DynamicStereo: Consistent Dynamic Depth from Stereo Videos

Nikita Karaev[1,2] Ignacio Rocco[1] Benjamin Graham[1] Natalia Neverova[1]
Andrea Vedaldi[1] Christian Rupprecht[2]

[1] Meta AI [2] Visual Geometry Group, University of Oxford

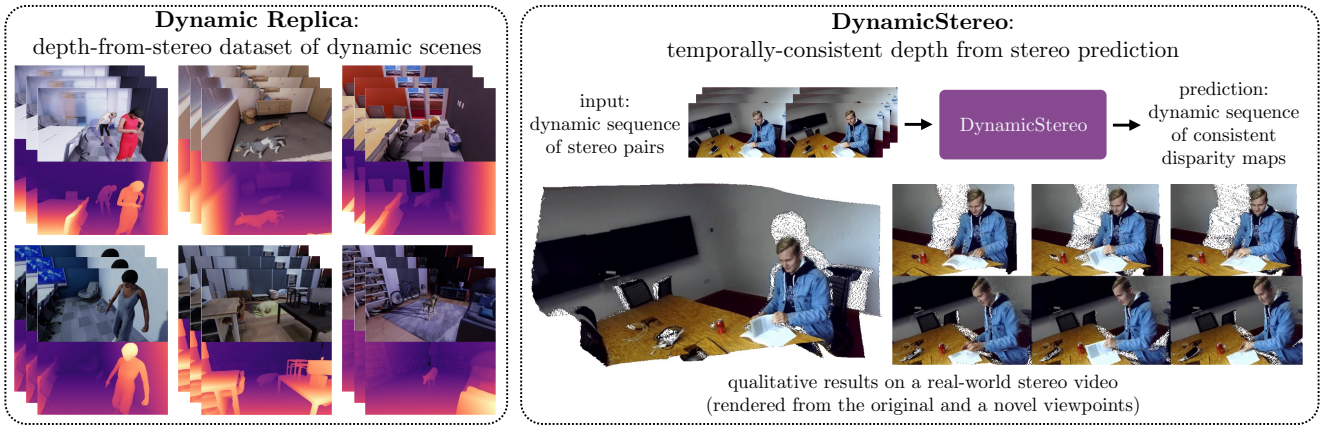


Figure 1. **Dynamic Replica and DynamicStereo.** In this work, we (a) introduce a new synthetic stereo video dataset (Dynamic Replica) to train and benchmark temporally consistent disparity estimators for dynamic scenes with people and animals, and (b) propose a method (DynamicStereo) exploiting recent advances in transformer architectures that performs efficient stereo matching at the level of videos.

Abstract

We consider the problem of reconstructing a dynamic scene observed from a stereo camera. Most existing methods for depth from stereo treat different stereo frames independently, leading to temporally inconsistent depth predictions. Temporal consistency is especially important for immersive AR or VR scenarios, where flickering greatly diminishes the user experience. We propose DynamicStereo, a novel transformer-based architecture to estimate disparity for stereo videos. The network learns to pool information from neighboring frames to improve the temporal consistency of its predictions. Our architecture is designed to process stereo videos efficiently through divided attention layers. We also introduce Dynamic Replica, a new benchmark dataset containing synthetic videos of people and animals in scanned environments, which provides complementary training and evaluation data for dynamic stereo closer to real applications than existing datasets. Training with this dataset further improves the quality of predictions of our proposed DynamicStereo as well as prior methods. Finally, it acts as a benchmark for consistent stereo methods. Project page: https://dynamic-stereo.github.io/

1. Introduction

Estimating depth from stereo is a fundamental computer vision problem, with applications in 3D reconstruction, robot navigation, and human motion capture, among others. With the advent of consumer devices featuring multiple cameras, such as AR glasses and smartphones, stereo can simplify the 3D reconstruction of everyday scenes, extracting them as content to be experienced in virtual or mixed reality, or for mixed reality pass-through.

Depth from stereo takes as input two images capturing the same scene from different viewpoints. It then finds pairs of matching points, a problem known as *disparity estimation*. Since the two cameras are calibrated, the matched points can be projected into 3D using triangulation. While this process is robust, it is suboptimal when applied to video data, as it can only reconstruct stereo frames individually, ignoring the fact that the observations infer properties of the *same underlying objects* over time. Even if the camera moves or the scene deforms non-rigidly, the instantaneous 3D reconstructions are highly correlated and disregarding this fact can result in inconsistencies.

In this paper, we thus consider the problem of *dynamic*

depth from stereo to improve the temporal consistency of stereo reconstruction from video data.

Traditional approaches to stereo compute the matching costs between local image patches, aggregating those in an objective function, and optimizing the latter together with a regularization term to infer disparities. Examples of such approaches include max-flow [35] and graph-cut [15]. More recently, stereo methods have used deep networks learned from a large number of image pairs annotated with ground-truth disparities [12, 14, 18, 23]. They usually follow an approach similar to the traditional methods, but using deep CNN features for computing the matching costs, and replacing the per-image optimization by a pre-trained regression deep network, which processes the cost volume and outputs the estimated disparities.

In the video setting, matching quality can potentially be improved by looking for matches across space *and* time. For instance, points occluded in one camera at a given point in time may be visible from both cameras at other times.

Transformer architectures have shown that attention can be a powerful and flexible method for pooling information over a range of contexts [6, 8, 9, 42]. Our *DynamicStereo* model incorporates self- and cross-attention to extract relevant information across space, time and stereo pairs. Our architecture relies on divided attention [3] to allow efficient processing of this high-dimensional space.

