

FlowSeek: Optical Flow Made Easier with Depth Foundation Models and Motion Bases

Matteo Poggi

Fabio Tosi

University of Bologna, Italy

Project page: <https://flowseek25.github.io/>



Figure 1. **FlowSeek in Action.** State-of-the-art optical flow models struggle at generalizing across different domains, with a lack of fine details in their predictions. FlowSeek achieves superior generalization by exploiting the strong priors from depth foundation models.

Abstract

We present *FlowSeek*, a novel framework for optical flow requiring minimal hardware resources for training. *FlowSeek* marries the latest advances on the design space of optical flow networks with cutting-edge single-image depth foundation models and classical low-dimensional motion parametrization, implementing a compact, yet accurate architecture. *FlowSeek* is trained on a single consumer-grade GPU, a hardware budget about $8\times$ lower compared to most recent methods, and still achieves superior cross-dataset generalization on *Sintel Final* and *KITTI*, with a relative improvement of 10 and 15% over the previous state-of-the-art *SEA-RAFT*, as well as on *Spring* and *LayeredFlow* datasets.

1. Introduction

Optical flow [22] is one of the classical problems in computer vision, reaching almost half a century of history. It consists of estimating the 2D motion fields connecting pixels across two or multiple frames in videos, and is the foundation of several higher-level tasks such as action recognition [74], video interpolation [25, 49, 90], or 4D synthesis and reconstruction [16, 85]. In the last decade,

this field has been reshaped by the advent of deep learning [15], which has pushed the research community towards the development of end-to-end deep architectures [27–30, 60, 70, 71, 76] to replace the hand-crafted algorithms [6, 7, 10, 23, 24, 41, 43, 61, 69] developed before. Throughout the years, several design strategies have been proposed, including coarse-to-fine architectures [27–29, 60, 70], 4D convolutions [77, 91] or recurrent neural networks [76], with the latter becoming the dominant [26, 50, 67, 68, 81, 97] following RAFT [76].

Associated with architectural design, two other factors are paramount for the development of accurate flow models: i) having access to vast amounts of training data, annotated with high-quality flow ground-truth labels, and ii) the availability of a substantial hardware budget, that is, multiple high-end GPUs, for training complex architectures and pushing their performance to their best [26, 50, 67, 68, 81, 97]. Dependence on both is common to any computer vision task based on deep learning, often making the difference between a suboptimal method or a state-of-the-art solution. As a result, academia and industry are racing to increase both the availability of training data and the number of GPUs to train deep models – e.g., FlowFormer [26] and GMFlow [87] trained on $4\times$ V100 GPUs, SEA-RAFT [81] on $8\times$ 3090 RTX GPUs. We feel the reliance

on brute force to push the bar of progress can be a shortcut for low-hanging fruits, yet may preclude pursuing further methodological advances. Furthermore, in the long run, over-reliance on hardware capabilities can make research inaccessible to groups that do not have sufficient budgets to compete with larger laboratories or companies.

Therefore, we argue that progress can be pursued even with a lower hardware budget, as evidenced by recent news from adjacent research fields such as natural language processing, where the DeepSeek model [18] proved unprecedented performance after being trained on a fraction of the hardware budget used by competitors [56]. We believe that similar stories can be written in computer vision by building on existing vision foundation models carefully repurposed for different tasks, aiming to *recycle* the effort to training them rather than training a new solution from scratch, still with prohibitive hardware requirements. Recent examples involve the fine-tuning of pre-trained image generation models [63] for tasks such as depth estimation [39] or optical flow itself [65], or embedding single-image depth foundation models [92] within deep stereo [2, 11, 35, 83] or multi-view stereo [31] architectures. We believe a path similar to the latter can be taken for optical flow.

In this paper, we introduce **FlowSeek**, a novel deep architecture for optical flow estimation designed at the intersection of three worlds. Indeed, FlowSeek harmonizes i) recent advances in the design space of optical flow networks [81], ii) cutting-edge depth foundation models [93], pre-trained on millions-scale datasets, and iii) low-dimensional motion parametrization [21] from the classical computer vision literature. By connecting these components at the opposites of a 30-year time span, FlowSeek implements a compact solution for optical flow that can be trained on a single consumer-scale GPU, yet achieve state-of-the-art accuracy and fine-grained details – as shown in Fig. 1, with models trained on TartanAir [80], FlyingChairs [15], FlyingThings3D [52] and tested on Spring [53].

Our contributions can be summarized as follows:

- We introduce FlowSeek, the first optical flow model that integrates a pre-trained depth foundation model.
- We explore different design strategies to best exploit the prior knowledge of the foundation model for the optical flow estimation task.
- We develop several variants of FlowSeek, implementing different trade-offs between accuracy and efficiency, yet maintaining the single-GPU requirement at training time.

2. Related Work

Optical Flow. Optical flow estimation has evolved from classical approaches that treated it as an optimization problem [3, 22, 95] to modern deep learning methods. The field was revolutionized by FlowNet [15, 30], which first formulated flow estimation as a supervised learning prob-

lem and later introduced stacked architectures to improve accuracy. SpyNet [60] combined classical spatial pyramid concepts with deep learning, while PWC-Net [70] advanced the field by incorporating warping and cost volumes. Later work focused on efficiency through lightweight architectures [28, 29] and improved volumetric processing [91]. In particular, PWC-Net+ [71] demonstrated the importance of training protocols beyond architectural choices.

