Any-Stereo: Arbitrary Scale Disparity Estimation for Iterative Stereo Matching

Zhaohuai Liang [1], **Changhe Li** [2*]

[1] School of Automation, China University of Geosciences, Wuhan 430074, China
[2] School of Artificial Intelligence, Anhui University of Science & Technology, Hefei 232001, China
techhuaier@gmail.com, changhe.lw@gmail.com

Abstract

Due to unaffordable computational costs, the regularized disparity in iterative stereo matching is typically maintained at a lower resolution than the input. To regress the full resolution disparity, most stereo methods resort to convolutions to decode a fixed-scale output. However, they are inadequate for recovering vital high-frequency information lost during downsampling, limiting their performance on full-resolution prediction. In this paper, we introduce AnyStereo, an accurate and efficient disparity upsampling module with implicit neural representation for the iterative stereo pipeline. By modeling the disparity as a continuous representation over 2D spatial coordinates, subtle details can emerge from the latent space at arbitrary resolution. To further complement the missing information and details in the latent code, we propose two strategies: intra-scale similarity unfolding and cross-scale feature alignment. The former unfolds the neighbor relationships, while the latter introduces the context in high-resolution feature maps. The proposed AnyStereo can seamlessly replace the upsampling module in most iterative stereo models, improving their ability to capture fine details and generate arbitrary-scale disparities even with fewer parameters. With our method, the iterative stereo pipeline establishes a new state-of-the-art performance. The code is available at https://github.com/Zhaohuai-L/Any-Stereo.

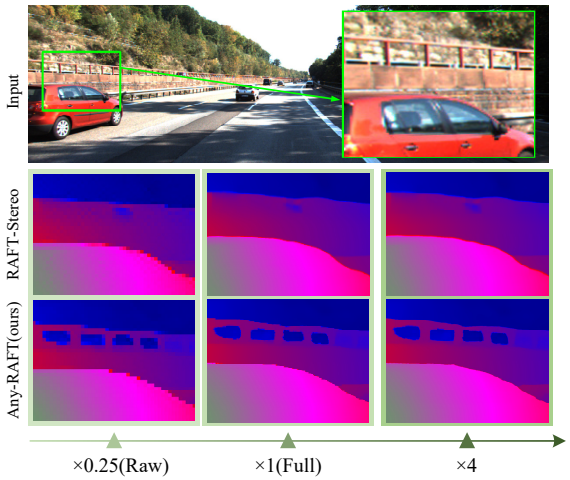

Figure 1: Overview examples of iterative methods at each stage on KITTI dataset. In RAFT-stereo (Lipson, Teed, and Deng 2021), the raw disparity is upsampled to full size by convolutions. For higher resolution, bilinear interpolation is used for comparison. In our Any-RAFT, fine disparities are decoded at arbitrary resolution by varying the upscale factor.

Introduction

Estimating depth from cameras is a fundamental task in many advanced vision applications such as augmented reality, autonomous driving, and 3D reconstruction. As a solution to this problem, stereo matching aims to identify the correspondence between two images from calibrated cameras, which represents the 3D information in the scene as a disparity map. The depth can be recovered from the camera-agnostic disparity map along with the intrinsics.

Motivated by the success of convolutional neural networks, learning-based stereo methods are proposed and have shown impressive results on challenge regions. Most early works focused on regularizing the 4D cost volume using 3D convolution (Chang and Chen 2018; Guo et al. 2019). To regress the disparity map, the regularized cost volume is upsampled to the original resolution, followed by a *soft argmin* operation on the disparity dimension. In this way, fine details can be well recovered from the cost volume. However, the high computational and memory consumption of 3D convolution hinders its application in high-resolution inputs.

Instead of regularizing the cost volume itself, recent work has focused on iteratively updating the disparity map by retrieving correlations from the cost volume (Lipson, Teed, and Deng 2021; Zhao et al. 2023; Xu et al. 2023). These iterative methods first construct a correlation volume with multiple receptive fields using feature pairs. After that, an update operator with GRU or LSTM is used to generate the residual disparity based on the correlation, context features, and current disparity map. By regularizing the 2D disparity map rather than the expensive cost volume, iterative methods are efficient enough to process high resolution images.

However, the capacity of iterative methods to recover fine depth details is constrained by upsampling from low-resolution disparity. The qualitative examples of the iterative methods at each stage are shown in Figure 1. For the

*Corresponding author

sake of reducing computational complexity, the update operator typically acts on feature maps at downsampled resolutions. Thus, the raw disparity map is maintained at a lower resolution than the input image. To recover the full-resolution disparity map, existing iterative methods exploit the convolution to predict neighbor combination weights from the hidden state of the update operator. Although it is possible to incorporate hidden context information into the raw low-resolution disparity map, there are still two limitations: (i) The convolutions are inadequate for recovering high-frequency information from the hidden state of the update operator. (ii) The context information cannot be fully preserved during downsampling and updating.