As a learning-based approach, we wish to learn priors from data representative of real-life 3D dynamic reconstruction applications, where videos depict people or animals moving and interacting with objects. There are several synthetic video datasets [10,27,41] commonly used for training stereo and optical flow methods, but they contain abstract scenes with several layers of moving objects that share little resemblance to real-life. More realistic stereo datasets also exist [45, 49], but they either do not contain video sequences or are focused on static scenes. Given these limitations, as an additional contribution we propose a new synthetic stereo dataset showing moving human and animal characters inside realistic physical spaces from the Replica dataset [40]. We call this new dataset *Dynamic Replica* (DR), and we use it for learning dynamic stereo matches. DR contains 524 videos of virtual humans and animals embedded in realistic digital scans of physical environments (see Tab. 1 and Fig. 2). We show that DR can significantly boost the quality of dynamic stereo methods compared to training them only on existing depth-from-stereo datasets.

To summarise, we make **three contributions**. (1) We introduce *DynamicStereo*, a transformer-based architecture that improves dynamic depth from stereo by jointly processing stereo videos. (2) We release *Dynamic Replica*, a new benchmark dataset for learning and evaluating models for dynamic depth from stereo. (3) We demonstrate state-of-the-art dynamic stereo results in a variety of benchmarks.

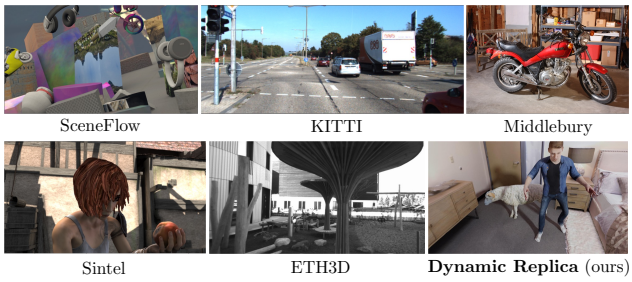


Figure 2. **Example frames from depth-from-stereo datasets.** We visually compare current datasets to *Dynamic Replica*, which contains renderings of every-day scenes with people and animals and differs from existing datasets in size, realism, and content.

2. Related work

Depth from stereo. Stereo matching is a classic problem in computer vision. The traditional way of solving it is to compute local matching costs between image patches and then either perform local aggregation [4, 13, 46, 56], or a global optimization based on an energy function featuring data and regularization terms [15, 35].

More recently, deep learning has become the dominant approach to stereo matching. Zbontar and LeCun [52] proposed to use a CNN to compute the matching cost between image patches. Then Mayer *et al*. [27] introduced the first fully learning-based approach to stereo estimation. Inspired by traditional stereo algorithms, the next line of work [7, 12, 14, 51, 53, 54] applied 3D convolutions in their fully learning-based approaches. They construct a dense 3D cost volume between the left and the right 2D feature-maps before filtering it with a 3D-CNN. These approaches often fail to generalize to data that they were not trained on.

More recent works focused on improving the computational efficiency of stereo estimation [1, 22, 23, 43]. Inspired by RAFT [44] which constructs a 4D cost volume between all pairs of pixels for optical flow, RAFT-Stereo [23] restricts this volume to 3D by collapsing it to the epipolar line. Similarly to RAFT, it iteratively updates the prediction at high resolution allowing to maintain global context and recover details. These updates are performed by a GRU [2] that operates at multiple resolutions.

CRE-Stereo [18] proposes to gradually increase resolution during iterative updates which simplifies the architecture. However, the lack of information propagation from lower to higher resolution may result in a loss of context. Our architecture also refines disparity in a coarse-to-fine manner, but unlike CRE-Stereo, we fuse low-resolution and high-resolution features together and propagate them to the final prediction. This allows high-resolution layers to use these features and keep track of previous states.

Some recent works incorporate attention [47]. Stereo Transformer [19] replaces cost volume with dense pixel

Dataset property	MPI Sintel [5]	KITTI [28]	SceneFlow [27]	Falling Things [45]	TartanAir [50]	Dynamic Replica (ours)
#Training frames	1 064	194+200	34 801	60 200	296 000	145 200
#Test frames	564	195+200	4 248	0	0	6 000 + 18 000
#Training sequences	25	194+200	2 256	(330)	1037	484
Resolution	1024×436	1242×375	960×540	960×540	640×480	1280×720
Disparity/Depth	✓	sparse	✓	✓	✓	✓
Optical flow	✓	(sparse)	✓	✗	✓	✓
Segmentation	✓	✗	✓	✓	✓	✓
Non-rigid objects	✓	✗	(✓)	✗	(✗)	✓
Realism	(✓)	✓	✗	(✓)	(✓)	(✓)

Table 1. **Comparison of depth-from-stereo datasets.** *Dynamic Replica* is larger than previous datasets in both resolution and number of frames. A main aspect is that it contains non-rigid objects such as animals and people, which is not available at scale in prior datasets.

matching using attention. LoFTR [42] and SuperGlue [36] use combinations of self and cross-attention layers for sparse feature matching. Inspired by these methods, CRE-Stereo [18] uses self and cross-attention to improve convolutional features. These works focus on disparity estimation for individual frames. As we predict disparity for videos, we apply attention across time, space, and stereo frames.

Dynamic video depth. Monocular depth estimators like MiDaS [30, 31] attempt to estimate depth from a single image. Due to the ambiguity of this task, they tend to be far less accurate than methods that use stereo. Recent works have proposed to extend these methods to use monocular *videos* instead, relying on motion parallax in static areas [21] or fusing information extracted from all the frames of a video, in an off-line fashion.