A major breakthrough came with RAFT [76], which established a new paradigm through iterative refinement and multi-scale cost volumes, with notable follow-up work [73] disentangling the contributions of architecture and training. This has led to many architectural advances: SEA-RAFT [81] enhanced accuracy through mixture-of-Laplace loss and rigid-motion pre-training, while several approaches explored efficient architectures [13, 55] and high-resolution estimation [32, 33, 98]. The emergence of transformer architectures led to further significant improvements, with FlowFormer [26], FlowFormer++ [67], CRAFT [68], GMFlowNet [97], and efficient high-resolution approaches [42] leveraging various forms of attention for global context. Other notable developments have addressed specific challenges like occlusion handling and spatial affinity [36, 51, 75], while novel directions include unifying flow with stereo and depth estimation [87, 88] and leveraging geometric matching pretraining [14]. Among emerging approaches, diffusion models [65] demonstrated surprising effectiveness without task-specific designs.

While our focus is on supervised learning, the field has seen parallel evolution in other directions. Temporal information has been explored by multi-frame approaches [12, 66, 78], while data scarcity has been addressed by unsupervised learning [38, 46] and joint learning with related tasks [37, 47]. The challenge of training data has inspired various solutions, from automatic generation and unlabelled video synthesis [19, 72] to novel approaches such as single-image flow synthesis [1] and augmentation strategies [34].

Motion and Flow Bases. Classical approaches to motion estimation leverage the fact that camera motion induces six-dimensional linear subspace of possible flow fields [21]. This was leveraged by [86] who introduced a learned PCA basis for efficient sparse-to-dense flow estimation. Recent methods have expanded these concepts: [48, 94] used flow bases for unsupervised homography estimation, Bowen et al. [5] proposed learning scene representations through flow subspaces, while Safadoust and Güney [64] demonstrated their utility for multi-object discovery through depth-aware flow decomposition. We build upon this line of work by combining motion bases with modern foundation models for depth estimation to bootstrap flow estimation.

Vision Foundation Models (VFMs) for Geometry. VFMs have greatly advanced computer vision through large-scale pre-training. While models like CLIP [57] have

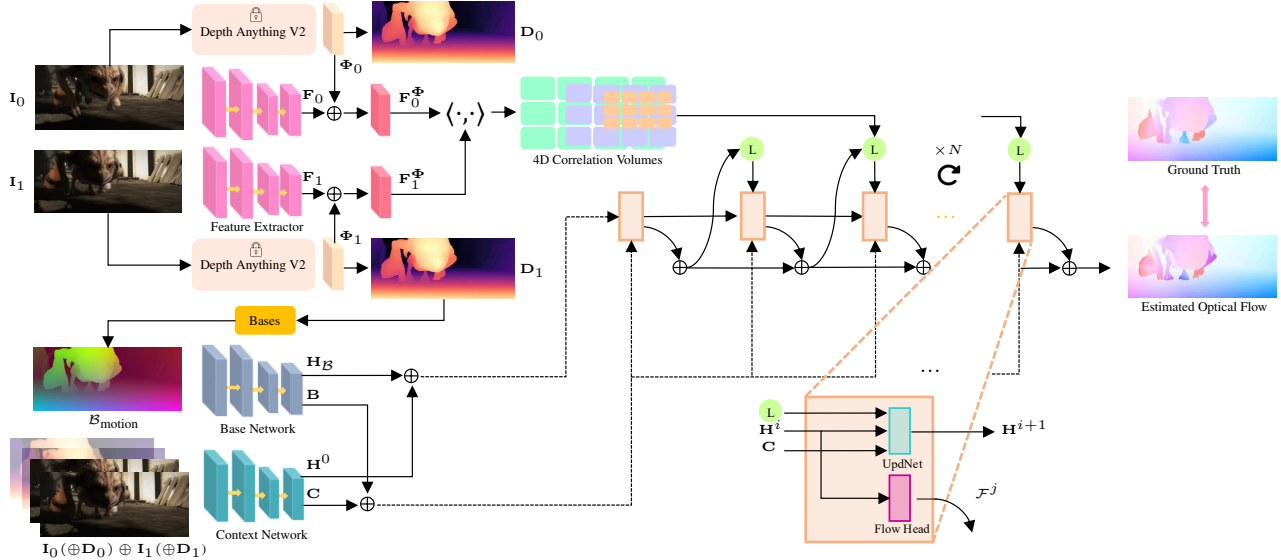


Figure 2. **Architecture Overview.** Our proposed FlowSeek architecture processes a pair of images I_0, I_1 through parallel paths: a shared-weight feature extractor produces F_0, F_1 , while a depth foundation model (e.g. Depth Anything v2 [93]) estimates depth maps D_0, D_1 and features Φ_0, Φ_1 . These are combined to obtain enriched features F_0^Φ, F_1^Φ for building 4D correlation volumes $\{V^s\}_s^S$. A Base Network extracts motion features B, H_B from geometric bases B_{motion} , while a Context Network processes images (and optionally depths) to obtain context features C, H^0 . Flow \mathcal{F}^j is iteratively refined through an UpdNet processing correlation lookups (L), hidden states H^i and context, with a FlowHead predicting updated at each step. Training relies on log-likelihood minimization over Laplacian mixture distributions.

demonstrated strong capabilities in image-level tasks, and DINO [9] in dense representation learning, their application to geometric vision problems has expanded rapidly across 3D reconstruction [79, 96], pose estimation [82], and depth prediction. In depth estimation, models like Depth Anything [92] and its successor Depth Anything v2 [93] have achieved state-of-the-art results through synthetic data training and knowledge distillation, alongside other VFMs for monocular depth estimation [4, 39, 58, 59]. Similarly, stereo matching has also benefited from VFMs, thanks to specialized adapters [2, 11, 35, 45, 83, 99] that mitigate the domain gap between pre-trained ViT features and geometric matching requirements. Despite these advances across geometric tasks, the potential of VFMs, particularly their rich semantic and geometric understanding, remains largely unexplored in the context of optical flow estimation. Our work addresses this gap by leveraging the prior knowledge infused in depth foundation models for flow estimation.