To address these two issues, we propose **AnyStereo**, an arbitrary scale upsampling module that efficiently and precisely recovers disparity at any scale. To decode the full-res disparity from the latent codes, we resort to implicit neural representation (INR). INR has been demonstrated to be effective in modeling various signals, such as images (Chen, Liu, and Wang 2021), optical flow (Jung et al. 2023) and 3D scenes (Mildenhall et al. 2021; Sitzmann et al. 2020), which is a fresh concept for representing the disparity map. Intuitively, we design an Implicit Neighbor Mask Function (INMF) to learn a mapping from the hidden state to a high-resolution disparity. Furthermore, we propose two strategies of Intra-scale Similarity Unfolding (ISU) and Cross-scale Feature Alignment (CFA) to achieve the embedding of intra-scale and cross-scale features, complementing the missing information during downsampling and iterative updating.

In summary, our main contributions are as follows:

- We introduce AnyStereo, an accurate and efficient disparity upsampling module with arbitrary scale outputs for the iterative stereo matching pipeline.
- We propose two novel strategies, ISU and CFA, to introduce intra-scale similarity and multi-scale context information to the upsampling.
- By replacing the convolutional upsampling with the designed module, iterative stereo models achieve a new SOTA even with fewer parameters.

Related Work

Learning-based Stereo Methods

Inspired by the conventional stereo matching pipeline, early learning-based stereo methods divided the model into four parts: feature extraction, cost construction, cost aggregation, and disparity regression. The given image pair is first encoded into 2D feature maps after feature extraction, and then the feature maps are constructed into a cost volume through correlation or concatenation. Correlation-based models (Mayer et al. 2016; Yang et al. 2018; Xu and Zhang 2020; Liang et al. 2018) construct a 3D cost volume by computing the similarity between pairs of pixels, and then utilize 2D convolutions to aggregate the cost volume. Given the limited representation ability of the 3D cost volume, concatenation-based methods (Kendall et al. 2017; Chang and Chen 2018; Zhang et al. 2019; Cheng et al. 2020) and hybrid methods (Guo et al. 2019; Xu et al. 2022) are proposed. The former concatenates the feature maps from both views at each disparity level, while the latter concatenates group-wise correlation, resulting in a 4D cost volume. Then a series of 3D convolutional layers are employed to regularize the informative 4D cost volume. To regress the refined disparity at full resolution, the regularized cost volume is upsampled to the original spatial size before the *soft argmin* operation on the disparity dimension. These methods are able to recover the refined disparity from the upsampled cost volume, however, their applicability to high-resolution inputs is limited by the computationally expensive 3D convolutions.

Recent methods pay their attention to the iterative pipeline, where the core idea is to iteratively update the disparity in a coarse-to-fine manner. RAFT-Stereo (Lipson, Teed, and Deng 2021) introduces a multi-level convolutional GRU to obtain the disparity deviation from the 3D correlation pyramid for the first time. IGEV-Stereo (Xu et al. 2023) incorporates 4D geometry encoding into the cost volume to complement the context information. DLNR (Zhao et al. 2023) designs a decoupled LSTM module and a normalization refinement strategy to harness high-frequency information. While these methods have shown impressive performance, the convolutional upsampling layers limit their final performance.

Implicit Neural Representation

Implicit neural representation, originating in 3D vision, is a technique for parameterizing various signals as continuous functions via multi-layer perceptron (MLP). It has been widely explored in the representation of 3D shapes (Chen and Zhang 2019; Michalkiewicz et al. 2019), 3D scenes (Sitzmann, Zollhoefer, and Wetzstein 2019) and 3D structures (Chen and Zhang 2019; Oechsle et al. 2019). Motivated by its success in 3D vision, recent works have attempted to apply the INR to 2D tasks. Among them, LIIF (Chen, Liu, and Wang 2021) proposes a local implicit function, which employs an MLP to replace sub-pixel convolution for super-resolution and predicts the image at arbitrary scales. By representing the image as a continuous function, LIIF shows great potential in capturing very fine details of an image, promoting applications of implicit neural representation in semantic segmentation (Gong et al. 2023; Sarkar et al. 2023), optical flow (Jung et al. 2023), and video (Chen et al. 2022b). Based on them, our choice for exploring effective disparity upsampling gravitates towards the implicit representation method of LIIF. Different from the previous method (Tosi et al. 2021), which learns a continuous representation to address the smoothness bias in classical stereo networks, we introduce LIIF into the iterative stereo matching pipeline as an implicit neighbor mask function.

Method

In this section, we first present an overview of the iterative stereo pipeline with the proposed AnyStereo as the upsampling module. We then detail the structure of AnyStereo, which consists of a implicit neighbor mask function, two strategies of intra-scale similarity unfolding and cross-scale feature alignment.