Consistent Video Depth (CVD) [25] assumes that objects move almost rigidly across neighboring frames. Robust CVD (RCVD) [16] excludes dynamic objects from the optimization objective. Dynamic VD (DVD) [55] explicitly models motion of non-rigid objects with a scene-flow network. All these methods require fine-tuning a monocular depth predictor like MiDAS on a specific video, a process that must be carried out from scratch for each new video.

Dynamic depth from stereo. The work of Li *et al.* (CODD) [20] is the closest to ours. It is based on three different networks: stereo, motion, and fusion, that are trained separately. During inference, consistency is achieved by extracting information from the memory state that is updated after each iteration. It is an online approach that assumes no access to future frames and thus does not allow global optimization over the whole sequence. Our network does not need a memory state and learns consistency from data. It can be applied in both online and offline settings.

Datasets for learning depth from stereo. Training data is an important factor for learning-based stereo algorithms. It is challenging to collect real data for this task because disparity is extremely difficult to annotate. [28, 48] Active sensors such as time-of-flight cameras can provide ground truth depth. KITTI [28] shows that converting such ground truth depth to pixel-accurate annotations is still challenging for dynamic objects due to the low frame rate and imprecise estimations. Synthetic datasets [17, 32, 33] can simplify the data collection process. SceneFlow [27] is the first large-scale synthetic dataset for disparity and optical flow estimation. It allowed training fully learning-based methods for these tasks. SceneFlow is less realistic compared to MPI Sintel [5], a much smaller dataset with optical flow and disparity annotations. Falling Things [45] is realistic and relatively large but does not contain non-rigid objects. TartanAir [49] is a realistic SLAM dataset with ground-truth stereo information but only few non-rigid objects.

To the best of our knowledge, we are the first to introduce a large-scale semi-realistic synthetic dataset with a focus on non-rigid objects for disparity estimation.

3. Method

We first formalize the *dynamic depth from stereo* problem. Given a sequence $S = [(I_t^L, I_t^R)]_{1 \le t \le T} \in \mathbb{R}^{2\times 3\times H\times W}$ of T rectified stereo frames, the task is to predict a sequence of disparity maps $\hat{D} = [D_t]_{1 \le t \le T} \in \mathbb{R}^{H\times W}$ aligned with the left frames I_t^L. While most disparity estimation methods operate on single frames $\hat{D}_t = \Phi(S_t)$, here we learn a model that operates on sequences of length T, constructing a function $\hat{D} = \Phi(S)$. This has the advantage that the model can fuse information along time and thus improve its temporal consistency.

The challenge is to design an architecture that can pass information across such a large volume of data efficiently. We achieve this via an encoder-decoder design (Fig. 3), building on prior work [11, 18, 34]. The encoder extracts features independently from all the frames and obtains a multi-scale representation of their content. The decoder then matches progressively more detailed features to recover the disparities from coarse to fine. Low-resolution matches easily spot large displacements of large image regions, capturing the rough structure of the disparity map, whereas high-resolution matches recover the details.

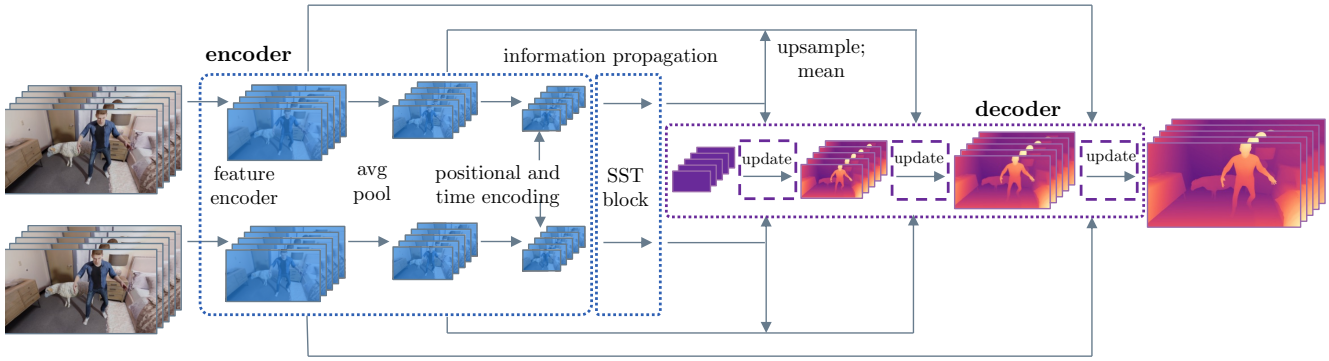


Figure 3. **Proposed architecture of DynamicStereo.** Our method consists of three main components. A convolutional encoder produces feature maps at three scales. As data spans time, space and stereo views, the SST-Block operating at the smallest resolution, ensures information exchange across all dimensions. Finally, the prediction is generated by a decoder composed of an iterative update function g.

For efficiency, matches are always carried out along epipolar lines. The task of exchanging information across space, view and time is delegated to two mechanisms. First, the encoder terminates in a transformer network that updates the lowest-resolution feature by running attention across these three dimensions in turn. This is efficient because it is done only for the lowest-resolution features. Second, the decoder further propagates information as it matches features to recover the disparities. The decoder itself consists of update blocks (Fig. 4a) that use both space and time information to gradually refine the disparity.