3. Method Overview

3.1. Optical Flow Backbone

FlowSeek is built on top of an optical flow backbone. Following the latest trends, we start over SEA-RAFT [81], a model assembling four basic modules, as shown in Fig. 2.

Features Extractor. The first step performed by any flow backbone involves extracting meaningful features from input images. For this purpose, either a convolutional neural network (CNN) or a Vision Transformer (ViT) can be em-

ployed. We deploy a classical CNN from the ResNet family [20] as our FeatNet, which we initialize with ImageNet pre-trained weights. This network processes both images I_0 and I_1 to generate dense feature maps F_0 and F_1 :

$$F_0 = \text{FeatNet}(I_0), \quad F_1 = \text{FeatNet}(I_1) \quad (1)$$

at $\frac{1}{8}$ of the input resolution and with K channels.

All-Pair Correlation Volume. These features are used to measure the per-pixel similarity between all possible candidate matches across the two images by building a 4D all-pair correlation volume [76]. For any pair of pixels in I_0 and I_1 , respectively at coordinates ij and uw , their correlation is computed as the dot product between the extracted features F_0 and F_1 . A correlation volume pyramid $\{V^s\}_s^S$ is then built:

$$V^s(ijw) = \sum_{k=0}^K F_0(ijk) \cdot F_1^s(uk) \quad (2)$$

with F_1^s being features F_1 downsampled through an AvgPool layer with stride s for different scales S .

Context Network. In addition to the FeatNet extracting F_0, F_1 , we deploy a further module, a ContextNet, to extract contextual features that guide the iterative flow estimation process. Specifically, following [81], this model processes both images to extract context features C , as well as the initial hidden state H^0 to bootstrap the iterative flow

estimation process:

$$\mathbf{C}, \mathbf{H}^0 = \text{ContexNet}(\mathbf{I}_0 \oplus \mathbf{I}_1) \quad (3)$$

with \oplus being the concatenation operator.

Flow Head. Optical flow $\Delta_{\mathcal{F}}$ is iteratively estimated from the hidden state \mathbf{H} . Specifically, a shallow `FlowHead` predicts an initial flow $\Delta_{\mathcal{F}}^0$ from \mathbf{H}^0 , and progressively refines it through residual updates $\Delta_{\mathcal{F}}^i$. At iteration j , the current flow estimate \mathcal{F}^j is defined as:

$$\mathcal{F}^j = \sum_{i=0}^j \Delta_{\mathcal{F}}^i = \sum_{i=0}^j \text{FlowHead}(\mathbf{H}^i) \quad (4)$$

where \mathbf{H}^i is the hidden state at iteration i . For efficiency, the `FlowHead` predicts residuals at $\frac{1}{8}$ resolution, which are then upsampled using convex upsampling [76].

Update Operator. Progressive refinement is performed by recurrently updating the hidden state \mathbf{H} with an `UpdNet`, a recurrent network processing context feature \mathbf{C} , the current hidden state \mathbf{H}^i and correlation scores retrieved from the pyramid $\{\mathbf{V}^s\}_s^S$ by means of a look-up operation based on current flow estimate $\Delta_{\mathcal{F}}$ and a radius r , predicting an updated hidden state \mathbf{H}^{i+1} :

$$\mathbf{H}^{i+1} = \text{UpdNet}(\mathbf{C}, \mathbf{H}^i, \text{LookUp}(\{\mathbf{V}^s\}_s^S, \Delta_{\mathcal{F}}^i, r)) \quad (5)$$

3.2. Depth Foundation Model

With the increasing availability of deep models trained on web-scale datasets, we aim to transfer the knowledge of such foundation models to the optical flow task. Specifically, we select candidates from the adjacent literature concerning single-image depth estimation [59, 92, 93], given the relationship between 3D geometry and induced optical flow on images – i.e., given a fixed motion of the camera or of a subject in the scene, pixels move in the images proportionally to the inverse depth of the 3D points they represent.

Given a depth foundation model `FoundModel`, we use it to predict inverse depth maps $\mathbf{D}_0, \mathbf{D}_1$ for both images $\mathbf{I}_0, \mathbf{I}_1$. During inference, we also retain the very last features produced by the decoder before depth regression, Φ_0 and Φ_1 , since they are strongly correlated with it [35, 83]:

$$\begin{aligned} \Phi_0, \mathbf{D}_0 &= \text{FoundModel}(\mathbf{I}_0), \\ \Phi_1, \mathbf{D}_1 &= \text{FoundModel}(\mathbf{I}_1) \end{aligned} \quad (6)$$

Then, we enrich the original features $\mathbf{F}_0, \mathbf{F}_1$ extracted by the flow backbone by concatenating them with those obtained from the depth foundation model. This is carried out after processing the latter with a shallow `BottNeck` network [83], composed of three 3×3 convolutional layers with stride 2 to downsample resolution to $\frac{1}{8}$ – the same resolution at which the flow backbone returns features $\mathbf{F}_0, \mathbf{F}_1$:

$$\begin{aligned} \mathbf{F}_0^\Phi &= \text{FeatNet}(\mathbf{I}_0) \oplus \text{BottNeck}(\Phi_0), \\ \mathbf{F}_1^\Phi &= \text{FeatNet}(\mathbf{I}_1) \oplus \text{BottNeck}(\Phi_1) \end{aligned} \quad (7)$$