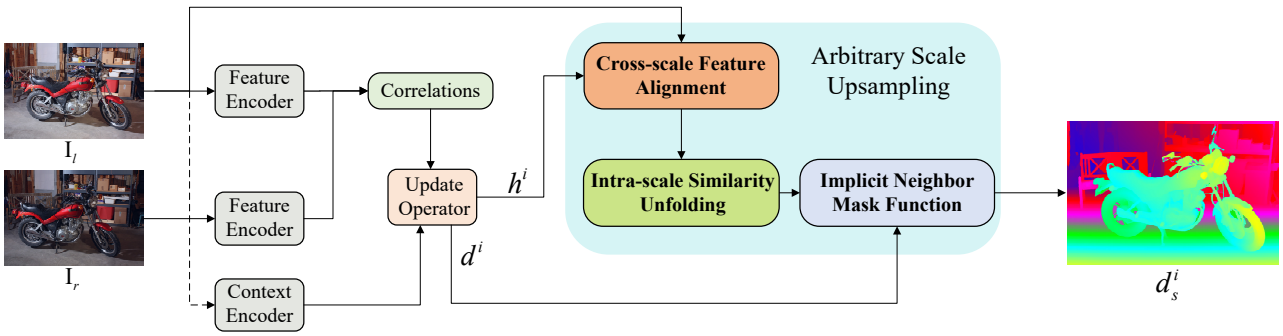

Figure 2: Overall architecture. Our network is based on the iterative stereo pipeline of RAFT-Stereo. The proposed arbitrary scale upsampling module is contained inside the highlighted box, where three novel components are introduced: the *Cross-scale Feature Alignment* (CFA) incorporates multi-scale context features into the latent code, the *Intra-scale Similarity unfolding* (ISU) enriches the local information contained in latent codes, and the *Implicit Neighbor Mask Function* (INMF) decodes the neighbor mask from the latent codes at arbitrary resolution.

Overall Framework

We base our design on the iterative stereo pipeline (Lipson, Teed, and Deng 2021). The overall network diagram is illustrated in Figure 2. Given a stereo image pair, the downscaled features can be obtained through feature encoders to construct the correlation volume. Then the correlation and encoded context features are concatenated and injected into the update operator to update the raw disparity d^i and hidden state h^i for ith iteration. Through intra-scale similarity unfolding and cross-scale feature alignment, local and context information is enriched in the hidden state. Based on it, the proposed implicit neighbor mask function can upsample the raw disparity d^i to arbitrary resolution d_s^i.

Implicit Neighbor Mask Function

The raw disparity outputted by the update operator is typically maintained at 1/4 resolution. To recover the full-resolution representations, current approaches employ convex upsampling, which treats the high-resolution disparity at each pixel as a convex combination of its 3×3 low resolution neighbors. By taking a weighted combination, high-resolution information can be incorporated in the form of a mask map. They compute disparity values at a predefined grid, and use convolutional layers to decode the corresponding neighbor mask from the hidden state of the update operator. While the convolutions are inadequate and inefficient for recovering high-frequency information. To address this problem, our key idea is to model the mask map as a continuous representation, which can restore fine details with only a few parameters and enable an arbitrary scale of output.

To map the discrete hidden state to a continuous neighbor mask, we introduce the INMF. It is inspired by implicit neural representations for image super-resolution (Chen, Liu, and Wang 2021). For a given query coordinate x_q, with hidden state $h^i \in \mathbb{R}^{H/4 \times W/4 \times C}$ as latent codes, the neighbor mask M at x_q is defined by:

$$M(x_q) = f_\theta\left(h^*, x_q - x^*\right) \tag{1}$$

where h^* is the nearest feature vector of hidden state h^i to the query coordinate x_q, and x^* is the coordinate of h^* .

The decoding function f_θ is parameterized by θ as a multi-layer perceptron (MLP). The output $M(x_q) \in \mathbb{R}^{3\times3}$ is the predicted neighbor weights of the nearest raw disparity from x_q. By sampling coordinates at different intervals for each spatial dimension in the $[-1, 1]$ range, the mask map $M \in \mathbb{R}^{sH \times sW \times 9}$ can be presented at arbitrary scales.

INFM performs upsampling in a pixel-based form, but the disparity is the displacement between two pixels, which changes with the size of the output. Thus the upscaled disparity at x_q is calculated by:

$$d_s^i(x_q) = s \times \sum_{n \in N} \left[d_n^i \times Softmax(M_n(x_q))\right] \tag{2}$$

where N denotes the set of 3×3 neighbor indices to x_q, and s is the upscale factor relative to the raw disparity, setting it to 4 means upscaling to the input resolution. For simplicity, we use s as the scale factor relative to the input in the following sections.

By adjusting the scale factor and the number of query samples, the raw disparity can be transformed into a disparity with arbitrary scales and arbitrary shapes.

Intra-scale Similarity Unfolding

Previous works (Chen, Liu, and Wang 2021; Sarkar et al. 2023) utilize Feature Unfolding to aggregate local information and context in each latent code, which is a common practice for implicit image representation. The feature unfolding of a feature map F is defined as:

$$\hat{F}_{jk} = \text{Concat}(\{F_{j+u,k+v}\}_{u,v\in\{-1,0,1\}}) \tag{3}$$

where each channel in $\hat{F}$ is the concatenation of its 3×3 neighbor vectors in F. An input of dimension $(H \times W \times C)$ would result in an output of $(H \times W \times 9C)$. Due to the 9-fold channel size, the computational cost incurred by feature unfolding is prohibitive for iterative disparity upsampling.