3.1. Encoder

The encoder starts by **extracting features** from every frame I_t^v, $v \in \{L, R\}$, $t \in \{1, \dots, T\}$ of the stereo video independently by applying the same CNN F to them. As it is typical for CNN backbones, the resolution of the output feature map is lower than the resolution of the input image. Features are extracted with minimum stride $k = 4$ and are further down-sampled with average pooling to obtain features at $1/8$ and $1/16$ of the original resolution. Overall, this results in the feature maps $\phi(I_t^v)_k \in \mathbb{R}^{d \times \frac{H}{k} \times \frac{W}{k}}$, $k \in \{4, 8, 16\}$. We will use the symbol $\phi_k \in \mathbb{R}^{T \times 2 \times d \times \frac{H}{k} \times \frac{W}{k}}$ to refer to the combined five-dimensional (temporal, stereo and spatial) feature volume at resolution k.

Information Propagation. The backbone F processes frames independently, so we require a different mechanism to exchange of information between left and right views v and different timestamps t. We do so by passing the features to a transformer network that uses self and cross-attention. Ideally, attention should compare feature maps across views, time, and space. However, it is computationally demanding to apply attention jointly even with linear [42] space and stereo attention. We thus rely on divided attention [3] to attend these three dimensions individually. We call this a Space-Stereo-Time attention block (SST, Fig. 4b) and we repeat it four times. Since attention remains computationally expensive, we only apply SST to the lowest-resolution feature map ϕ_{16} only.

3.2. Decoder

The output of the encoder is a multi-resolution feature volume, where the lowest resolution features incorporate information across time, view and space dimensions due to the SST block described above. The task of the decoder is to convert this feature volume into the final disparity values.

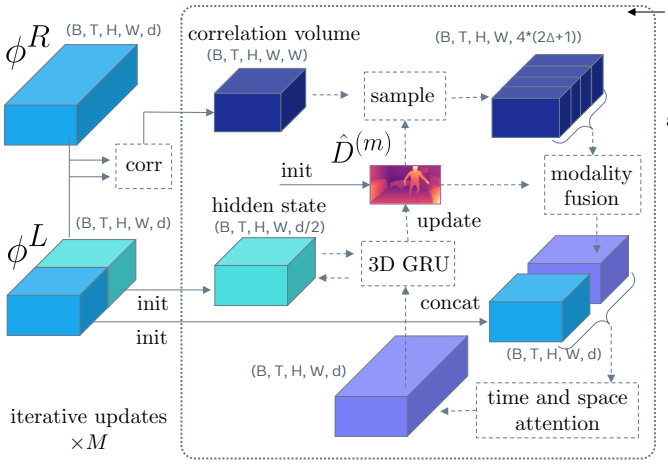
The decoder is based on three ideas: (1) Disparities are updated from coarse to fine [18], using features of increasing resolution to update earlier estimates. (2) At each resolution, a feature correlation volume is computed and correspondences are refined iteratively [44] with reference to this volume. (3) Similar to the SST attention blocks in the encoder, information is exchanged between the three dimensions (space, view and time) throughout decoding. We describe next each component in detail.

Iterative correspondence updates. Our model produces a sequence of progressively more accurate disparity estimates $\hat{D} = \hat{D}^{(m)} \in \mathbb{R}^{\frac{H}{k} \times \frac{W}{k}}$, $0 \leq m \leq M$ starting from $\hat{D}^{(0)} = 0$ and then applying M times the update rule:

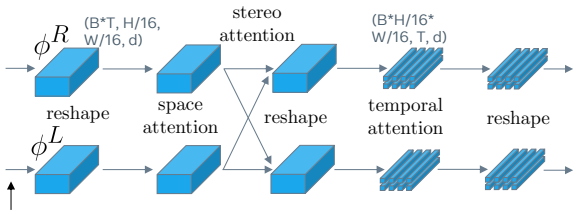
$$\hat{D}^{(m+1)} = \hat{D}^{(m)} + g(\hat{D}^{(m)}, \phi). \quad (1)$$

We apply the rule $\frac{M}{4}$ times to obtain disparities $\hat{D}^{(m)}$ at the coarser resolution $(\frac{H}{16}, \frac{W}{16})$. We then upsample $\hat{D}^{(\frac{M}{4})}$ to resolution $(\frac{H}{8}, \frac{W}{8})$, apply the update rule $\frac{M}{4}$ more times to obtain $\hat{D}^{(\frac{M}{2})}$, upsample the disparity again, and apply the update $\frac{M}{2}$ more times to obtain $\hat{D}^{(M)}$ at resolution $(\frac{H}{4}, \frac{W}{4})$. Finally, we upsample the predicted disparity $\hat{D}^{(M)}$ to (H, W). Different versions of the function g are trained for each of the three resolutions and upsampling uses the same mechanism as in RAFT.

The detailed structure of the update network g is illustrated in Fig. 4a and described below.



(a) **Update block** g. We use the current disparity estimate $\hat{D}^{(m)}$ to sample the corresponding point and its epipolar neighbourhood from the correlation volume C at different resolutions. The resulting correlation estimate $\hat{C}^{(m)}$ is fused with the disparity $\hat{D}^{(m)}$ and with a part of feature map ϕ^L. We then apply space-time attention before passing the fused output to the 3D GRU that updates the disparity.



(b) **SST-Block**: combines all dimensions of the input spatial, stereo and temporal attention layers. Divided attention greatly reduces computational complexity of the network.