Enriched features $\mathbf{F}_0^\Phi, \mathbf{F}_1^\Phi$ are then used to build the correlation volumes pyramid. Concurrently, $\mathbf{D}_0, \mathbf{D}_1$ can also be forwarded to the `ContextNet` together with the images to extract stronger contextual and hidden state features:

$$\mathbf{C}, \mathbf{H}^0 = \text{ContexNet}(\mathbf{I}_0 \oplus \mathbf{D}_0 \oplus \mathbf{I}_1 \oplus \mathbf{D}_1) \quad (8)$$

3.3. Low-Dimensional Motion Priors

We choose to cast the foundation model priors into a form more suitable for our model. Specifically, it is well established [21] that for a static scene with known depth, the space of possible optical flow fields can be reduced to a linear combination of six basis vectors, corresponding to the six degrees of freedom in 3D motion:

$$\mathcal{B}_{\text{motion}} = \{\Delta_{\mathbf{T}x}, \Delta_{\mathbf{T}y}, \Delta_{\mathbf{T}z}, \Delta_{\mathbf{R}x}, \Delta_{\mathbf{R}y}, \Delta_{\mathbf{R}z}\} \quad (9)$$

These six bases are categorized into two types: three model the translational component of motion and depend on inverse depth \mathbf{D}_0 , while the other three represent rotational components. The six bases are defined as:

$$\begin{aligned} \Delta_{\mathbf{T}x} &= \begin{bmatrix} f_x \mathbf{D}_0 \\ 0 \end{bmatrix}, & \Delta_{\mathbf{R}x} &= \begin{bmatrix} f_y^{-1} \bar{\mathbf{U}} \bar{\mathbf{V}} \\ f_y + f_y^{-1} \bar{\mathbf{V}}^2 \end{bmatrix} \\ \Delta_{\mathbf{T}y} &= \begin{bmatrix} 0 \\ f_y \mathbf{D}_0 \end{bmatrix}, & \Delta_{\mathbf{R}y} &= \begin{bmatrix} f_x + f_x^{-1} \bar{\mathbf{U}}^2 \\ f_x^{-1} \bar{\mathbf{U}} \bar{\mathbf{V}} \end{bmatrix} \\ \Delta_{\mathbf{T}z} &= \begin{bmatrix} -\bar{\mathbf{U}} \mathbf{D}_0 \\ -\bar{\mathbf{V}} \mathbf{D}_0 \end{bmatrix}, & \Delta_{\mathbf{R}z} &= \begin{bmatrix} f_x f_y^{-1} \bar{\mathbf{V}} \\ -f_y f_x^{-1} \bar{\mathbf{U}} \end{bmatrix} \end{aligned} \quad (10)$$

where f_x, f_y represent the camera focal length, and $\bar{\mathbf{U}}, \bar{\mathbf{V}}$ are pixel coordinates grids normalized according to the principal point (c_x, c_y) , which we can assume to be at the center of the image. This formulation, however, would require explicit knowledge of camera focal length. As the bases are combined linearly, we can arbitrarily scale them and eliminate the focal length requirement [5, 64]. Specifically, by assuming $f_x = f_y$, we can remove the focal length from basis $\Delta_{\mathbf{R}z}$, while we can split $\Delta_{\mathbf{R}x}$ and $\Delta_{\mathbf{R}y}$ and redefine them as a linear combination of two sub-bases:

$$\begin{aligned} \Delta_{\mathbf{R}^{1x}} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, & \Delta_{\mathbf{R}^{2x}} &= \begin{bmatrix} \bar{\mathbf{U}} \bar{\mathbf{V}} \\ \bar{\mathbf{V}}^2 \end{bmatrix} \\ \Delta_{\mathbf{R}^{1y}} &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & \Delta_{\mathbf{R}^{2y}} &= \begin{bmatrix} \bar{\mathbf{U}}^2 \\ \bar{\mathbf{U}} \bar{\mathbf{V}} \end{bmatrix} \end{aligned} \quad (11)$$

Model Name	Priors			Iters.	Backbones		TartanAir (val)		KITTI 2012 (train)		#MACs
	$\Phi_{0,1}$	$D_{0,1}$	BaseNet		Optical Flow	Depth	EPE	lpx	Fl-EPE	Fl-All	
SEA-RAFT (S)				4	SEA-RAFT (S)		1.38	6.24	1.94	6.31	284.7G
SEA-RAFT (S*)				12	SEA-RAFT (S)		1.28	5.83	1.86	6.16	452.9G
SEA-RAFT (M)				4	SEA-RAFT (M)		1.35	6.13	1.91	5.93	486.9G
SEA-RAFT (L)				12	SEA-RAFT (M)		1.30	5.93	1.79	5.99	655.1G
	✓			4			1.30	5.88	1.76	5.69	435.0G
		✓		4			1.15	5.36	1.42	4.67	435.4G
FlowSeek (T)	✓	✓		4			1.11	5.20	1.43	4.70	400.2G
			✓	4	SEA-RAFT (S)	Depth Any. v2 (S)	1.04	4.79	1.31	4.23	659.5G
	✓		✓	4			1.03	4.72	1.30	4.16	694.7G
	✓	✓	✓	4			1.03	4.82	1.29	4.19	694.7G
FlowSeek (S)	✓	✓	✓	12	SEA-RAFT (S)	Depth Any. v2 (S)	0.99	4.50	1.20	3.95	1241.9G
FlowSeek (M)	✓	✓	✓	4	SEA-RAFT (M)	Depth Any. v2 (B)	0.90	4.15	1.35	4.37	1312.2G
FlowSeek (L)	✓	✓	✓	12	SEA-RAFT (M)	Depth Any. v2 (B)	0.85	3.87	1.25	4.12	1859.4G
	✓	✓	✓	4		DPT-Hybrid	1.08	5.10	1.52	5.19	865.6G
FlowSeek (T)	✓	✓	✓	4	SEA-RAFT (S)	Depth Any. v1 (S)	1.04	5.01	1.44	4.82	694.7G
	✓	✓	✓	4		Depth Any. v2 (S)	1.03	4.72	1.30	4.16	694.7G
CRAFT				12	CRAFT		1.77	8.31	2.17	9.03	315.6G
CRAFT (FlowSeek)			✓	12		Depth Any. v2 (S)	1.39	7.00	1.62	7.22	423.6G
FlowFormer				-	FlowFormer		1.63	7.57	2.67	9.13	974.6G
FlowFormer (FlowSeek)			✓	-			1.30	6.06	1.54	6.36	1587.6G

