Fast-FoundationStereo: Real-Time Zero-Shot Stereo Matching

Bowen Wen Shaurya Dewan Stan Birchfield

NVIDIA

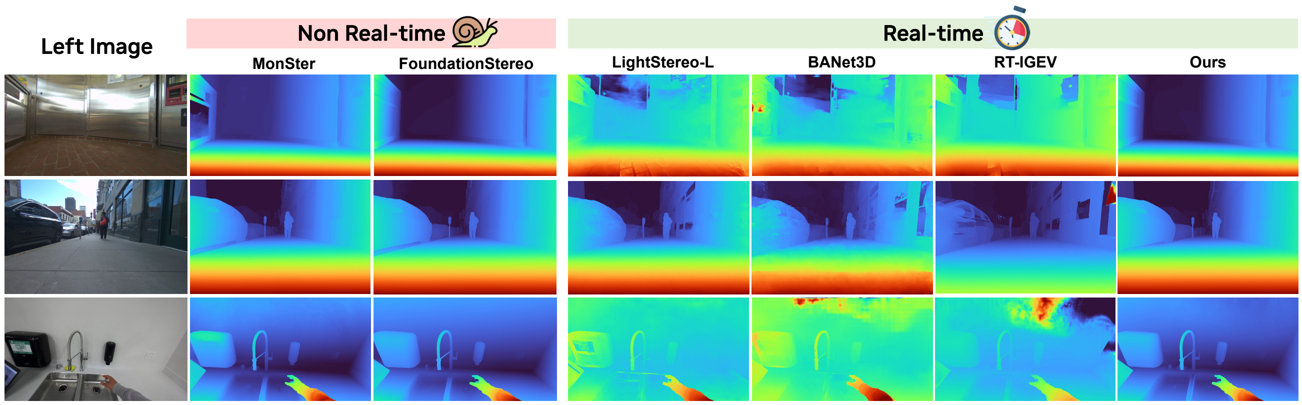

Figure 1. Our *Fast-FoundationStereo* achieves comparable results to MonSter [7] and FoundationStereo [68] while running nearly 10 times faster. Shown are disparity maps obtained by zero-shot inference on in-the-wild images. (Note that our results occasionally exceed those of [7], *e.g.*, the shiny door in the top row, and the paper towel bin in the bottom row.)

Abstract

Stereo foundation models achieve strong zero-shot generalization but remain computationally prohibitive for real-time applications. Efficient stereo architectures, on the other hand, sacrifice robustness for speed and require costly per-domain fine-tuning. To bridge this gap, we present Fast-FoundationStereo, a family of architectures that achieve, for the first time, strong zero-shot generalization at real-time frame rate. We employ a divide-and-conquer acceleration strategy with three components: (1) knowledge distillation to compress the hybrid backbone into a single efficient student; (2) blockwise neural architecture search for automatically discovering optimal cost filtering designs under latency budgets, reducing search complexity exponentially; and (3) structured pruning for eliminating redundancy in the iterative refinement module. Furthermore, we introduce an automatic pseudo-labeling pipeline used to curate 1.4M in-the-wild stereo pairs to supplement synthetic training data and facilitate knowledge distillation. The resulting model can run over 10× faster than FoundationStereo while closely matching its zero-shot accuracy, thus establishing a new state-of-the-art among real-time methods. Project page: https://nvlabs.github.io/Fast-FoundationStereo/

1. Introduction

The field of stereo matching has advanced significantly since its inception exactly 50 years ago [38]. Modern algorithms, driven by an abundance of high-quality training

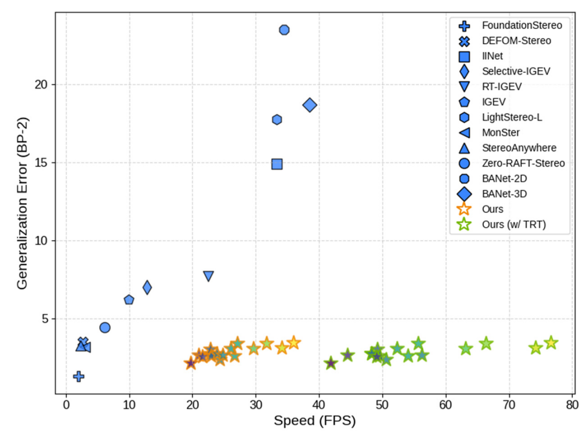

Figure 2. Zero-shot generalization accuracy (on Middlebury-Q dataset) of various stereo methods versus speed, measured on the same hardware NVIDIA 3090 GPU. Our model family achieves real-time performance with only slight decrease in accuracy compared with the best slow method. Green outlined stars are further accelerated by TensorRT.

datasets and innovations in deep neural network architectures, now yield impressive results, often approaching saturation on the most demanding benchmarks. Such accuracy is critical for applications requiring precise 3D reconstruction, such as robotics [31] and augmented reality [30].

This remarkable progress, however, has split the field into two distinct research paths [62]. On the one hand, the rise of foundation models in computer vision has pushed stereo research toward strong zero-shot generalization [2, 7, 66, 68]. Such leading zero-shot networks leverage rich priors from computationally intensive foundation models such as DepthAnythingV2 [78] or DINO models [47, 60]; and they employ computationally intensive architectures, such as the Disparity Transformer [68], to perform self-attention for long-range context. Such limitations have, to date, hindered their deployment in any latency-bound system.

On the other hand, the non-negotiable constraints of practical applications demand computationally efficient performance. Architectures designed for such real-time inference [22, 33, 74, 75] achieve high frame rates by relying on lightweight backbones, 2D convolutional layers, and local iterative refinement modules. Such methods struggle to generalize due to their reliance upon per-domain fine-tuning. The difficulty of obtaining the required dense, high-quality ground-truth depth at scale has prevented such efficient methods from being used as an off-the-shelf solution for embodied agents operating in in-the-wild environments.

To address this critical gap, we propose Fast-FoundationStereo (Fig. 1), a novel stereo matching approach for both strong zero-shot generalization and real-time inference. Unlike existing real-time methods which sacrifice the rich architectural capacity and typically designed and trained from scratch, our work builds upon the powerful yet computationally intensive FoundationStereo [68]. Addressing its three main components (feature extraction, cost filtering, and disparity refinement), our divide-and-conquer acceleration strategy takes into account the unique properties of each. First, knowledge distillation is leveraged to compress the computationally expensive hybrid feature backbone into a single, efficient student backbone that retains the rich monocular and stereo priors. Second, the intensive cost filtering network is divided into blocks, numerous candidate blocks are trained via distillation, and combinatorial search automatically discovers a family of effective architectures under varying latency budgets. Third, structured pruning is applied to the refinement module, guided by a recurrent dependency graph to identify and remove redundancy, followed by retraining to recover performance. Finally, training is supplemented with a large-scale (1.4M pairs) dataset of in-the-wild stereo images, curated via an automatic pseudo-labeling pipeline.