To compensate for local information with an acceptable cost, we introduce the ISU, as shown in Figure 3. The self-similarity is a robust local descriptor, indicating the relationship between a pixel and its neighborhood region. After a

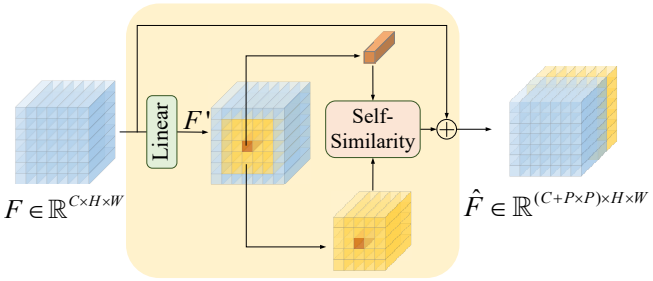

Figure 3: The Intra-scale Similarity Unfolding diagram. It unfolds the self-similarity scalar instead of feature vectors to reduce computational cost. The similarity unfolding of a feature map $F \in \mathbb{R}^{C\times H\times W}$ would end up as $\hat{F} \in \mathbb{R}^{(C+P\times P)\times H\times W}$

linear layer to reduce channel size, we concatenate features for each pixel with the channel-wise self-similarity to $P \times P$ neighbors. The similarity unfolding process is formulated as:

$$\hat{F}_{jk} = \text{Concat}(F_{jk}, \{\mathcal{S}(F_{jk}, F'_{j+u,k+v})\}_{u,v\in\{-P/2,\dots,P/2\}}) \tag{4}$$

where F' is the projection of input feature map F, $\mathcal{S}$ is a function to compute non-negative cosine similarity.

Cross-scale Feature Alignment

For introducing context information into the upsampling, current approaches exploit the accumulated information of the update operator, i.e., the hidden state, to avoid additional encoding operations. While the limited context information in the hidden state constrains the INMF to align the disparity map with the context of the input image. During the iterative process, the update operator leverages the hidden state to regularize the low-resolution disparity, in which the context information cannot be fully maintained. Moreover, the downsampling convolution in the encoder also leads to the loss of high-resolution context information. To alleviate the problem, we introduce a specific branch to enhance the context information in latent codes.

To encode context information in multi-scale feature maps, we devise a Spatial Downsampling Block (SDB), as shown in the right of Figure 4. PixelUnshuffle is used to downscale the spatial dimensions while preserving high-resolution information. Besides, Simplified Channel Attention (SCA) (Chen et al. 2022a) is employed to maintain the global information. We use two SDB blocks to extract the context information in the left image, resulting in multi-scale feature maps $\{F^s\}_{s\in\{1/2,1/4\}}$.

To embed multi-scale features into latent codes, we extend the INMF to:

$$M(x_q) = f_\theta(\{z_l^*\}_{l\in\{1/2,1/4\}}, \{x_q - x_l^*\}_{l\in\{1/2,1/4\}}) \tag{5}$$

where l denotes the index of features at different scales. z_l^* is the nearest feature vector from x_q at scale l, and x_l^* is the corresponding coordinates. The feature vectors at 1/4 scale are obtained by stacking the information from hidden state

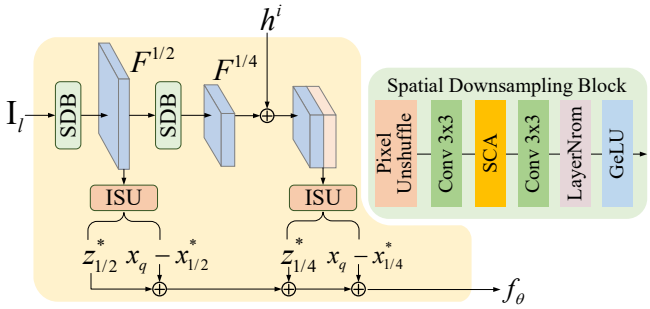

Figure 4: Illustration of the Cross-scale Feature Alignment. The left image is passed into a series of SDBs, which output multi-scale feature maps with context information. SCA denotes simplified channel attention (Chen et al. 2022a). To benefit from cross-scale information, we obtain the nearest feature vectors and relative coordinates from x_q for each scale, which are then concatenated and fed into the decoding function.

$\mathbf{h}^i$ and feature map $F^{1/4}$:

$$z_{1/4} = \text{Concat}(F^{1/4}, h_i) \tag{6}$$

ISU is performed for features at each scale. Given a query coordinate x_q, we extract the nearest feature vector $\{z_l^*\}_{l\in\{1/2,1/4\}}$ for each scale, as well as the relative coordinate to x_q, which are then concatenated and fed into the MLP of the neighbor implicit mask function. The entire process is illustrated in Figure 4.