Figure 4. **The proposed architecture of (a) update block g and (b) SST-Block.**

Correlation volume. Similar to classical disparity estimation methods, g starts by computing the correlation volumes between left and right features $C_{t,s,k} = \text{corr}(\phi^L_{t,s,k}, \phi^R_{t,s,k}) \in \mathbb{R}^{\frac{H}{sk} \times \frac{W}{sk} \times \frac{W}{sk}}$ for each feature resolution k and at different scales $s \in \{1, 2, 4, 8\}$. The correlation is computed along epipolar lines, which correspond to image rows as the input images are rectified. Each element (h, w, w') of $C_{t,s,k}$ is the inner product between feature vectors of $\phi^L_{t,s,k}$ and $\phi^R_{t,s,k}$:

$$C_{t,s,k}(h, w, w') = \frac{1}{\sqrt{d}} \langle \phi^L_{t,s,k}(h, w), \phi^R_{t,s,k}(h, w') \rangle. \quad (2)$$

Thus, $C_{t,s,k}(h, w, w')$ is proportional to how well the left image point (h, w) matches the right image point (h, w'). As the correlation volume does not depend on the update iteration m, it is computed only once.

Correlation lookup. The function g updates the disparity at location (h, w) by observing the correlation volume in a local neighbourhood centered on the current disparity value $\hat{D}^{(m)}_t(h, w)$. The necessary samples are collected and stacked as feature channels of the tensor

$$\hat{C}^{(m)}_{t,k}(h, w) = \underset{s,\delta}{\text{cat}} \left[C_{t,s,k} \left(\frac{h}{s}, \frac{w}{s}, \frac{w + \hat{D}^{(m)}_t(h, w)}{s} + \delta \right) \right]. \quad (3)$$

The current correlation estimate $\hat{C}^{(m)}_{t,k} \in \mathbb{R}^{\frac{H}{k} \times \frac{W}{k} \times 4(2\Delta+1)}$ captures the correlation volume across all four scales and in a neighborhood $\delta \in \{-\Delta, \ldots, \Delta\}$ for additional context, with cat representing the concatenation operation.

Modality fusion. The estimated correlation, disparity, and feature maps need to be combined to provide the update block with enough information to update the disparity. Using a late fusion scheme, we encode correlation $\hat{C}^{(m)}_{t,k}$ and disparity $\hat{D}^{(m)}_t$ separately, before concatenating them with the feature map of the left frame $\phi^L_{t,k}$, as it is the reference frame. To incorporate temporal and spatial information, we apply self-attention across time (T) and space $(\frac{H}{k}, \frac{W}{k})$ to the output of the modality fusion step. For efficiency, we do it only for $k = 16$. Architecture details can be found in Fig. 4 and the supplementary material.

3D CNN-based GRU. The update function g is implemented using a 3D convolutional GRU.

At each step m, the GRU takes as input the fused features and a hidden state that is initialized with the reference feature map ϕ^L. All internal operations of the GRU are implemented as separable 3D convolutions across space and time to propagate temporal and spatial information. The output of each iteration is an update to the current disparity estimate as in eq. (1). Each subsequent iteration is preceded by correlation lookup and modality fusion.

3.3. Training

Due to its iterative nature, the proposed model generates M predictions for every timestamp. During training, we supervise the network over the full sequence of predictions $\hat{D}^{(m)}_t$, with exponentially increasing weights towards the final estimate at step M as follows:

$$\mathcal{L}(\hat{D}, D) = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma^{M-m} \| \hat{D}^{(m)}_t - D_t \|, \quad (4)$$

where $\gamma = 0.9$ and D is the ground truth disparity sequence. Lower resolution disparity estimates are up-sampled to ground truth resolution.

3.4. Inference

The model is trained on stereo sequences with a fixed length of T. To increase the temporal coherence at test time,

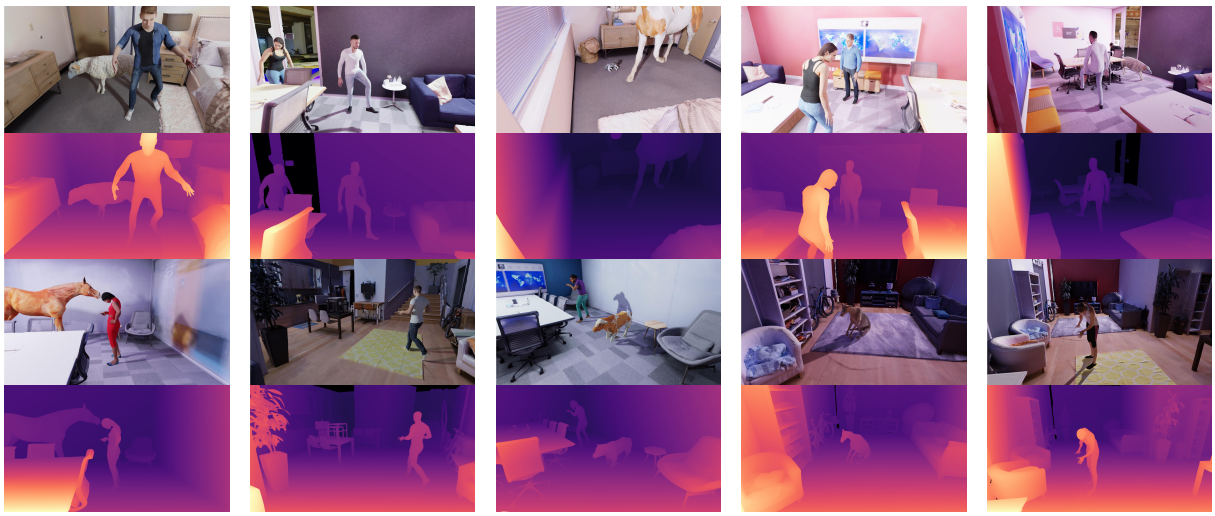


Table 2. **Dynamic Replica.** Example video frames and disparity maps from our proposed dataset. The proposed dataset contains diverse indoor scenes filled with animated people and animals to provide a benchmark for disparity estimation in close-to-real environments.

we apply it to videos of arbitrary length using a sliding window approach with overlap, where we discard the predictions in overlapping time steps. For details see supp. mat. Videos shorter than T time steps can be used by repeating the first or the last frame up to the minimal length T. This corresponds to a static scene without moving camera and is well within the training distribution of the model.