Our contributions can be summarized as follows:

- We present Fast-FoundationStereo, a novel stereo matching architecture that achieves both strong zero-shot generalization and real-time inference, with varying accuracy-speed trade-off (Fig. 2). Our method significantly outperforms other real-time models by a large margin across multiple public datasets, and even outperforms several recent strongly generalizable models.
- We present several novelties to address the computational bottleneck of common components adopted in modern stereo matching models, while inheriting the strengths from the teacher model. Our divide-and-conquer strategy includes: (1) distillation from hybrid monocular and stereo priors, (2) cost filtering via efficient blockwise architecture search, and (3) iterative refinement via structured pruning.
- To harness the large diversity, internet-scale abundance and unique realism from in-the-wild stereo images, we propose an automatic pseudo-labeling pipeline to supplement synthetic training data for knowledge distillation.

2. Related Work

Generalizable Stereo Matching. Recent progress in generalizable stereo matching has centered on leveraging Vision Foundation Models (VFMs) and monocular priors to achieve strong zero-shot performance. FoundationStereo [68] establishes a strong baseline by adapting DepthAnythingV2 with side-tuning, while StereoAnywhere [2] demonstrates robustness where stereo or mono cues fail independently, and MonSter [7] marries monocular depth with stereo matching to unleash complementary strengths. ZeroStereo [66] synthesizes additional training data based on monocular depth estimation and diffusion models. DEFOM-Stereo [25] builds upon depth foundation models, All-in-One [88] systematically transfers VFMs into stereo frameworks, and recent work diving into the fusion of monocular priors [79] analyzes effective integration strategies. Beyond direct adaptation, domain generalization has been pursued through domain-invariant representations [83], learning from foundation models for domain generalized stereo matching [84], and information-theoretic approaches that avoid shortcut learning [9]. Additional architectural innovations include hierarchical visual transformations [5], masked representation learning for domain generalized stereo matching [51], and harnessing broad-spectrum task-oriented features [35]. Despite impressive zero-shot generalization, their computational overhead remains prohibitive for real-time applications.

Efficiency-Oriented Stereo Matching. Efficiency-oriented stereo matching architectures have traditionally pursued real-time performance through three primary strategies: compact cost volume representations, lightweight processing modules, and streamlined network designs. The first strategy reduces memory footprint via low-resolution feature pyramids [27], 2D cost signatures [80], attention-based disparity selection [72], or learned parameterized functions that replace explicit volumes entirely [81]. The second strategy accelerates cost aggregation by pruning search spaces in coarse-to-fine cascades [19], operating in efficient bilateral grid spaces [71], or employing 3D separable convolutions [49] to avoid expensive 3D kernels. The third strategy designs mobile-specific architectures [57], tile-based iterative refinement [61], or binary operations [3] from the ground up, while more sophisticated approaches employ neural architecture search to automatically discover efficient networks [8, 64]. While these methods achieve impressive

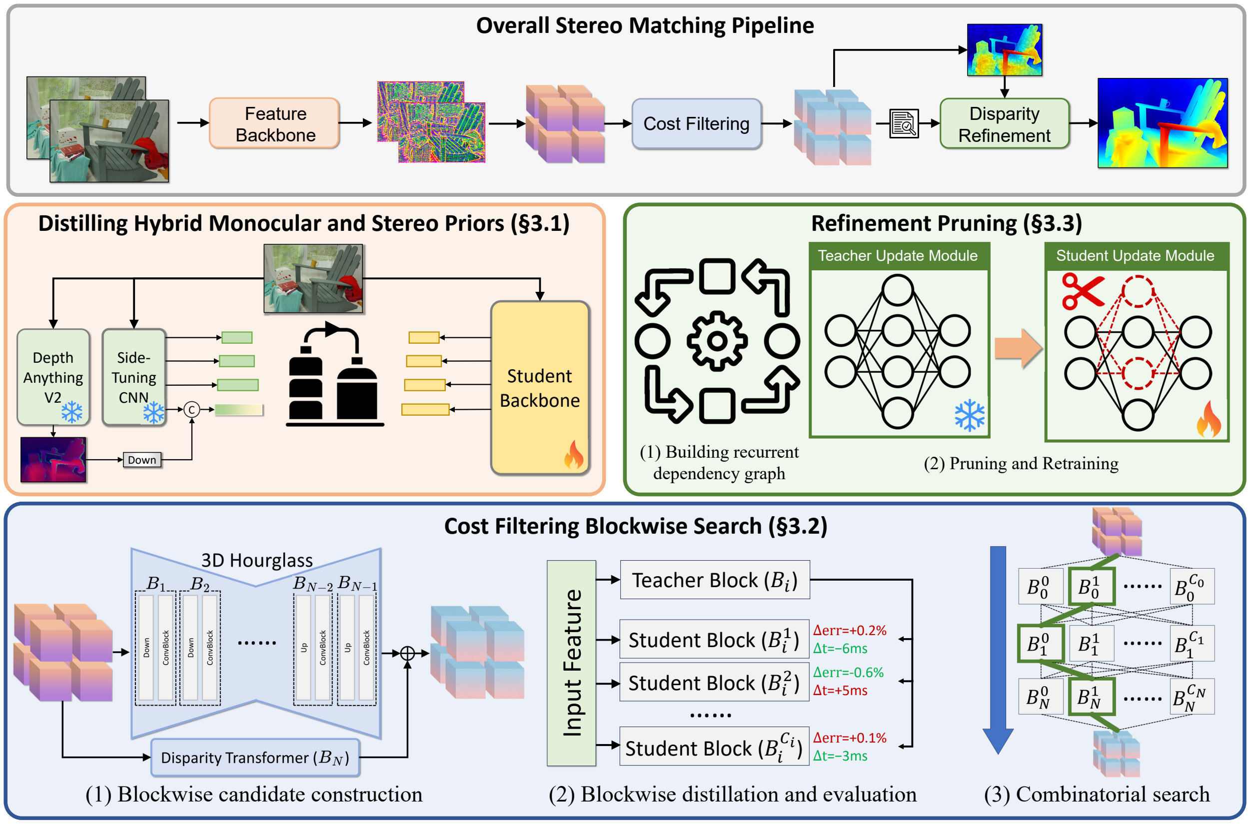

Figure 3. Overview of our framework. **Top:** Foundational stereo matching networks (*e.g.*, [68]) consist of three key steps: feature extraction, cost filtering, and disparity refinement. Each step is accelerated by a divide-and-conquer strategy. **Middle-Left:** Hybrid monocular and stereo priors from the teacher foundation model are distilled into a single backbone student model. **Middle-right:** Refinement network is pruned by first constructing a dependency graph that models the recurrent nature of the GRU module, followed by structured pruning and retraining to recover the accuracy. **Bottom:** Cost filtering network is divided into separate local blocks; block candidates are trained to match the teacher block's output, taking as input the local feature from the previous block; and combinatorial search finds the best performing block combination for a given runtime constraint.

frame rates, they fundamentally sacrifice the rich architectural capacity. In addition, the models are usually designed and trained from scratch, ignoring recent powerful foundation models. Consequently, they remain tethered to per-domain fine-tuning on target distributions, making them unsuitable solutions for in-the-wild environments.