Multi-scale Training Strategy

We jointly train the iterative network with our upsampling module in a multi-scale manner. During preparing training pairs, we first uniformly sample scale factor s from $[\times 1, \times 3]$, and then randomly crop $sH \times sW$ patches from the image pair and ground truth. After that, we downscale the image patch to a fixed size of $H \times W$ by bicubic interpolation, and randomly sample HW disparities from the ground truth patch. During training, we set the scale factor to s and query HW pixels, resulting in HW predictions with scale s for supervision. Following the iterative pipeline (Lipson, Teed, and Deng 2021), we use $L1$ loss with exponential incremental weights to supervise the predicted disparities at each iteration:

$$\mathcal{L} = \sum_{i=1}^{T} \gamma^{T-i} ||d_{gt} - d_s^i||_1 \tag{7}$$

where $\gamma = 0.9$, T is the number of update iteration for training, d_{gt} denotes ground truth.

Experiments

Datasets & Evaluation Metrics

Datasets The SceneFlow dataset (Mayer et al. 2016), KITTI dataset (Geiger, Lenz, and Urtasun 2012; Menze and Geiger 2015) and Middlebury dataset (Scharstein et al.

Model	KITTI2015 All (%))			KITTI2015 Noc(%)			KITTI2012		Params.
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	Out-Noc	Out-All	(M)
PSMNet(2018)	1.86	4.62	2.32	1.71	4.31	2.14	1.49	1.89	5.5
GANet (2019)	1.48	3.46	1.81	1.60	3.11	1.63	1.19	1.60	6.58
AANet (2020)	1.65	3.96	2.03	1.49	3.66	1.85	1.55	2.04	3.9
LEAStereo (2020)	1.40	2.91	1.65	1.29	2.65	1.51	1.13	1.45	1.81
SMD-Net (2021)	1.69	4.01	2.08	1.54	3.70	1.89	-	-	6.10
RAFT-Stereo (2021)	1.58	3.05	1.82	1.45	2.94	1.69	1.30	1.66	11.23
ACVNet (2022)	**1.37**	3.07	1.65	**1.26**	2.84	1.52	1.13	1.47	6.22
DLNR (2023)	1.60	2.59	1.76	1.45	2.39	1.61	-	-	57.38
IGEV-Stereo (2023)	1.38	2.67	1.59	1.27	2.62	1.49	1.12	1.44	12.60
Any-RAFT (ours)	1.44	3.04	1.70	1.30	2.88	1.56	1.18	1.52	10.90
Any-IGEV (ours)	1.43	**2.35**	**1.58**	1.31	**2.27**	**1.47**	**1.11**	**1.41**	12.49

Table 1: Quantitative evaluation on KITTI 2015 and KITTI 2012. The best results for each metric are bolded.

2014) are used. SceneFlow (Mayer et al. 2016) is a large synthetic dataset including 35k training pairs of stereo images with dense ground-truth disparities, which is produced by rendering a 3D model. KITTI, including KITTI12 (Geiger, Lenz, and Urtasun 2012) and KITTI15 (Menze and Geiger 2015), is a real-world outdoor dataset for driving scenes with sparse annotations recorded by LIDAR. KITTI12 consists of 194 training pairs and 195 testing pairs, and KITTI15 consists of 200 training pairs and 200 testing pairs. Middlebury (Scharstein et al. 2014) is a real-world indoor dataset, including 15 image pairs for training and 15 pairs for testing with multiple resolutions.

Evaluation Metrics The EPE (End-Point Error) and $>\tau$ px (percentage of pixels with errors greater than τ) are used to evaluate the performance of prediction and upsampling. The EPE measures the mean disparity error over all pixels, while the $>\tau$ px computes the percentage of points with absolute error larger than a specific threshold τ. For KITTI, >3 px is extended into "D1" (max(3 px, 0.05d)).

Implementation Details

We implement our upsampling module with iterative stereo methods (Lipson, Teed, and Deng 2021; Xu et al. 2023) by Pytorch and perform our experiments on two NVIDIA A40 GPUs. For all training, we use the AdamW (Kingma and Ba 2014) optimizer with a one-cycle learning rate schedule and clip gradients to [-1,1]. For the ISU, we set window size p as 5. Following (Lipson, Teed, and Deng 2021), we use data augmentation for training. The model for ablation is trained on SceneFlow for 100k steps with a batch size 6. The final model is pretrained on SceneFlow for 200k steps with a batch size of 8, and then finetuned on KITTI and Middlebury. All experiments are run with 22 update iterations during training, 32 during evaluation. The others are set as same as in the basic model.

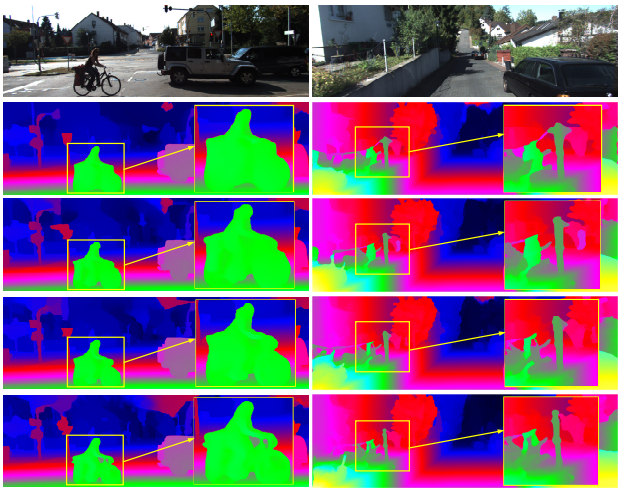

Figure 5: Qualitative comparisons on the test set of KITTI (from top to bottom: left images, disparity maps estimated by RAFT-Stereo, Any-RAFT (ours), IGEV-Stereo, and Any-IGEV (ours)).