Table 1. **Ablation and Generality Studies.** We ablate different **priors combinations**, **model sizes**, **depth foundation models**, and **optical flow backbones** on TartanAir and KITTI 2012. The impact is measured against baseline SEA-RAFT models, reported at the top. All the models are trained for 100K steps on TartanAir [80], using a single RTX 3090 GPU. (S)* means SEA-RAFT (S) running 12 iterations.

BaseNet inputs	KITTI 2015 (val)							
	TartanAir (val)		KITTI 2012 (train)		static		dynamic	
	EPE	lpx	Fl-EPE	Fl-All	Fl-EPE	Fl-All	Fl-EPE	Fl-All
D_0	1.05	4.79	1.27	4.21	1.21	2.07	2.49	8.61
B_{motion}	1.03	4.72	1.30	4.16	1.06	1.96	2.60	9.07

Table 2. **Ablation Study – different inputs to the BaseNet.** Model: FlowSeek (T). On the right: models fine-tuned on the first 160 images of KITTI 2015 training set, evaluated on the other 40.

Therefore, we define a set of eight bases by knowing D_0 :

$$B_{\text{motion}} = \{\Delta_{T_x}, \Delta_{T_y}, \Delta_{T_z}, \Delta_{R^{1x}}, \Delta_{R^{2x}}, \Delta_{R^{1y}}, \Delta_{R^{2y}}, \Delta_{R^z}\} \quad (12)$$

as a prior for the space of possible flows.

Although this representation holds only for rigid motions, whether from camera movement alone or from a single independently moving object in the scene, we argue it can provide an initial guess to the flow model, which will further refine it through successive iterations. Purposely, we introduce an additional sub-module, a *BasesNet*, responsible for extracting dense features from the set of bases:

$$B, H_B = \text{BasesNet}(B_{\text{motion}}) \quad (13)$$

These features are concatenated with the original context and hidden state features C, H^0 , which are used throughout the iterative flow estimation process described earlier.

3.4. Supervision

Following SEA-RAFT [81], we model the output optical flow as a mixture of two Laplace distributions. Accordingly, each flow update $\Delta_{\mathcal{F}}^i$ is defined as:

$$\Delta_{\mathcal{F}}^i = \alpha^i \cdot \frac{e^{-\frac{|x-\mu^i|}{e^{\beta_1}}}}{2e^{\beta_1}} + (1-\alpha^i) \cdot \frac{e^{-\frac{|x-\mu^i|}{e^{\beta_2}}}}{2e^{\beta_2}} \quad (14)$$

where $\alpha^i, \beta_2^i, \mu^i$ are the six parameters defining the mixture (two for each flow coordinate) predicted by the *FlowHead* at iteration i , and β_1 is fixed to 0. Finally, optical flow predictions \mathcal{F}^j at each iteration j are supervised through log-likelihood minimization [81]:

$$\mathcal{L}_{\mathcal{F}} = \sum_{j=0}^{\text{iters}} \gamma^{N-j} (-\log \mathcal{F}^j) \quad (15)$$

4. Experimental Results

4.1. Implementation Details

FlowSeek is implemented on top of SEA-RAFT [81] code-base. Specifically, the *FeatNet*, *ContextNet*, and *BasesNet* are implemented by a subset of layers of either a ResNet-18 or a ResNet-34 [20]: depending on the choice, the original SEA-RAFT model comes in two variants, respectively *small* (S) or *medium* (M). The number of iterative updates *iters* is usually set to 4, with a third SEA-RAFT variant running 12 iterations on top of the (M) model – namely, SEA-RAFT *large* (L). We select Depth Anything v2 [93] as the depth foundation model for our experiments, either in its *small* (S) or *base* (B) variants. By playing with different backbone sizes and iterations, we implement the *tiny* (T) and *small* (S) variants using ResNet-18, Depth Anything v2 (S) and setting *iters* to 4 or 12 respectively; we also implement *medium* (M) and *large* (L) ones, by select-