Vision Foundation Model Acceleration. The significant computational overhead of Vision Foundation Models (VFMs) has spurred a large body of research focused on their acceleration for practical deployment. A recent active area has been the optimization of SAM [29] and VGGT [63], with several distinct approaches. Many works propose efficient architectures, introducing entirely new lightweight models or modifying existing ones for speed [17, 70, 85]. Another common strategy is quantization, which reduces numerical precision to speed up inference, as demonstrated by PTQ4SAM [36] and Quantized-VGGT [16]. Methods like SlimSAM [6] employ structured pruning, followed by distillation to create highly compact models. Knowledge distillation is also often used independently to transfer knowledge from a large teacher model to a smaller student [82, 87]. Finally, some methods leverage domain-specific knowledge to accelerate computationally expensive components, such as Fast-VGGT [58] which uses token merging. In comparison, accelerating large foundation models for stereo matching has been largely underexplored, leaving a substantial research gap.

3. Approach

Our approach (Fig. 3) is based on FoundationStereo [68], which consists of three key steps: feature extraction, cost filtering, and disparity refinement. Each of these steps is accelerated by a divide-and-conquer strategy, as detailed in the following subsections. We also describe our automatic data curation pipeline.

3.1. Distilling Hybrid Monocular and Stereo Priors

Given a pair of left and right images $I_l, I_r \in \mathbb{R}^{H\times W\times 3}$, the feature backbone extracts multi-level pyramid features $f_l^{(i)}, f_r^{(i)} \in \mathbb{R}^{C_i\times \frac{H}{i}\times \frac{W}{i}}$, $i \in \{4, 8, 16, 32\}$ for the subsequent cost volume construction and aggregation. To compute such features, FoundationStereo [68] combines

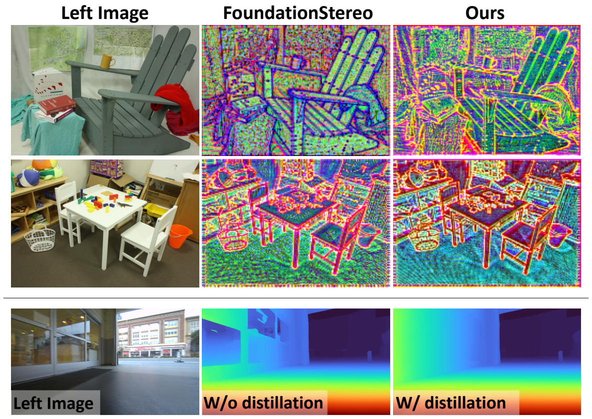

Figure 4. **Top:** Distilling the hybrid monocular and stereo priors from FoundationStereo [68] into a unified single backbone captures similar high-frequency edges and relative depth—while significantly reducing computational cost. **Bottom:** Distillation enhances robustness to translucency, which is challenging to traditional stereo matching.

DepthAnything V2 [78] with a side-tuning CNN. The former provides rich monocular priors learned from large-scale internet data, and the latter adapts the monocular features for the binocular stereo setup. Although such hybrid monocular and stereo feature extraction is powerful, it remains a significant computational bottleneck.

We leverage knowledge distillation to replace the dual module in FoundationStereo's backbone with a single student module. This approach was chosen because it is agnostic to architecture and allows to build upon the well-established feature backbones studied on ImageNet [10, 69]. As an alternative, we also considered model pruning, but it has two drawbacks: (1) it would require us to keep the dual module, which is constrained by the computational bottleneck of its underlying ViT [12]; and (2) any deterioration in accuracy would be difficult to recover without retraining on internet-scale imagery.

During distillation, DepthAnything V2 and side-tuning CNN modules from FoundationStereo are frozen and used to predict a multi-level feature pyramid $\bar{f}^{(i)}$, which the student model is trained to match via MSE loss. In the case of channel dimension mismatch, a linear projection layer is added. Even though the feature extractors take only a single image as input, we include both stereo images in each training batch to retain the statistical similarity. To provide a family of stereo models with different speed-accuracy trade-off, we train multiple variants of feature extractors [37, 54]. Fig. 4 visualizes examples of distilled features, showing that they capture similar high-frequency edges and relative depth.

3.2. Cost Filtering Blockwise Search

Given the unary features extracted in the previous step, the cost volume $\mathbf{V_C} \in \mathbb{R}^{C\times\frac{D}{4}\times\frac{H}{4}\times\frac{W}{4}}$ is constructed by combining the group-wise correlation and concatenation volumes (where D is the maximum disparity). To effectively scale the learning process with the abundant training data, FoundationStereo [68] uses a dual branch architecture to perform cost filtering. Specifically, a 3D hourglass architecture consisting of Axial-Planar Convolution (APC) layers effectively processes $\mathbf{V_C}$ by enlarging the kernel size over the disparity dimension without significantly increasing memory consumption. Meanwhile, a Disparity Transformer branch tokenizes $\mathbf{V_C}$ and performs multi-head self-attention to further enhance the long-range context reasoning within the 4D cost volume.

Direct pruning of the cost filtering modules yields severe performance degradation for only marginal speedup, since the channel dimension in $\mathbf{V_C}$ is already small (mostly under 100). We avoided direct knowledge distillation, because it requires manually designing the cost-filtering module alternatives, which remain less explored than feature backbones. Instead, we leverage Neural Architecture Search (NAS) [13] to automatically discover non-intuitive designs. In the following, we describe our efficient blockwise search strategy (Fig. 3 bottom).

Blockwise Candidate Construction. The cost filtering module is divided into a series of operation blocks: $\Phi_t(\mathbf{V_C}) = B_N \circ \cdots \circ B_2 \circ B_1(\mathbf{V_C})$, where N represents the total number of blocks. Within the 3D hourglass module, blocks are divided at the transition of the channel dimension, which typically corresponds to the spatial dimension change of the feature volume. We define five types of layers: (1) 3D conv layer with varying channel dimensions; (2) 3D deconv layer that doubles the spatial dimensions of the cost volume, (3) APC layer [68] that performs separate spatial and disparity convolution with different respective kernel sizes; (4) residually connected 3D conv layers, similar to ResNet [23]; and (5) feature guided volume excitation [1].

Meanwhile, the entire Disparity Transformer module is regarded as a single block consisting of a number of repeated multi-head self-attention transformer layers. We reuse the disparity attention layers as in [68] while varying the feed-forward layer dimensions, number of heads, and number of layers.

In both cases, the number of layers in each block and the intermediate channel dimension can vary, as long as (1) the entire block's running time t^s_B is faster than its teacher counterpart t^t_B and (2) the input and output channel dimension remains the same as the original block. Details of the search space can be found in the appendix.

Blockwise Distillation and Evaluation. After blockwise candidate construction, we obtain $C = C_1 \cdot C_2 \cdots C_N$ total number of possible cost filtering module candidates, where C_i denotes the number of candidates in block B_i. In practice, when $N = 8$ and $C_i = 200$, C is $200^8 \approx 10^{18}$. As a result, standard evolutionary search based NAS methods [11, 53] are not tractable, due to the extremely large computational cost. Moreover, training from scratch in the whole search space does not fully leverage the strengths of the teacher model.