4. Dynamic Replica: dynamic stereo dataset

Training temporally consistent models requires a dataset with stereo videos and dense ground-truth annotations. While the SceneFlow dataset [26] fulfils both criteria, it is comprised of very short sequences of randomly textured objects moving. In this paper, we introduce a more realistic synthetic dataset and benchmark of animated humans and animals in every-day scenes: *Dynamic Replica*.

The dataset consists of 524 videos of synthetic humans and animals performing actions in virtual environments (see Tab. 2). Training and validation videos are 10 seconds long and each contain 300 frames. There are 484 training and 20 validation videos. Our test split consists of 20 videos of length 30 seconds to benchmark models on longer videos.

Videos are rendered at a resolution of 1280×720. This is close to the resolution of modern screens and higher than the resolution of such popular stereo datasets as Sintel (1024×536) and SceneFlow (960×540).

The dataset is based on Facebook Replica [40] reconstructions of indoor spaces. We take 375 3D scans of humans from the RenderPeople[1] dataset and animate them using motion capture sequences from real humans scans. We use artist-created animated 3D models of animals from 13 categories (chimp, dog, horse, sheep, etc.). We use different environments, scans, textures and motions for training, validation and test splits to evaluate generalization to unseen environments and dynamic objects.

We randomize camera baselines in our training subset to ensure generalization across different stereo setups. Baselines are sampled uniformly between 4cm and 30cm.

For each scene we generate a camera trajectory imitating a person filming the scene with their mobile phone or AR glasses. These virtual cameras have smooth trajectories and are located approximately at 1.25m above the ground.

All samples in the dataset contain ground-truth depth maps, optical flow, foreground / background segmentation masks and camera parameters for both stereo views.

5. Experiments

We structure the experiments as follows. First, we evaluate the *Dynamic Replica* dataset by comparing generalization performance of prior models trained on other datasets and on *Dynamic Replica*. Then we evaluate our model and compare it to the state of the art in temporal consistency. Finally, we ablate design choices in the model architecture and verify the importance of its individual components.

Implementation Details. We implement *DynamicStereo* in PyTorch [29] and train on 8 NVIDIA TESLA Volta V100 32GB GPUs. The SF version is trained for 70k iterations with a batch size 8. We train the DR+SF version for 120k iterations which takes about 4 days.

Both models are trained on random 384×512 crops of sequences of length $T = 5$ and evaluated in the original resolution with $T = 20$ and overlapping windows of size 10. During training, we use the AdamW optimizer [24] and set

[1] http://renderpeople.com/

Mtd	Data	KITTI	Middlebury			ETH3D	Sintel Stereo		DR
			full	half	quarter		Clean	Final	full
[23]	DR	7.25	19.51	13.13	13.86	3.78	7.36	13.04	2.90
[23]	SF	**5.55**	17.76	12.80	9.64	3.05	**5.89**	9.20	4.01
[23]	(DR+SF)/2	5.63	**15.08**	**10.36**	**9.24**	**3.02**	5.96	**9.12**	**2.20**
[18]	DR	6.04	31.94	23.82	15.93	3.94	15.03	19.35	4.13
[18]	SF	6.22	23.95	15.64	10.50	3.95	7.51	11.01	6.59
[18]	(DR+SF)/2	**5.35**	**22.02**	**13.69**	**8.96**	**3.53**	**6.96**	**9.98**	**3.49**

Table 3. **Dynamic Replica generalization – non-temporal disparity estimation.** SF - SceneFlow [27], DR - *Dynamic Replica* (ours). For (DR+SF)/2, we replace half of SF with samples from DR. The performance improves over pure SF training, showing that DR is valuable for generalization, compared to training only with SF data. Errors are the percent of pixels with end-point-error greater than the specified threshold. We average across 3 runs with different seeds and use the standard evaluation thresholds: 3px for KITTI 2015 [28] and Sintel Stereo [5], 2px for Middlebury [37], 1px for ETH3D [38] and DR.

the number of iterative updates $M = 20$. We train with one-cycle learning rate schedule [39] with a maximum learning rate $3 \cdot 10^{-4}$. We set the lookup neighborhood $\Delta = 4$ (see Sec. 3.2). For attention layers we use positional encoding in both space and time. We apply linear attention [42] for space and use standard quadratic attention for time. For other implementation details, please see supp. mat.

5.1. Dynamic Replica

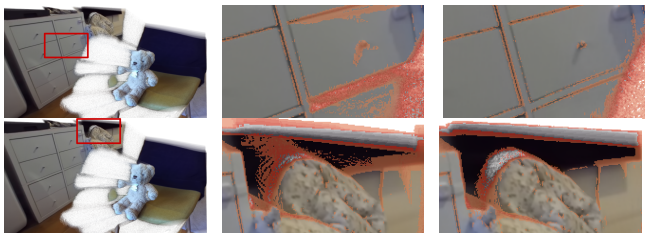
In Tab. 3 we show results obtained by training two recent state-of-the-art models, RAFT-Stereo [23] and CRE-Stereo [18] on SceneFlow, our dataset and their combination and testing the models' generalization to other disparity estimation datasets. While the main objective of our method is to produce temporally consistent disparity estimates, here we train state-of-the-art disparity estimation models to evaluate the usefulness of the dataset in this setting.