Extra Data	Method	Sintel		KITTI 2015	
		Clean↓	Final↓	Fl-EPE↓	Fl-all↓
	PWC-Net [70]	2.55	3.93	10.4	33.7
	RAFT [76]	1.43	2.71	5.04	17.4
	GMA [36]	1.30	2.74	4.69	17.1
	SKFlow [75]	1.22	2.46	4.27	15.5
	FlowFormer [26]	1.01	2.40	4.09†	14.7†
	DIP [98]	1.30	2.82	4.29	13.7
	EMD-L [13]	0.88	2.55	4.12	13.5
	CRAFT [68]	1.27	2.79	4.88	17.5
	RPKNet [55]	1.12	2.45	-	13.0
	GMFlowNet [97]	1.14	2.71	4.24	15.4
	-----	1.27	4.32	4.61	15.8
	SEA-RAFT (S) [81]	1.28*	3.02*	5.10*	16.5*
	-----	1.21	4.04	4.29	14.2
	SEA-RAFT (M) [81]	1.30*	3.09*	5.30*	15.8*
	-----	1.19	4.11	3.62	12.9
	SEA-RAFT (L) [81]	1.21*	3.08*	4.37*	14.3*
	-----	1.12	2.53	3.95	12.7
	FlowSeek (T)	1.04	2.43	3.36	11.5
	FlowSeek (S)	1.15	2.40	4.59	13.7
	FlowSeek (M)	1.07	2.21	3.82	12.5
	FlowSeek (L)	1.07	2.21	3.82	12.5
	-----	-	-	8.70	24.4
Tartan	GMFlow [87]	-	-	8.70	24.4
	-----	1.27	3.74	4.43	15.1
	SEA-RAFT (S) [81]	1.27*	2.89*	4.99	15.7
	-----	1.27	3.85	4.30	14.3
	SEA-RAFT (M) [81]	1.36*	2.91*	5.45*	16.0*
	-----	1.23	3.37	3.73	12.7
Tartan	SEA-RAFT (L) [81]	1.22*	2.73*	4.21*	13.5*
	-----	1.13	2.48	4.06	12.2
	FlowSeek (T)	1.05	2.37	3.32	11.0
	FlowSeek (S)	1.10	2.31	3.99	12.1
	FlowSeek (M)	1.03	2.18	3.31	11.2
	FlowSeek (L)	1.03	2.18	3.31	11.2
AF → AF+T	DDVM [65]	1.48	2.22	3.71	14.07
AF → AF+T+KU+Tartan	DDVM [65]	1.24	2.00	2.19	7.58

Table 3. **Zero-Shot Generalization – Sintel (train) and KITTI 2015 (train).** Methods on top are trained with “C→T” schedule. † denotes tiling at test time. * denotes model trained with one GPU.

ing ResNet-34, Depth Anything v2 (B) and setting `iters` to 4 or 12 respectively.

Datasets and Metrics. Following the existing literature [81], our experiments involve TartanAir [80], FlyingChairs [15], FlyingThings3D [52] and HD1K [40] mainly for training purposes, as well as KITTI 2012 [17], Spring [53] and LayeredFlow [84] mainly for evaluation purposes, with KITTI 2015 [54] and Sintel [8] train sets being involved in both phases. To measure the accuracy of the different flow models, we use standard metrics such as the End-Point-Error (EPE) over any dataset, with additional metrics on TartanAir and Spring (% of pixels with error larger than 1 pixel, *1px*), KITTI datasets (% of pixels with error larger than 3 pixels or relative error higher than 5%, *Fl-All*), and LayeredFlow (% of pixels with error larger than 1, 3 or 5 pixels, respectively *1px*, *3px* and *5px*).

Training Schedule. For ablation studies, we train the original SEA-RAFT and FlowSeek variants on TartanAir [80], excluding the `westerndesert` and `soulcity` sequences, which we reserve for evaluation. Each training run is carried out for 100K steps, using standard hyperparameters from SEA-RAFT codebase [81]. All models are

trained on a **single RTX 3090 GPU**, with batch sizes of 6 and 4 for variants using ResNet-18 and ResNet-34.

For generalization experiments on Sintel and KITTI 2015, FlowSeek follows the multi-stage training schedule outlined in [81], again on a single RTX 3090 GPU. First, the models are trained on TartanAir [80] for 300K steps, maintaining the batch size as mentioned above. Then, they undergo a second stage on FlyingChairs [15] (C), with batch sizes of 8 and 6 for the two ResNet variants, followed by a third stage on FlyingThings3D [52] (T) for 120K steps, with batch 4 and 2. Finally, for experiments on Spring and LayeredFlow, we further fine-tune for 300K steps on a mixture of FlyingThings3D [52], Sintel [8], KITTI [54] and HD1K [40] (TSKH), with batch sizes 4 and 2.

4.2. Ablation Studies and Analysis

We start our study by evaluating the effectiveness of different design choices for implementing FlowSeek. Table 1 collects the outcome of this analysis carried out on TartanAir [80] and KITTI 2012 [17], with the original SEA-RAFT models being reported at the very top as a reference.

Prior Combinations. The first set of experiments aim to assess the impact of the different priors provided by the depth foundation model to FlowSeek (T). Using either features $\Phi_{0,1}$ or sending $D_{0,1}$ to the `ContextNet` alone improves performance over the SEA-RAFT (S) baseline, while combining the two further decreased the error on TartanAir at the expenses of generalization. However, the `BaseNet` alone yields consistently better results, proving to be the core component for optimally exploiting the depth foundation model, although with a noticeable increase in complexity. Finally, combining the `BaseNet` with $\Phi_{0,1}$ yields the absolute best results – this configuration, highlighted in yellow, will be used from now on – while combining the three priors slightly decreases accuracy.

Model Size. By playing with both the ResNet type and the `iters` parameter, we can implement different trade-offs between accuracy and complexity. Switching to ResNet-34 and replacing Depth Anything v2 (S) with the *base* model (B) improves results on TartanAir, with minor drops on KITTI, while running more iterations consistently yields better performance. Notably, every FlowSeek variant outperforms its baseline counterpart reported at the top – our (T) vs (S), our (S) vs (S*), and so on.