To overcome these limitations, we train each block B_i independently. Specifically, B_i is treated as a standalone network and trained to mimic the teacher counterpart's output: $\left\|B_i(f_{i-1}) - \bar{B}_i(f_{i-1})\right\|_2^2$, given the feature output f_{i-1} from the previous teacher block. For the final block that predicts the initial disparity, smooth L_1 loss is computed against the ground truth. The teacher model is frozen throughout the distillation process. After distillation, a candidate block B_i^c is evaluated by replacing its counterpart at block level i in the teacher model and inferring the complete model end-to-end on a separate validation dataset. Both the relative error metric change Δm_i^c and running time change Δt_i^c that result by introducing B_i^c are measured.

Compared with standard NAS, our blockwise distillation reduces training complexity from $O(n^N)$ to $O(n)$ [32, 45], where n is the number of per-layer candidates. Furthermore, since B_i is small, the block distillation can be performed efficiently in terms of both speed and memory, allowing easy parallelization.

Combinatorial Search. The student cost filtering module is found by solving for the optimal combination of candidate blocks, which can be formulated as:

$$\min_{\mathcal{E}} \sum_{i=1}^{N} (\Delta \mathbf{m}_i)^\top \mathbf{e}_i, \quad \text{s.t.} \quad \sum_{i=1}^{N} (\Delta \mathbf{t}_i)^\top \mathbf{e}_i \leq \Delta\tau, \qquad (1)$$

where $\Delta\mathbf{m}_i$ and $\Delta\mathbf{t}_i$ denote the vector of error metric and running time changes, respectively, for all candidates at block B_i; $\mathbf{e}_i \in \mathcal{E}$ denotes the one-hot vector representing the selection operation of a candidate block at B_i; and $\Delta\tau$ denotes the runtime budget relative to the teacher model for the entire cost filtering module. Optimization is performed by Integer Linear Programming (ILP) [41, 44], using different values for τ to obtain a family of cost filtering student models with different speed-accuracy trade-off.

3.3. Refinement Pruning

Given the initial disparity map d_0 (predicted by the filtered cost volume) and the hidden feature (initialized from the context network), the ConvGRU module progressively refines the disparity map. Fig. 5 shows the dependency graph and data flow. At each iteration, ConvGRU module consumes the disparity d_{k-1}, h_{k-1} and predicts their updated values d_k, h_k, resulting in recurrent dependencies. This significant redundancy in refinement module (as shown in Sec. 4.4), motivates the use of structured pruning [14, 24, 42, 46], a simple yet effective technique and can benefit from GPU hardware acceleration techniques such as TensorRT.

Building Recurrent Dependency Graph. The first step in structured pruning is to identify the inter-dependencies between layers, since depth or channel pruning at one layer

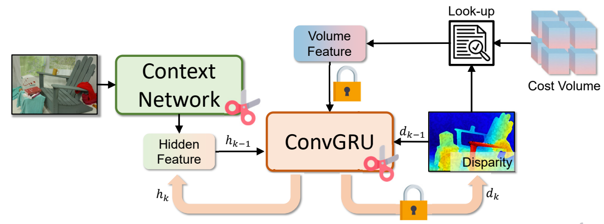

Figure 5. Recurrent dependency graph of the refinement module. ✂ denotes where pruning is performed. 🔒 denotes where channel dimension remains fixed during the pruning process.

changes the intermediate feature dimensions fed to adjacent layers. In addition to the normal adjacent layer dependencies which can be automatically constructed by tracing the computation flow [14], we introduce three more pruning constraints given the unique properties of refinement module in stereo matching: (1) within the ConvGRU module, the final layers that predict the disparity map and convex upsampling mask retain fixed output channel dimensions; (2) within the ConvGRU module, the input channel of the layer that consumes h_{k-1}, and the output channel of the layer that outputs h_k, are inter-dependent and thus jointly pruned; and (3) the motion encoder that consumes the indexed volume feature retains a fixed input channel dimension.

Pruning and Retraining. To identify which layers or channels to remove, we evaluate their importance using first-order Taylor expansion [43]. Specifically, inputs are feed forward to the complete teacher model [68] end-to-end with multiple refinement iterations, and accumulate gradients for the refinement module. The importance of each parameter in the refinement module is ranked globally, and the least important α parameters are pruned, where $\alpha \in (0, 1)$ is the pruning ratio. We also explored isomorphic pruning strategy [15] but observed slightly degraded performance. After pruning, we retrain the refinement module end-to-end (while freezing the rest of the teacher model) to recover the performance, using the loss:

$$\mathcal{L} = \sum_{k=1}^{K} \gamma^{K-k} \left\|d_k - \bar{d}\right\|_1 + \lambda \sum_{i=1}^{L} \left\|x_i - \overline{x}_i\right\|_2^2 \qquad (2)$$

where x_i and $\overline{x}_i$ are the per-layer latent features (student and teacher, respectively) from each of the L layers; $\overline{d}$ is the ground truth disparity; k is the iteration number; $\gamma = 0.9$ exponentially increases weights to supervise the iteratively refined disparity; and $\lambda = 0.1$ weighs the distillation objective. The initial disparity supervision is excluded since it is not affected by the refinement module.

3.4. Pseudo-Labeling on In-the-Wild Data

Real-world data offers greater diversity and realism than synthetic data. However, obtaining real stereo images with ground-truth metric depth annotation is notoriously difficult. To address this challenge, we propose an automatic

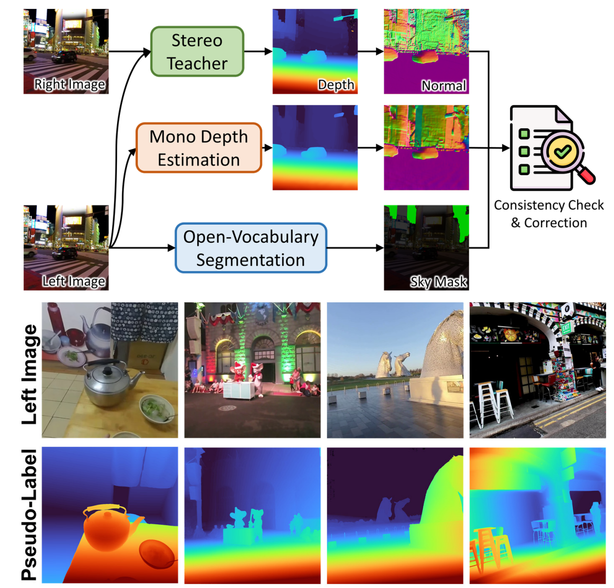

Figure 6. **Top:** Pseudo-labeling pipeline on in-the-wild internet stereo data. **Bottom:** Visualization of our generated pseudo-labels.

data curation pipeline to generate pseudo-labels on internet-scale stereo images. As shown in Fig. 6, given a rectified stereo pair from Stereo4D [26], the teacher model [68] produces a disparity map for the left image. To identify the imperfect predictions which can mislead the subsequent training process for the student model, we also feed left image to a monocular depth estimator [48] to obtain a corresponding depth map. Both the disparity map and monocular depth are further converted into normal maps via 3D unprojection and Sobel operator using the same set of camera parameters provided by [26]. To assess local geometric consistency, we compute the per-pixel cosine similarity between the two normal maps, which is thresholded to produce a consistency mask. Stereo samples with insufficient agreement are discarded. Due to the uniqueness of sky regions (which have infinite depth and are underrepresented in common synthetic datasets used for training), the similarity computation excludes the sky regions, which are detected by open-vocabulary segmentation models [52, 77].

