since our method only requires relative pose between two consecutive frames, many SLAM pipelines falls within noise level 1. *E.g.* a modified ORB-SLAM3 [6] runs within 0.3 degree and 0.5mm inter-frame noise on our headset, with local bundle adjustment.

Movement speed Since our model is specifically designed to utilize the relation between consecutive frames, inter-frame motion can affect our model's performance. High speed motions reduce overlap between consecutive frames and can further introduce occlusion/disocclusion. We increase inter-frame motion intensity by skipping frames in a 30Hz input stereo video stream, simulating up to 20x original speed. Results are shown in Figure 8. Our full model is robust to high speed motions when the fast model (without warping) is vulnerable to such changes.

Performance curve on initialization We now analyze the performance curve on model initialization. Since our model rely on temporal iterative cost aggregation, in a fresh start, it require certain amount of frames to be processed to reach a stable performance. In Figure 7, we show the disparity accuracy curve as more frames are fed into our model (1 iter) upon initialization. The disparity accuracy of the very first few frames gradually improved along with more iterations of temporal cost aggregation been performed. The

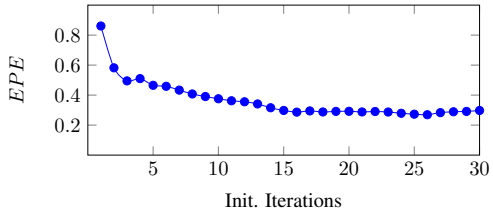

Figure 7. **XR-Stereo dataset.** EPE curve as our model progresses through an input video. Error converges after 15 stereo frames.

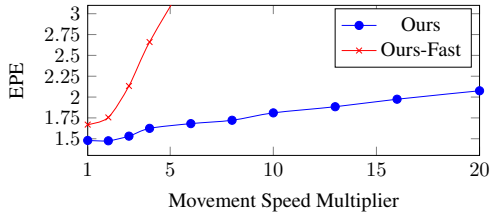

Figure 8. **XR-Stereo dataset.** Model robustness with respect to movement speed. Our fast model, which lacks temporal warping, can be largely affected by high movement speed. Our full model with temporal warping achieve reasonable performance even in extreme speed (20x).

performance of our model stabilized after around 15 frames, which roughly corresponds to a 0.5 second video duration.

5.8. Limitation

Our proposed approach, while effective in stereo matching scenarios with stereo video inputs, may face limitations in applications where such inputs are not available. Besides, it may not provide reliable and accurate disparity estimation at the very first stereo frame, thereby limiting its use in on-demand stereo matching applications.

6. Conclusion

We present two novel contributions to facilitate the development of XR, including an indoor XR video stereo dataset implemented in high-fidelity and a highly efficient real-time video stereo matching framework that can potentially run in real-time on low-power stand-alone VR/AR headsets. In the future, we would like to improve our dataset to enable the development of more XR algorithms, and extend our video stereo framework to sceneflow estimation.

References

[1] Antyanta Bangunharcana, Jae Won Cho, Seokju Lee, In So Kweon, Kyung-Soo Kim, and Soohyun Kim. Correlate-and-excite: Real-time stereo matching via guided cost volume excitation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3542–3548. IEEE, 2021. 3

[2] Wei Bao, Wei Wang, Yuhua Xu, Yulan Guo, Siyu Hong, and Xiaohu Zhang. Instereo2k: a large real dataset for stereo matching in indoor scenes. *Science China Information Sciences*, 63:1–11, 2020. 2

[3] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11*, pages 29–43. Springer, 2010. 3

[4] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011. 3

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 2, 4

[6] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 8

[7] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 4

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 7

[9] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4384–4393, 2019. 3

[10] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15324–15333, 2021. 3, 6, 7

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 4, 6

[12] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020. 3

[13] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019. 3, 6, 7

[14] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005. 3

[15] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2702–2719, 2019. 2, 4

[16] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 573–590, 2018. 3

[17] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272, 2022. 3

[18] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021. 3, 5, 6, 7, 8

[19] Yamin Mao, Zhihua Liu, Weiming Li, Yuchao Dai, Qiang Wang, Yun-Tae Kim, and Hong-Seok Lee. Uasnet: Uncertainty adaptive sampling network for deep stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6311–6319, 2021. 3

[20] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 2, 4

[21] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134. 4

[22] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 5

[23] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition: 36th German Confer-*

ence, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36, pages 31–42. Springer, 2014. 2, 4

[24] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47:7–42, 2002. 2, 3

[25] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017. 2, 3, 4

[26] Zhelun Shen, Yuchao Dai, Xibin Song, Zhibo Rao, Dingfu Zhou, and Liangjun Zhang. Pcw-net: Pyramid combination and warping cost volume for stereo matching. In *European Conference on Computer Vision*, pages 280–297. Springer, 2022. 7

[27] ANNA SKOROBOGATOVA. Real-time global illumination in unreal engine 5. 4

[28] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14372, 2021. 3, 7

[29] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8942–8952, 2021. 3

[30] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large synthetic indoor robotics stereo dataset for disparity and surface normal estimation. *arXiv preprint arXiv:1912.09678*, 6, 2019. 2, 4

[31] 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. 2020. 2, 3, 4

[32] Bin Xu, Yuhua Xu, Xiaoli Yang, Wei Jia, and Yulan Guo. Bilateral grid learning for stereo matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12497–12506, 2021. 3

[33] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. Attention concatenation volume for accurate and efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12981–12990, 2022. 7

[34] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020. 3

[35] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. 3

[36] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. 7

[37] Youmin Zhang, Matteo Poggi, and Stefano Mattoccia. Temporalstereo: Efficient spatial-temporal stereo matching network. *arXiv preprint arXiv:2211.13755*, 2022. 3, 6

[38] Yiran Zhong, Hongdong Li, and Yuchao Dai. Open-world stereo video matching with deep rnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–116, 2018. 3, 6