SceneFlow is an abstract dataset of moving shapes on colorful backgrounds. It aims at generalization through domain randomization, while our dataset consists of more realistic home and office scenes with people. Thus, while training on *Dynamic Replica* alone improves performance on the DR test set, it does not generalize as well as models trained on SceneFlow. However, combining both datasets boosts performance across both, datasets and models.

This highlights the benefits of including *Dynamic Replica* in the standard set of disparity estimation training datasets, even if its main goal is to enable training of temporally consistent models on longer sequences.

5.2. Temporal Consistency

The main objective of our method and dataset is to enable training of temporally consistent disparity estimators.



(mean image) RAFT Stereo [23] **DynamicStereo**

Table 4. **Temporal consistency.** Mean and variance of a 40-frame reconstructed static video, visualized. Both models are trained on DR & SF. We predict depth for each frame and convert it to globally aligned point clouds. Render combined point cloud with a camera displaced by 15 angles. Finally, we compute mean and variance across all images. Pixels with variance higher than 50px^2 are shown in red. Our method has lower variance.

To evaluate temporal consistency, we compute the temporal end-point-error (TEPE) defined as follows:

$$\text{TEPE}(\hat{D}, D) = \sqrt{\sum_{t=1}^{T-1} \left((\hat{D}_t - \hat{D}_{t+1}) - (D_t - D_{t+1}) \right)^2}. \quad (5)$$

This effectively measures the variation of the end-point-error across time. Lower values mean greater temporal consistency. In Tab. 5 we show that our model is more temporally consistent than prior methods across training datasets and benchmarks. It is even better than CRE-Stereo [18] that is trained on a combination of seven datasets. Additionally, models trained with *Dynamic Replica* are more consistent than models trained on SceneFlow alone.

With the above experiments, we have shown that *Dynamic Replica* is a useful dataset for training temporally consistent disparity estimators as well as standard stereo-frame models, and that our proposed model improves over the state of the art in both tasks when trained on our dataset.

In Tab. 4, we show a qualitative comparison of the temporal consistency between our model and RAFT-Stereo on a real-world sequence. We show the mean reconstruction over time of the sequence and color each pixel more red, the higher its variance across time. Our model significantly reduces the flickering that single-timestep models such as RAFT-Stereo produce. For more qualitative results and videos, please see the supplementary material.

5.3. Ablation Studies

In this section, we validate our method by ablating the design choices of our model. We train the model on SceneFlow [27] for 50k iterations with the hyper-parameters described in Implementation Details (see Sec. 5). We evaluate these models on the clean pass of Sintel [5] and on the test split of *Dynamic Replica*. We measure both accuracy and

Training data	Method	Sintel Stereo								Dynamic Replica			
		Clean				Final				First 150 frames			
		δ_{3px}	TEPE	δ^t_{1px}	δ^t_{3px}	δ_{3px}	TEPE	δ^t_{1px}	δ^t_{3px}	δ_{1px}	TEPE	δ^t_{1px}	δ^t_{3px}
SF	CODD [20]	8.68	1.44	10.78	5.65	17.46	2.32	18.56	9.79	6.59	0.105	1.04	0.42
	RAFT-Stereo [23]	6.12	0.92	9.33	4.51	10.40	2.10	13.69	7.08	5.51	0.145	2.03	0.65
	Ours	**6.10**	**0.77**	**8.41**	**3.93**	**8.97**	**1.45**	**11.95**	**5.98**	**3.44**	**0.087**	**0.75**	**0.24**
DR+SF	RAFT-Stereo [23]	**5.71**	0.84	9.15	4.40	9.16	2.27	13.45	7.17	**1.89**	**0.075**	0.77	0.25
	Ours	5.77	**0.76**	**8.46**	**3.93**	**8.68**	**1.42**	**11.93**	**5.92**	3.32	**0.075**	**0.68**	**0.23**
SF+M+K	CODD [20]	9.11	1.33	12.16	6.23	11.90	2.01	16.16	8.64	10.03	0.152	2.16	0.77
SF+M	RAFT-Stereo [23]	5.86	0.85	8.79	4.13	**8.47**	1.63	12.40	6.23	3.46	0.114	1.34	0.41
7 datasets (incl. Sintel)	CRE-Stereo [18]	4.58	0.67	6.36	3.26	8.17	1.90	12.29	6.87	**1.75**	0.088	0.88	0.29
DR+SF	Ours	**5.77**	**0.76**	**8.46**	**3.93**	8.68	**1.42**	**11.93**	**5.92**	3.32	**0.075**	**0.68**	**0.23**

Table 5. **Accuracy ◎ and Temporal consistency 🕒.** SF - SceneFlow [27], K - KITTI [28], M - Middlebury [37], DR - Dynamic Replica (ours). Temporal end-point error (TEPE) measures the consistency of the disparity estimation over time. We also compute δ^t_{1px} and δ^t_{3px} that show the proportion of pixels with TEPE higher than the threshold. Our model is more consistent than prior work and models. Additionally, other methods trained on our dataset also improve in temporal consistency. As CRE Stereo trains on Sintel and disparity estimation is evaluated on the training set of Sintel, CRE Stereo results are training set results and cannot be directly compared.