The remaining stereo disparity maps become the final pseudo-labels, where the sky regions are set to zero disparity. The consistency mask can be optionally used to determine the supervision pixels. We subsample the videos temporally by a stride of 10, yielding 1.4M suitable stereo pairs in total. In contrast to directly comparing in the depth or disparity space, our proposed normal consistency check is more robust to extremely diverse depth ranges or noisy predictions on in-the-wild images. This automatically pseudo-labeled data is included in our final training of student models. Such output-space distillation complements the feature-based distillation performed in previous steps.

4. Experiments

4.1. Implementation Details

Our Fast-FoundationStereo was trained on the same mixed datasets as FoundationStereo [68], as well as the pseudo-labeled real data (Sec. 3.4). For deployment, our framework provides the flexibility to assemble different candidates from each step to compose the final stereo matching model (examples are shown in Fig. 2). The final model is then trained end-to-end. Once trained, the fixed set of weights for each candidate model are used to perform zero-shot inference on unseen data. Unless otherwise mentioned, we use 8 refinement iterations and 192 as the maximum disparity for constructing the cost volume. For evaluation, the disparity range is not constrained.

4.2. Benchmark Datasets and Metric

Datasets. Four common public datasets were used for evaluation: Middlebury [55] consists of indoor stereo image pairs with high-quality ground-truth disparity captured via structured light. ETH3D [56] provides grayscale stereo image pairs covering both indoor and outdoor scenarios. and KITTI 2012 [18] and KITTI 2015 [40] feature real-world driving scenes, where sparse ground-truth disparity maps are derived from LIDAR sensors. Booster [50] features a large variety of translucent and specular scenes and is used to evaluate the robustness to non-Lambertian surfaces.

Metrics. "BP-X" computes the percentage of pixels where the disparity error is larger than X pixels. "D1", commonly used on KITTI [18, 40], computes the percentage of pixels whose disparity error is larger than 3 pixels and 5% of the ground-truth disparity. Results are evaluated on non-occluded regions.

4.3. Zero-Shot Generalization Comparison

Quantitative Comparison. Comparison of zero-shot generalization on public datasets is shown in Table 1. These datasets are unseen to all the evaluated methods. For cost-filtering based methods that support dynamic maximum disparity configuration, 416 is used on Middlebury-H for the best performance; otherwise their default setting is used for other lower resolution datasets. Existing real-time methods are usually not targeted for zero-shot generalization, and are thus mainly trained on SceneFlow [39]. For those competitive ones with publicly released training code [22, 74], we additionally train them on the exact same datasets as ours (including our proposed pseudo-labels). The inference runtimes for all methods are profiled over Middlebury-Q (similar to typical resolution for real-time robotic applications) on the same hardware with NVIDIA 3090 GPU.

As can be observed, our Fast-FoundationStereo outperforms other real-time models by a significant margin across the board, even when they are trained on the exact same

Method	Middlebury-H BP-1	BP-2	BP-3	Middlebury-Q BP-1	BP-2	BP-3	ETH3D BP-1	BP-2	BP-3	KITTI 2012 BP-1	BP-2	BP-3	D1	KITTI 2015 BP-1	BP-2	BP-3	D1	Runtime (ms)
StereoAnywhere [2]	9.67	4.75	2.45	8.00	3.25	2.10	1.43	0.61	0.41	11.66	4.67	3.52	2.81	21.81	6.72	3.79	3.52	427
DEFOM-Stereo [25]	8.84	3.76	2.46	7.51	3.50	2.22	2.16	1.03	0.78	13.10	5.32	3.39	3.12	23.92	8.12	4.76	4.58	371
MonSter [7]	9.33	4.24	2.69	7.08	3.19	1.94	0.99	0.46	0.28	9.58	4.39	2.99	2.84	20.61	6.44	3.59	3.41	336
Zero-RAFT-Stereo [66]	8.48	4.68	3.32	8.15	4.42	3.26	2.14	1.17	0.85	9.15	4.17	2.93	2.76	21.13	7.43	4.67	4.48	164
FoundationStereo [68]	**2.49**	**1.10**	**0.88**	**2.64**	**1.30**	**0.96**	**0.50**	**0.30**	**0.24**	**8.16**	**3.50**	**2.47**	**2.30**	**18.65**	**5.20**	**2.95**	**2.80**	496
IINet* [33]	25.88	16.69	13.03	24.90	14.90	10.42	21.21	12.55	9.19	33.12	15.71	9.72	9.30	36.22	14.16	7.86	7.58	30
LightStereo-L* [22]	37.49	23.76	18.48	30.08	17.75	13.11	45.46	37.21	34.15	42.42	22.39	14.49	13.98	40.56	19.10	12.35	12.08	30
LightStereo-L [22]	22.64	12.55	9.07	16.34	7.70	4.99	16.34	7.70	4.99	17.59	6.71	3.97	3.73	27.66	9.07	4.75	4.51	30
RT-IGEV* [74]	16.95	11.52	9.40	14.02	7.71	5.52	5.66	2.81	2.26	16.70	7.28	4.85	4.54	25.89	9.89	6.19	6.00	45
RT-IGEV [74]	12.75	7.82	5.73	11.28	5.59	3.77	5.05	2.78	1.63	11.38	5.05	3.44	3.25	22.70	7.32	4.24	4.00	45
BANet-2D* [75]	43.78	28.45	22.28	37.33	23.51	18.62	44.89	37.87	35.81	42.92	22.45	14.48	13.88	42.92	22.45	14.48	13.88	29
BANet-3D* [75]	44.90	30.10	24.17	32.02	18.69	13.70	29.27	20.99	18.55	45.43	25.20	17.07	16.59	50.26	26.38	17.13	16.87	26
Ours	**4.80**	**2.20**	**1.60**	**4.51**	**2.12**	**1.57**	**1.22**	**0.62**	**0.50**	**8.52**	**3.61**	**2.50**	**2.35**	**19.62**	**5.78**	**3.43**	**3.25**	49 (21)

Table 1. Zero-shot generalization on public datasets. Methods are grouped based on their feasibility for real-time application. *Denotes methods trained only on SceneFlow [39]; others are trained on large-scale combined datasets, or they leverage foundation models pretrained on large-scale datasets. Bold indicates the best method within each group; note that ours is also second-best in each column. The number in parentheses is the runtime using TensorRT.