Depth Foundation Models. To demonstrate the generality of our design scheme, we train FlowSeek (T) variants using different depth foundation models, including DPT [59] and Depth Anything v1 [92]. We can observe that all variants substantially outperform the SEA-RAFT (S) baseline, with accuracy increasing when switching to newer models such as those in the Depth Anything series [92, 93]. We speculate that future, more accurate foundation models could further enhance FlowSeek performance.

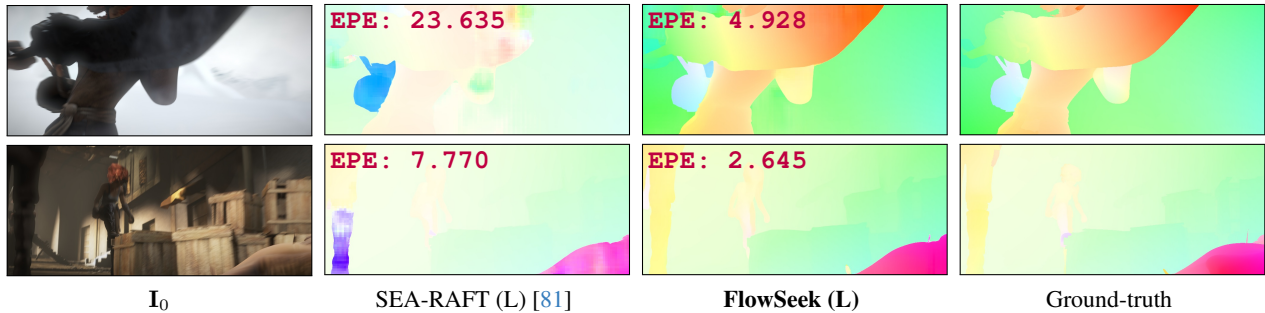


Figure 3. **Qualitative Results on Sintel** [8]. From left to right: first frame, flow by SEA-RAFT (L) and FlowSeek (L), ground-truth flow.

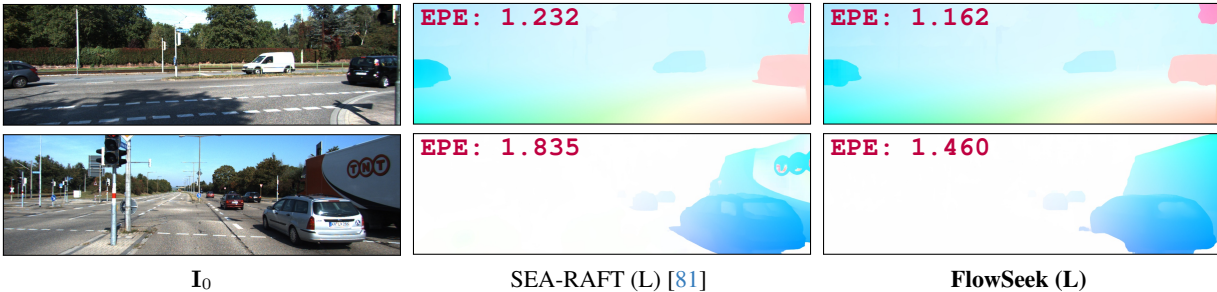


Figure 4. **Qualitative Results on KITTI 2015** [54]. From left to right: first frame, flow by SEA-RAFT (L) and FlowSeek (L).

Optical Flow Backbones. We further prove the generality of our approach by empowering two different flow backbones, respectively CRAFT [68] and FlowFormer [26], with the priors extracted by a BaseNet. This addition largely improves the accuracy of both baseline model.

Input to the BaseNet. Finally, we compare the performance of FlowSeek (T) when replacing the motion bases with the original depth map as input to the BaseNet. Tab. 2 shows how the bases yield improvements on both TartanAir and KITTI 2012. These datasets, however, mostly contain ego-motion induced by the camera. We further fine-tune both models for 10K steps on the first 160 images of the KITTI 2015 training set and evaluate on the remaining 40. We can notice a consistent improvement in static regions, at the price of a drop on moving objects.

4.3. Zero-Shot Generalization

We now assess the ability of FlowSeek to generalize across datasets compared to SEA-RAFT [81] and other methods. From now on, we will highlight the **absolute**, the **second**, and the **third** best methods in each table among those trained following the protocol defined by SEA-RAFT.

Sintel and KITTI 2015. Table 3 collects results achieved on Sintel and KITTI 2015 training sets, following the evaluation protocol established by SEA-RAFT [81]. Models reported at the top were trained on “C+T” only, while at the bottom we show the results achieved when pre-training on TartanAir is performed. For SEA-RAFT, we report both the results by the authors, as well as those reproduced by retraining on a single GPU to highlight the impact that the hardware budget has on final accuracy – dramati-

cally dropping on KITTI, while improving on Sintel *Final*.

Among the former category, FlowSeek (T) already outperforms most existing methods, including the original SEA-RAFT (S) against which it marks a consistent margin on both datasets despite the disparity in hardware budget. FlowSeek (S) further improves over (T), while FlowSeek (M) and (L) outperform all competitors on Sintel *Final*, though losing some accuracy on KITTI as previously observed in our ablation studies. We attribute this to the higher complexity of (M) and (L) variants, whose training is likely constrained on a single GPU without strong pre-training, making FlowSeek (S) as the best solution under this setting.

When pre-training on TartanAir, we observe moderate improvements in the SEA-RAFT models, with much larger gains achieved by FlowSeek variants. In particular, FlowSeek (L) achieves the overall best results across Sintel *Final* and KITTI 2015, despite the moderate hardware budget used for training – resulting in a batch size $8\times$ smaller than its SEA-RAFT (L) counterpart. As a reference, we also report the results achieved by DDVM [65]: when only AutoFlow (AF) [72] and T are used (a setting similar to ours), FlowSeek outperforms it, whereas DDVM achieves superior results when also using Kubric (KU) and TartanAir.