Comparisons with State-of-the-art

To demonstrate the gains brought by our method, we embed it into RAFT-Stereo and IGEV-Stereo to replace the original upsampling layers, and submit them to KITTI15 and KITTI12 benchmarks for comparison. All models are pretrained on the SceneFlow dataset, and fine-tuned on the mixed KITTI2012 and KITTI 2015 training sets for 50k iterations with a one-cycle learning rate of 0.0002. Note that we fixed the scale s to 1 to output full resolution disparity for fine-tuning.

Evaluation results are listed in Table 1. As can be seen, iterative methods achieve significant improvements by embedding our method as an upsampler, and even have fewer parameters than the original model. Benefiting from the pro-

Model	KITTI2015				KITTI2012				Middlebury Q			
	100%	75%	50%	25%	100%	75%	50%	25%	100%	75%	50%	25%
RAFT-Stereo(2021)	5.61	6.30	7.47	21.0	5.24	5.56	6.68	21.5	6.71	9.52	16.5	39.9
IGEV-Stereo(2023)	5.85	6.57	7.56	18.2	5.70	6.09	7.49	20.1	6.21	7.96	14.8	36.0
Any-RAFT(ours)	5.60	5.89	6.44	10.6	4.78	4.67	5.28	9.84	5.85	6.58	9.51	24.9
Any-IGEV(ours)	5.64	6.09	6.59	11.5	6.02	5.82	6.36	12.9	6.00	6.59	10.4	26.0

Table 2: Quantitative comparisons of performance for downsampled images on KITTI15, KITTI12 and MIddlebury. D1(%) and >2 px(%) are taken for KITTI and Middlebury, respectively.

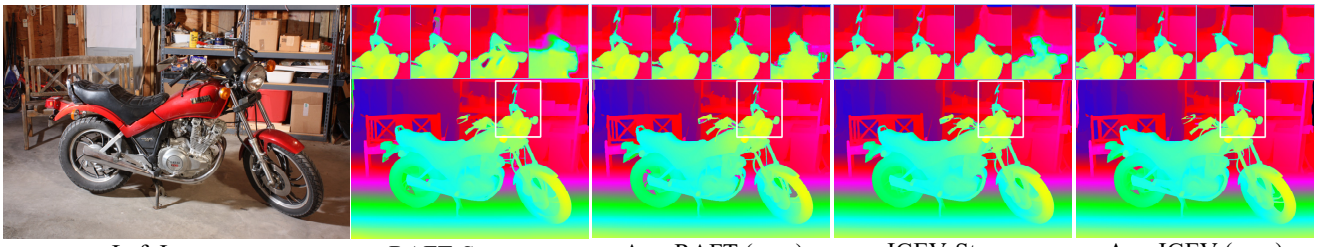

Figure 6: Qualitative comparisons on the MIddlebury. The top row shows the performance degradation with the scaled input from 100% to 25%, and the bottom is the disparity map with the original input (100%).

posed upsampling module, Any-RAFT outperforms RAFT-Stereo by a large margin on both benchmarks. With SOTA model IGEV-Stereo as the baseline, we have improved the D1-fg (Noc) from 2.62% to 2.27% (13.4% improvement) on KITTI15, and Out-All from 1.44% to 1.41% on KITTI12. At the time of submission, our Any-IGEV achieves the best results among all published works on D1-fg and D1-all metrics, and ranks 1^{st} on KITTI 2015 leaderboard. The qualitative results are given in Figure 5, which shows our method excels at recovering detailed structures.

Recover Details from Downsampled Images

To quantitatively evaluate the effectiveness of the learned disparity representation, we compare the performance degradation of models for downsampled inputs. All models are trained on SceneFlow and evaluated on the training sets of KITTI15, KITTI12 and Middlebury. As shown in Table 2, our method significantly stabilizes the degradation of iterative models as the scale varies. On both datasets, the models perform similarly to the original size, but when the image is gradually scaled to 25%, the models with our method have nearly 50% improvements compared to the baseline. On the KITTI12 dataset, it is worth noting that our Any-RAFT and Any-IGEV even do not show any degradation for up to 75% of downsampling.

The degradation at each scale is visualized in Figure 6. As illustrated, the original RAFT-Stereo and IGEV-Stereo can hardly maintain thin structures in the disparity with 50% downsampled images. After downsampling, the structures of the original image become very tiny, which is lost during low-resolution updating and hard to be recovered by the convolutional layers from limited information. Instead, through the learned continuous representation, Any-RAFT and Any-IGEV maintain the structure up to 50% of dwonsampling. This further demonstrates that our method restores boundaries and tiny objects even from low-resolution images, and is agnostic to the resolution of images.