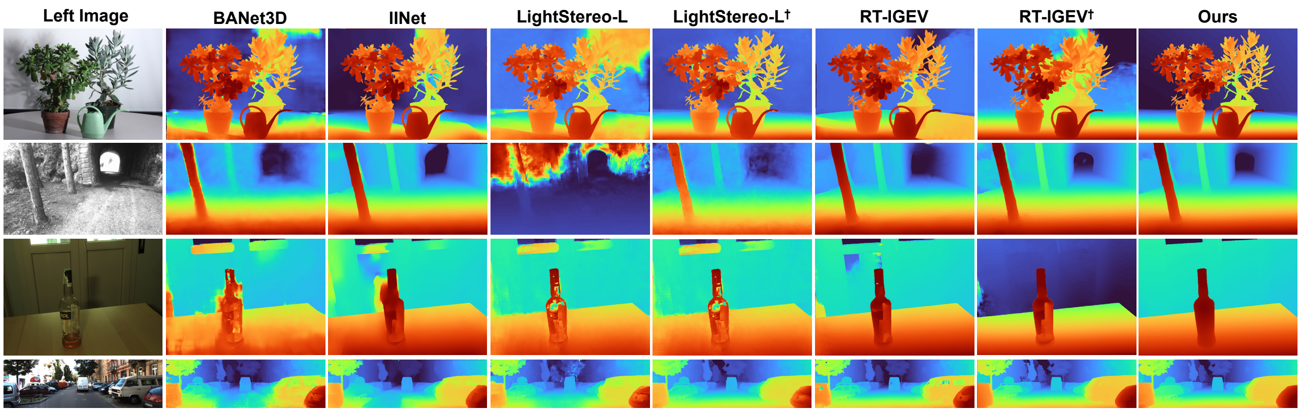

Figure 7. Qualitative results of real-time methods on Middlebury, ETH3D, Booster and KITTI-2015 datasets (top to bottom). Results are obtained by zero-shot inference without training on any split of the target datasets. †Indicates methods trained on the exact same datasets as ours, including our proposed pseudo-labels. More results on in-the-wild images can be found in the appendix.

Methods	BP-2	BP-4	BP-6	BP-8	EPE (px)	Real-time
RAFT-Stereo [34]	17.84	13.06	10.76	9.24	3.59	✗
PSMNet [4]	34.47	24.83	20.46	17.77	7.26	✗
GMStereo [76]	32.44	22.52	17.96	15.02	5.29	✗
PCVNet [81]	22.63	16.51	13.81	12.08	4.70	✗
DLNR [86]	18.56	14.55	12.61	11.22	3.97	✗
Selective-IGEV [65]	18.52	14.24	12.14	10.77	4.38	✗
IGEV [73]	16.90	13.23	11.40	10.20	3.94	✗
NMRF [20]	27.08	19.06	15.43	13.21	5.02	✗
StereoAnywhere [2]	9.01	5.40	4.12	3.34	1.21	✗
FoundationStereo [68]	**5.18**	**4.07**	**2.91**	**2.59**	**1.13**	✗
RT-IGEV [74]	23.09	16.86	14.10	12.47	5.03	✓
RT-IGEV† [74]	18.19	13.39	11.37	10.16	4.20	✓
Ours	**6.61**	**4.62**	**3.91**	**3.49**	**1.54**	✓

Table 2. Zero-shot generalization on non-Lambertian surfaces, evaluated on the Booster-Q dataset [50]. †Denotes training on the exact same datasets as ours (including our proposed pseudo-labels).

datasets, including our proposed pseudo-labels. Moreover, our model achieves comparable or even better results than most of the computationally expensive models, including Zero-RAFT-Stereo [66] which leverages additional synthesized training data via multiple large foundation models. Compared to FoundationStereo [68], our method runs more than 10 times faster with only a modest increase in error.

Robustness to Non-Lambertian Surfaces. Table 2 shows zero-shot generalization results on Booster-Q dataset [50]. Numbers are from the StereoAnywhere paper [2]; FoundationStereo [68] and the most competitive real-time model RT-IGEV [74] are also included for comparison.

Qualitative Comparison. Visualizations of zero-shot inference are demonstrated in Figs. 1 and 7. The stereo images represent diverse challenges including textureless regions, transparency, specular highlights, complex illuminations, varying depth ranges, viewing perspectives and both indoor / outdoor scenarios. Despite these challenges, our model significantly outperforms other real-time models. It even achieves comparable or sometimes more favorable results than computationally expensive generalizable models.

4.4. Framework Analysis

Effects of Backbone Distillation. Table 3 shows an ablation study on no distillation (feature backbone weights pretrained only on ImageNet [10]) and different distillation losses. By distilling from hybrid monocular and stereo priors from the teacher model, the feature backbone generally enhances zero-shot generalization. The effectiveness is also demonstrated in Fig. 4, where the translucent glass door challenges the traditional stereo matching process without distillation.

Effects of Cost Filtering Blockwise Search. Our blockwise search strategy significantly reduces training complex-

Variants	Midd.-H BP-2	ETH3D BP-1	KITTI-12 D1	KITTI-15 D1
No Distillation	2.87	2.11	2.67	4.32
Cosine Similarity	2.29	**1.19**	2.39	3.31
MSE (Ours)	**2.20**	1.22	**2.35**	**3.25**

Table 3. Ablations on feature backbone distillation strategies.

ity from $O(n^N)$ to $O(n)$. However, it leverages a surrogate objective, Eq. (1), which accumulates the impacts of perturbing each local block. This is a proxy to the actual performance of a candidate model, which would otherwise require training the full assembled cost filtering module with the remaining parts of the network end-to-end for evaluation. In order to verify if such proxy is effective, we compare our searched cost filtering candidate, based on Eq. (1), against randomly assembled candidates under the same latency constraint $\Delta\tau$. All cost filtering module candidates are trained end-to-end (with the remaining parts from the teacher model), followed by zero-shot evaluation. For each $\Delta\tau$, 10 random candidate models are sampled (note that training each of them end-to-end is expensive). As shown in Fig. 8: (1) as the latency constraint relaxes ($\Delta\tau$ increases), our architecture search can successfully find better performing candidate models; (2) under varying $\Delta\tau$, the searched candidate consistently outperforms randomly assembled candidates; and (3) as $\Delta\tau$ decreases, some randomly assembled candidates yield substantial performance degradation, highlighting the importance of network design under tight latency constraint.

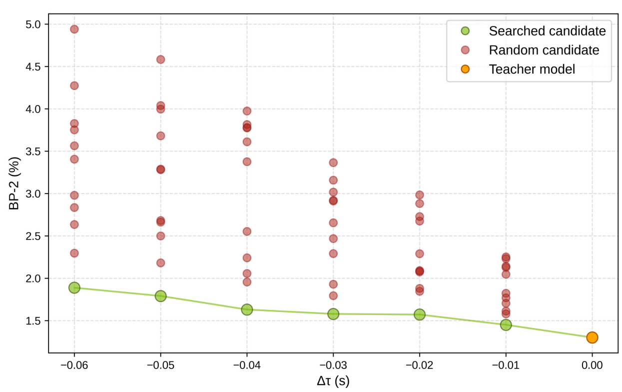

Figure 8. Effects of blockwise architecture search for cost filtering module under varying latency budget $\Delta\tau$, evaluated on Middlebury-Q.

Effects of Pruning Ratio. Fig. 9 demonstrates how the pruning ratio affects the prediction accuracy on Middlebury-Q dataset and runtime under one refinement iteration. While aggressive pruning dramatically degrades the prediction accuracy, it can be effectively recovered through retraining with Eq. (2), implying large redundancy in the original refinement module.