Method	Sintel Clean		Dynamic Replica	
	δ_{3px}	TEPE	δ_{1px}	TEPE
shared weights	6.60	0.908	6.48	**0.101**
sep. weights	**6.24**	**0.823**	**4.76**	0.126

Table 6. **Update Block Weight sharing.** Learning separate update blocks—one per resolution—consistently improves the results compared to weight sharing across all update blocks.

Method	Sintel Clean		Dynamic Replica	
	δ_{3px}	TEPE	δ_{1px}	TEPE
Conv2D	6.46	1.05	7.31	0.140
Conv3D	**6.24**	**0.823**	**4.76**	**0.126**

Table 7. **Update Block Convolution.** A GRU with a 3D convolution across space and time improves the performance over the 2D variant, especially in terms of the temporal metric TEPE.

temporal consistency. For accuracy, we use an end-point-error threshold of 3px for Sintel and 1px for DR. This shows the proportion of pixels with an end-point-error higher than 3px. For consistency, we use TEPE (see Sec. 5.2).

Update Block. In Tab. 6 we compare sharing weights of the three blocks across the three resolutions of the decoder to learning separate update blocks. As different scales exploit features of different resolutions, learning separate update blocks improves the results over weight-sharing.

Update Block Convolution. While prior works such as CRE Stereo use 2D convolutions in the iterative update block as they operate on single time steps, we find it beneficial to extend the processing of the update block across time (Tab. 7). This results in a general improvement but shows especially large improvements in temporal consistency.

Please see the supplement for additional analysis.

6. Conclusion

In this paper, we make two main contributions. We introduce a new stereo video dataset—*Dynamic Replica*—that allows training temporally consistent disparity estimators. Additionally, we introduce a new method that is able to improve over the state of the art in temporally consistent stereo estimation using the new dataset.

We show that other methods benefit from training on this new dataset as it contains realistic scenes and can thus reduce the domain gap between the real world and typical synthetic, abstract training datasets such as SceneFlow. Our combines spatial, temporal, and stereo information, enabling precise and consistent predictions across time. In extensive ablation studies, we show that each component of the model contributes to the final performance.

Acknowledgements

C. R. is supported by VisualAI EP/T028572/1. We would like to thank Filippos Kokkinos for helping to capture test videos, Roman Shapovalov and Luke Melas-Kyriazi for insightful discussions.

References

[1] Abhishek Badki, Alejandro Troccoli, Kihwan Kim, Jan Kautz, Pradeep Sen, and Orazio Gallo. Bi3d: Stereo depth estimation via binary classifications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1600–1608, 2020. 2

[2] Nicolas Ballas, Li Yao, Chris Pal, and Aaron C. Courville. Delving deeper into convolutional networks for learning video representations. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 2

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021. 2, 4

[4] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999. 2

[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, October 2012. 3, 7

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 2020. 2

[7] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418, 2018. 2

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *Proceedings of the 2019 Conference of the North*, 2019. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 3

[12] 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. 2

[13] Heiko Hirschmüller, Peter R Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1):229–246, 2002. 2

[14] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pages 66–75, 2017. 2

[15] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 508–515. IEEE, 2001. 2

[16] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[17] Philipp Krähenbühl. Free supervision from video games. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2955–2964, 2018. 3

[18] 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. 2, 3, 4, 7, 8

[19] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X. Creighton, Russell H. Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6197–6206, October 2021. 2

[20] Zhaoshuo Li, Wei Ye, Dilin Wang, Francis X Creighton, Russell H Taylor, Ganesh Venkatesh, and Mathias Unberath. Temporally consistent online depth estimation in dynamic scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3018–3027, 2023. 3, 8

[21] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4521–4530, 2019. 3

[22] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2811–2820, 2018. 2

[23] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021. 2, 7, 8

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[25] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. 39(4), 2020. 3

[26] 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. 6

[27] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. 2, 3, 7, 8

[28] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3, 7, 8

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[30] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 3

[31] 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*, 44(3):1623–1637, 2020. 3

[32] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222, 2017. 3

[33] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016. 3

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 3

[35] Sebastien Roy and Ingemar J Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, 1998. 2

[36] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 3

[37] 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 *German conference on pattern recognition*, pages 31–42. Springer, 2014. 7, 8

[38] 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. 7

[39] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 7

[40] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 6

[41] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T. Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10093–10102, June 2021. 2

[42] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 2, 3, 4, 7

[43] 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. 2

[44] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow (extended abstract). In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4839–4843. ijcai.org, 2021. 2, 4

[45] Jonathan Tremblay, Thang To, and Stan Birchfield. Falling things: A synthetic dataset for 3d object detection and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2038–2041, 2018. 2, 3

[46] Geert Van Meerbergen, Maarten Vergauwen, Marc Pollefeys, and Luc Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47(1):275–285, 2002. 2

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[48] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *2019 International Conference on 3D Vision (3DV)*, pages 348–357. IEEE, 2019. 3

[49] 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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916, 2020. 2, 3

[50] 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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 3

[51] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 2

[52] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015. 2

[53] 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. 2

[54] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-invariant stereo matching networks. In *European Conference on Computer Vision*, pages 420–439. Springer, 2020. 2

[55] Zhoutong Zhang, Forrester Cole, Richard Tucker, William T Freeman, and Tali Dekel. Consistent depth of moving objects in video. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 3

[56] C Lawrence Zitnick and Takeo Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on pattern analysis and machine intelligence*, 22(7), 2000. 2