Upsample with Arbitrary Scale

Estimating high-resolution disparity from low-resolution images is beneficial to reduce the computational cost, which is important for deployment to mobile devices. Towards this end, we compare our method with the previous disparity upsampling method SMD-Net (Tosi et al. 2021) and other super-resolution methods, including bicubic interpolation and LIIF (Chen, Liu, and Wang 2021). For a fair comparison with super-resolution methods, we use RAFT-Stereo to predict full-resolution disparity maps, which are rendered to a color map and then upsampled to an arbitrary scale.

The results of upsampling are visualized in Figure 7. As can be observed, bicubic interpolation blurs the details, especially in the boundaries. LIIF propagates the errors of coarse inputs when the scale becomes finer, resulting in artifacts on the edge. Different from them, our Any-RAFT is capable of recovering more clear edges, showing a smoother handle and fewer artifacts in Figure 7. It is because we exploit the cumulative hidden state as latent codes, and incorporate contextual information from the input image. Compared to SMD-Net, our method represents the object geometry more precisely, benefiting from the information encoded by the iterative stereo pipeline.

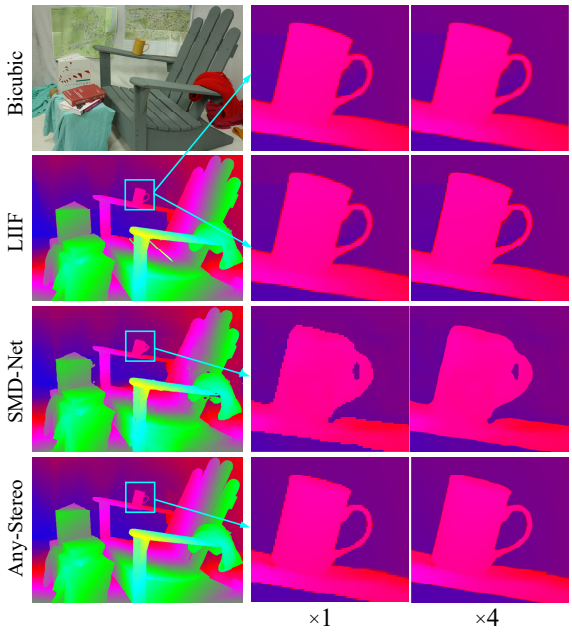

Figure 7: Qualitative comparison with super-resolution methods on Middlebury. The left image is attached in the top-left corner.

	Method	MST.	EPE (px)	>1px (%)	Params. (M)
INMF	Conv.	✗	0.88	7.87	11.23
	INMF	✗	0.83	7.61	10.82
	INMF	✓	0.83	7.69	10.82
ISU	w/o ISU	✓	0.77	6.99	10.90
	ISU($P = 3$)	✓	0.76	6.87	10.90
	ISU($P = 5$)	✓	0.75	6.80	10.90
CFA	w/o CFA	✓	0.80	7.34	10.82
	CFA(w/o SCA)	✓	0.76	6.85	10.90
	CFA	✓	0.75	6.80	10.90

Table 3: Albation study of proposed three components on the SceneFlow test set. "Conv." denotes the decoding convolutional layers, and "MST" is abbreviated for multi-scale training strategy.

Ablation Study

The ablation study is performed to explore the best setting and validate the effectiveness of each component of the proposed method. RAFT-Stereo (Lipson, Teed, and Deng 2021) is adopted as the backbone in this part. All models are trained on SceneFlow training sets, and evaluated on the testing sets with a fixed scale to full resolution.

In the first three rows of Table 3, we compare our INMF with the original convolutional upsampling. By taking full advantage of the encoded information, INFM achieves better results with fewer parameters than convolutional decoding layers. Furthermore, INMF enables a multi-scale training strategy to train the network. Although slightly inferior to fixed-scale training on SceneFlow, multi-scale training helps the network learn a continuous representation, which is beneficial for arbitrary scale disparity prediction and subsequent fine-tuning.

In the middle of Table 3, we report the effect of intra-scale similarity unfolding (ISU) and its window size setting. The results indicate that ISU improves the performance of continuous representation by revealing the local neighborhood for each latent code. Enlarging the window size of ISU yields better results due to larger receptive fields. Considering the computational cost, we set window size P to 5.

Finally, we evaluate the proposed cross-feature alignment (CFA), as shown in the last three rows of Table 3. After aligning multi-scale features as the latent codes, the performance of the full model is significantly improved. This is because CFA introduces context information into the latent codes, which is vital in recovering the high-resolution disparity. As a strong complement to the iterative pipeline, CFA helps the INMF align the disparity with the context of the input image. The simplified channel attention (SCA) further enriches the global information contained in latent codes.

Methods	Runtime(s)	Flops(G)	Params.(M)
RAFT	0.415	2844	11.23
Any-RAFT	0.422 ↑(1.7%)	2857	10.90
Any-RAFT w/o ISU	0.419 ↑(1.0%)	2854	10.90
Any-RAFT w/o CFA	0.413 ↓(0.5%)	2846	10.82

Table 4: Complexity analysis of the proposed method with input resolution at 960 × 540.