Effects of Pseudo-Labeling. Table 4 ablates on training with pseudo-labeled data and their zero-shot generalization results on public datasets under commonly used metrics.

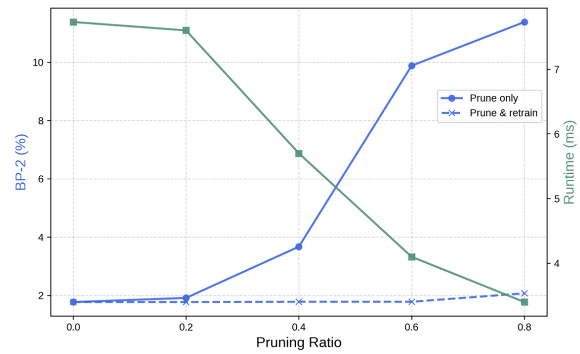

Figure 9. Effects of pruning ratio for accuracy and speed.

We also ablate on a few competitive real-time methods with this data. Pseudo-labeling enhances generalization performance consistently for all methods. The elevation is even more significant for those methods [22, 74] that were previously trained only on SceneFlow.

Method	Midd.-H BP-2	ETH3D BP-1	KITTI-12 D1	KITTI-15 D1
RT-IGEV [74]	11.52 (8.69)	5.66 (5.12)	4.54 (3.55)	6.00 (4.40)
LightStereo-L [22]	23.76 (18.41)	45.46 (21.12)	13.98 (5.27)	12.08 (7.63)
Ours	2.53 (2.20)	1.31 (1.22)	2.44 (2.35)	3.48 (3.25)

Table 4. Results on in-the-wild data without (and with) pseudo-labeling.

Runtime Analysis. Fig. 10 shows the detailed runtime decomposition between FoundationStereo [68] and our slowest model from Fig. 2. Results are profiled on the same hardware (NVIDIA 3090 GPU). Each of the three essential steps are accelerated by a large margin, leading to a total runtime performance boost of more than 10×.

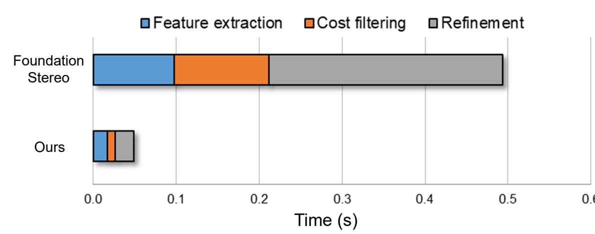

Figure 10. Runtime decomposition.

5. Conclusion

Fast-FoundationStereo bridges the gap between zero-shot generalization and real-time performance. Through a principled divide-and-conquer acceleration strategy, we demonstrate that the computational bottlenecks of foundation stereo models can be systematically addressed without sacrificing robustness. Our extensive evaluations confirm that Fast-FoundationStereo not only establishes a new state-of-the-art among real-time methods by a substantial margin, but it also competes favorably with computationally intensive generalizable models. For future work, exploring quantization techniques offers an orthogonal avenue to further enhance inference speed, potentially enabling deployment on even more resource-constrained edge devices.

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. 4, 1, 2

[2] Luca Bartolomei, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Stereo anywhere: Robust zero-shot deep stereo matching even where either stereo or mono fail. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1013–1027, 2025. 1, 2, 7

[3] Jiaxuan Cai, Zhi Qi, Keqi Fu, Xulong Shi, Zan Li, Xuanyu Liu, and Hao Liu. Pbcstereo: A compressed stereo network with pure binary convolutional operations. In *Proceedings of the Asian Conference on Computer Vision*, pages 4378–4394, 2022. 2

[4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. 2018. 7

[5] Tianyu Chang, Xun Yang, Tianzhu Zhang, and Meng Wang. Domain generalized stereo matching via hierarchical visual transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9559–9568, 2023. 2

[6] Zigeng Chen, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Slimsam: 0.1% data makes segment anything slim. *Advances in Neural Information Processing Systems*, 37: 39434–39461, 2024. 3

[7] Junda Cheng, Longliang Liu, Gangwei Xu, Xianqi Wang, Zhaoxing Zhang, Yong Deng, Jinliang Zang, Yurui Chen, Zhipeng Cai, and Xin Yang. Monster: Marry monodepth to stereo unleashes power. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6273–6282, 2025. 1, 2, 7

[8] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in neural information processing systems*, 33:22158–22169, 2020. 2

[9] WeiQin Chuah, Ruwan Tennakoon, Reza Hoseinnezhad, Alireza Bab-Hadiashar, and David Suter. ITSA: An information-theoretic approach to automatic shortcut avoidance and domain generalization in stereo matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13022–13032, 2022. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 4, 7

[11] Travis Desell. Large scale evolution of convolutional neural networks using volunteer computing. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 127–128, 2017. 4

[12] 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 (ICLR)*, 2021. 4

[13] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019. 4

[14] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16091–16101, 2023. 5

[15] Gongfan Fang, Xinyin Ma, Michael Bi Mi, and Xinchao Wang. Isomorphic pruning for vision models. In *European Conference on Computer Vision*, pages 232–250. Springer, 2024. 5

[16] Weilun Feng, Haotong Qin, Mingqiang Wu, Chuanguang Yang, Yuqi Li, Xiangqi Li, Zhulin An, Libo Huang, Yulun Zhang, Michele Magno, et al. Quantized visual geometry grounded transformer. *arXiv preprint arXiv:2509.21302*, 2025. 3

[17] Jianhai Fu, Yuanjie Yu, Ningchuan Li, Yi Zhang, Qichao Chen, Jianping Xiong, Jun Yin, and Zhiyu Xiang. Lite-sam is actually what you need for segment everything. In *European conference on computer vision*. Springer, 2024. 3

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 6

[19] 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. 2

[20] Tongfan Guan, Chen Wang, and Yun-Hui Liu. Neural markov random field for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2024. 7

[21] 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 (CVPR)*, pages 3273–3282, 2019. 2

[22] Xianda Guo, Chenming Zhang, Youmin Zhang, Wenzhao Zheng, Dujun Nie, Matteo Poggi, and Long Chen. Light-stereo: Channel boost is all you need for efficient 2d cost aggregation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8738–8744. IEEE, 2025. 2, 6, 7, 8

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4, 1

[24] Yang He and Lingao Xiao. Structured pruning for deep convolutional neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(5):2900–2919, 2023. 5

[25] Hualie Jiang, Zhiqiang Lou, Laiyan Ding, Rui Xu, Minglang Tan, Wenjie Jiang, and Rui Huang. Defom-stereo: Depth foundation model based stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21857–21867, 2025. 2, 7

[26] Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4D: Learning how things move in 3d from internet stereo videos. In *CVPR*, 2025. 6

[27] 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. 2

[28] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 5

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 3

[30] Chen Kong, James Fort, Aria Kang, Jonathan Wittmer, Simon Green, Tianwei Shen, Yipu Zhao, Cheng Peng, Gustavo Solaira, Andrew Berkovich, et al. Aria gen 2 pilot dataset. *arXiv preprint arXiv:2510.16134*, 2025. 1

[31] Taeyeop Lee, Gyuree Kang, Bowen Wen, Youngho Kim, Seunghyeok Back, In So Kweon, David Hyunchul Shim, and Kuk-Jin Yoon. Delta: Demonstration and language-guided novel transparent object manipulation. *arXiv preprint arXiv:2510.05662*, 2025. 1

[32] Changlin Li, Jiefeng Peng, Liuchun Yuan, Guangrun Wang, Xiaodan Liang, Liang Lin, and Xiaojun Chang. Block-wisely supervised neural architecture search with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1989–1998, 2020. 5

[33] Ximeng Li, Chen Zhang, Wanjuan Su, and Wenbing Tao. Iinet: Implicit intra-inter information fusion for real-time stereo matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3225–3233, 2024. 2, 7

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

[35] Biyang Liu, Huimin Yu, and Guodong Qi. GraftNet: Towards domain generalized stereo matching with a broad-spectrum and task-oriented feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13012–13021, 2022. 2

[36] Chengtao Lv, Hong Chen, Jinyang Guo, Yifu Ding, and Xianglong Liu. Ptq4sam: Post-training quantization for segment anything. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 15941–15951, 2024. 3

[37] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. EdgeNeXt: Efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–20, 2022. 4

[38] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976. 1

[39] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 6, 7

[40] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 6

[41] Stuart Mitchell, Paul O'Sullivan, and Christopher D'Andrea. PuLP: A Linear Programming Toolkit for Python. *Optimization Online*, 2011. 5

[42] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *ICLR*, 2017. 5

[43] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019. 5

[44] Pavlo Molchanov, Jimmy Hall, Hongxu Yin, Jan Kautz, Nicolo Fusi, and Arash Vahdat. Lana: latency aware network acceleration. In *European Conference on Computer Vision*, pages 137–156. Springer, 2022. 5

[45] Bert Moons, Parham Noorzad, Andrii Skliar, Giovanni Mariani, Dushyant Mehta, Chris Lott, and Tijmen Blankevoort. Distilling optimal neural networks: Rapid search in diverse spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12229–12238, 2021. 5

[46] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. *Advances in Neural Information Processing Systems*, 37:41076–41102, 2024. 5

[47] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 1

[48] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025. 6

[49] Rafia Rahim, Faranak Shamsafar, and Andreas Zell. Separable convolutions for optimizing 3d stereo networks. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3208–3212. IEEE, 2021. 2

[50] Pierluigi Zama Ramirez, Alex Costanzino, Fabio Tosi, Matteo Poggi, Samuele Salti, Stefano Mattoccia, and Luigi Di Stefano. Booster: A benchmark for depth from images of specular and transparent surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2023. 6, 7

[51] Zhibo Rao, Bangshu Xiong, Mingyi He, Yuchao Dai, Renjie He, Zhelun Shen, and Xing Li. Masked representation learning for domain generalized stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5435–5444, 2023. 2

[52] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 6

[53] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, pages 4780–4789, 2019. 4

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 4

[55] 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 Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*, pages 31–42. Springer, 2014. 6

[56] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3260–3269, 2017. 6

[57] Faranak Shamsafar, Samuel Woerz, Rafia Rahim, and Andreas Zell. Mobilestereonet: Towards lightweight deep networks for stereo matching. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pages 2417–2426, 2022. 2

[58] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025. 3

[59] 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. 2

[60] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 1

[61] 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

[62] Fabio Tosi, Luca Bartolomei, and Matteo Poggi. A survey on deep stereo matching in the twenties. *International Journal of Computer Vision (IJCV)*, 2025. 1

[63] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 3

[64] Qiang Wang, Shaohuai Shi, Kaiyong Zhao, and Xiaowen Chu. Easnet: Searching elastic and accurate network architecture for stereo matching. In *European Conference on Computer Vision*, pages 437–453. Springer, 2022. 2

[65] Xianqi Wang, Gangwei Xu, Hao Jia, and Xin Yang. Selective-Stereo: Adaptive frequency information selection for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19701–19710, 2024. 7

[66] Xianqi Wang, Hao Yang, Gangwei Xu, Junda Cheng, Min Lin, Yong Deng, Jinliang Zang, Yurui Chen, and Xin Yang. Zerostereo: Zero-shot stereo matching from single images. *ICCV*, 2025. 1, 2, 7

[67] Yingqian Wang, Longguang Wang, Jungang Yang, Wei An, and Yulan Guo. Flickr1024: A large-scale dataset for stereo image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 4

[68] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5249–5260, 2025. 1, 2, 3, 4, 5, 6, 7, 8

[69] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 4

[70] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. Efficientsam: Leveraged masked image pretraining for efficient segment anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16111–16121, 2024. 3

[71] 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. 2

[72] 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. 2

[73] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023. 7

[74] Gangwei Xu, Xianqi Wang, Zhaoxing Zhang, Junda Cheng, Chunyuan Liao, and Xin Yang. IGEV++: Iterative multi-

range geometry encoding volumes for stereo matching. *arXiv preprint arXiv:2409.00638*, 2024. 2, 6, 7, 8

[75] Gangwei Xu, Jiaxin Liu, Xianqi Wang, Junda Cheng, Yong Deng, Jinliang Zang, Yurui Chen, and Xin Yang. Banet: Bilateral aggregation network for mobile stereo matching. *ICCV*, 2025. 2, 7

[76] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2023. 7

[77] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2955–2966, 2023. 6

[78] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2024. 1, 4

[79] Chengtang Yao, Lidong Yu, Zhidan Liu, Jiaxi Zeng, Yuwei Wu, and Yunde Jia. Diving into the fusion of monocular priors for generalized stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14887–14897, 2025. 2

[80] Kyle Yee and Ayan Chakrabarti. Fast deep stereo with 2d convolutional processing of cost signatures. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 183–191, 2020. 2

[81] Jiaxi Zeng, Chengtang Yao, Lidong Yu, Yuwei Wu, and Yunde Jia. Parameterized cost volume for stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18347–18357, 2023. 2, 7

[82] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 3

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

[84] Yongjian Zhang, Longguang Wang, Kunhong Li, Yun Wang, and Yulan Guo. Learning representations from foundation models for domain generalized stereo matching. In *European Conference on Computer Vision*, pages 146–162. Springer, 2024. 2

[85] Zhuoyang Zhang, Han Cai, and Song Han. Efficientvit-sam: Accelerated segment anything model without performance loss. In *CVPRW*, 2024. 3

[86] Haoliang Zhao, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao. High-frequency stereo matching network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[87] Chong Zhou, Xiangtai Li, Chen Change Loy, and Bo Dai. Edgesam: Prompt-in-the-loop distillation for sam. *International Journal of Computer Vision*, pages 1–17, 2025. 3

[88] Jingyi Zhou, Haoyu Zhang, Jiakang Yuan, Peng Ye, Tao Chen, Hao Jiang, Meiya Chen, and Yangyang Zhang. All-in-one: Transferring vision foundation models into stereo matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10797–10805, 2025. 2