Complexity Analysis

We report the runtime breakdown of our method based on images of SceneFlow, as listed in Table 4. In initial experiments, we noticed the huge computational cost of LIIF, mainly incurred by local ensemble and feature unfolding. For this reason, instead of adopting both, we propose an alternative strategy, i.e., ISU. Moreover, the original upsampling contains a lot of costly 2D convolution layers. After replacing it with our method, the overall time consumption is similar to the baseline.

Conclusion

In this paper, we present a precise and efficient disparity upsampling module, Any-Stereo, to replace the imperfect convolutional upsampling in iterative stereo models. The proposed INMF models the disparity as a continuous representation, recovering the fine disparity from the coarse intermediate features. The ISU with an acceptable cost is introduced as compensation for the absence of feature unfolding. The CFL with SDB blocks is proposed as a specific branch to encode multi-scale context. As a lightweight plug-in module, AnyStereo can seamlessly combine with iterative stereo models, boosting their ability to capture fine details and generate arbitrary-scale disparities.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62076226, in part by the Hubei Provincial Natural Science Foundation of China under Grant 2023AFA049, in part by the 111 project under Grant B17040.

References

Chang, J.-R.; and Chen, Y.-S. 2018. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5418.

Chen, L.; Chu, X.; Zhang, X.; and Sun, J. 2022a. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*.

Chen, Y.; Liu, S.; and Wang, X. 2021. Learning Continuous Image Representation with Local Implicit Image Function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8624–8634.

Chen, Z.; Chen, Y.; Liu, J.; Xu, X.; Goel, V.; Wang, Z.; Shi, H.; and Wang, X. 2022b. VideoINR: Learning Video Implicit Neural Representation for Continuous Space-Time Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2047–2057.

Chen, Z.; and Zhang, H. 2019. Learning Implicit Fields for Generative Shape Modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5932–5941.

Cheng, X.; Zhong, Y.; Harandi, M.; Dai, Y.; Chang, X.; Li, H.; Drummond, T.; and Ge, Z. 2020. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 22158–22169.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361.

Gong, R.; Wang, Q.; Danelljan, M.; Dai, D.; and Van Gool, L. 2023. Continuous Pseudo-Label Rectified Domain Adaptive Semantic Segmentation With Implicit Neural Representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7225–7235.

Guo, X.; Yang, K.; Yang, W.; Wang, X.; and Li, H. 2019. Group-wise correlation stereo network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3273–3282.

Jung, H.; Hui, Z.; Luo, L.; Yang, H.; Liu, F.; Yoo, S.; Ranjan, R.; and Demandolx, D. 2023. AnyFlow: Arbitrary Scale Optical Flow With Implicit Neural Representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5455–5465.

Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; and Bry, A. 2017. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 66–75.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liang, Z.; Feng, Y.; Guo, Y.; Liu, H.; Chen, W.; Qiao, L.; Zhou, L.; and Zhang, J. 2018. Learning for Disparity Estimation Through Feature Constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2811–2820.

Lipson, L.; Teed, Z.; and Deng, J. 2021. RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. In *2021 International Conference on 3D Vision (3DV)*, 218–227.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. 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*, 4040–4048.

Menze, M.; and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3061–3070.

Michalkiewicz, M.; Pontes, J. K.; Jack, D.; Baktashmotlagh, M.; and Eriksson, A. 2019. Implicit Surface Representations As Layers in Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Oechsle, M.; Mescheder, L.; Niemeyer, M.; Strauss, T.; and Geiger, A. 2019. Texture Fields: Learning Texture Representations in Function Space. In *Proceedings of the IEEE International Conference on Computer Vision*, 4530–4539.

Sarkar, M.; Nikitha, S.; Hemani, M.; Jain, R.; and Krishnamurthy, B. 2023. Parameter Efficient Local Implicit Image Function Network for Face Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 20970–20980.

Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nešić, N.; Wang, X.; and Westling, P. 2014. High-resolution stereo datasets with subpixel-accurate ground truth. In *Proceedings of the German Conference on Pattern Recognition*, 31–42. Springer.

Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; and Wetzstein, G. 2020. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*, 7462–7473.

Sitzmann, V.; Zollhoefer, M.; and Wetzstein, G. 2019. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *Advances in Neural Information Processing Systems*.

Tosi, F.; Liao, Y.; Schmitt, C.; and Geiger, A. 2021. SMD-Nets: Stereo Mixture Density Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8942–8952.

Xu, G.; Cheng, J.; Guo, P.; and Yang, X. 2022. Attention Concatenation Volume for Accurate and Efficient Stereo Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12981–12990.

Xu, G.; Wang, X.; Ding, X.; and Yang, X. 2023. Iterative Geometry Encoding Volume for Stereo Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21919–21928.

Xu, H.; and Zhang, J. 2020. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1959–1968.

Yang, G.; Zhao, H.; Shi, J.; Deng, Z.; and Jia, J. 2018. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Vomputer Vision*, 636–651.

Zhang, F.; Prisacariu, V.; Yang, R.; and Torr, P. H. 2019. GA-Net: Guided Aggregation Net for End-To-End Stereo Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 185–194.

Zhao, H.; Zhou, H.; Zhang, Y.; Chen, J.; Yang, Y.; and Zhao, Y. 2023. High-Frequency Stereo Matching Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1327–1336.