Figures 3 and 4 showcase qualitative results by SEA-RAFT (L) and FlowSeek (L) on Sintel and KITTI.

Spring. We continue our investigation on the Spring dataset [53], evaluating SEA-RAFT and FlowSeek variants after further training on “TSKH”, by downsampling input images by a factor $2\times$ as in [81]. Table 4 collects the outcome of this evaluation. Generally, the domain gap from “C→T→TSKH” to Spring is much lower compared to what

Extra Data	Method	Spring (train)	
		1px↓	EPE↓
Tartan	RAFT [76]	4.788	0.448
	GMA [36]	4.763	0.443
	RPKNet [55]	4.472	0.416
	DIP [98]	4.273	0.463
	SKFlow [75]	4.521	0.408
	GMFlow [87]	29.49	0.930
	GMFlow+ [88]	4.292	0.433
	Flowformer [26]	4.508	0.470
	CRAFT [68]	4.803	0.448
	SEA-RAFT (S)	4.161	0.410
	SEA-RAFT (M)	3.888	0.406
	SEA-RAFT (L)	3.842	0.426
	FlowSeek (T)	4.111	0.410
	FlowSeek (S)	4.058	0.406
FlowSeek (M)	3.941	0.419	
FlowSeek (L)	3.838	0.402	
MegaDepth [44]	MatchFlow(G) [14]	4.504	0.407
YT-VOS [89]	Flowformer++[67]	4.482	0.447
VIPER [62]	MS-RAFT+ [33]	3.577	0.397

Table 4. **Zero-Shot Generalization – Spring**. Methods on top are trained with “C → T → TSKH” schedule.

occurs from “C→T” to Sintel and KITTI, with EPE values falling below 0.5. Nevertheless, we observe that FlowSeek (L) beats any SEA-RAFT model, despite being trained on a single GPU compared to the multiple GPUs used by its counterparts. Only MS-RAFT+ performs slightly better, though it uses $2\times$ A100 GPUs and VIPER [62] data.

LayeredFlow. We conclude our evaluation by conducting a further zero-shot evaluation experiment on the recent LayeredFlow dataset [84], which features challenging transparent and reflective surfaces. Following [84], we evaluate both SEA-RAFT and FlowSeek “C→T→TSKH” models on the validation set by down-sampling images to $\frac{1}{8}$ of their original resolution and evaluating the predictions on the *first Layer* – i.e., the surfaces closest to the camera. Table 5 presents the results achieved by representative optical flow architectures, as well as those involved in our experiments at the bottom. First and foremost, we note that every FlowSeek variant outperforms its SEA-RAFT counterpart, often by substantial margins – e.g., FlowSeek (L) improves EPE on *All* category by more than 2 pixels. Moreover, when compared to other existing methods, FlowSeek (M) and (L) achieve consistently lower errors on *All* pixels. In the interest of space, we refer the reader to the supplementary material for detailed results on single categories – classifying image regions into *Transparent*, *Reflective* and *Diffuse*.

Figure 5 shows qualitative comparisons between the predictions by SEA-RAFT (L) and FlowSeek (L), with the latter recovering finer details and exhibiting fewer artifacts.

Qualitative Results. We refer the reader to the supplementary material for additional qualitative samples concerning the experiments discussed so far.

Method	All			
	EPE↓	1px↓	3px↓	5px↓
FlowNet-C [15]	9.71	89.07	61.51	43.93
FlowNet2 [30]	10.07	77.56	54.22	42.13
PWC-Net [70]	9.49	74.93	50.47	39.05
GMA [36]	9.77	72.46	46.93	36.97
SKFlow [75]	9.86	72.02	47.44	36.88
CRAFT [68]	10.36	72.34	47.54	37.00
GMFlow [87]	9.09	81.99	51.79	37.75
GMFlow+ [88]	9.46	82.71	53.14	39.70
FlowFormer [26]	10.20	73.59	48.97	38.56
RAFT [76]	9.38	71.98	46.46	36.15
SEA-RAFT (S)	10.05	71.48	46.90	36.32
SEA-RAFT (M)	10.17	69.73	45.94	34.78
SEA-RAFT (L)	10.99	69.46	45.59	34.78
FlowSeek (T)	9.09	70.82	43.74	32.36
FlowSeek (S)	9.16	69.99	43.67	31.90
FlowSeek (M)	8.30	68.85	41.81	32.09
FlowSeek (L)	8.30	68.98	41.49	31.64

Table 5. **Zero-shot Generalization – LayeredFlow (train) first layer evaluation**. Images are down-sampled by a factor 8.

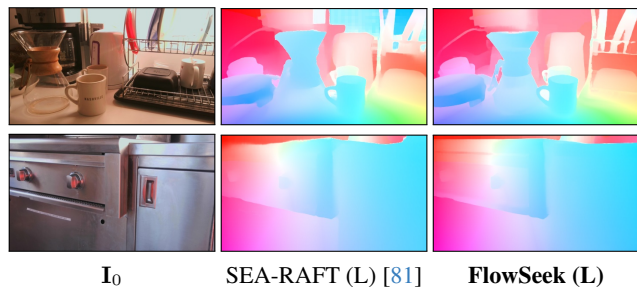


Figure 5. **Qualitative Results on LayeredFlow [84]**. From left to right: first frame, flow by SEA-RAFT (L) and FlowSeek (L).

5. Conclusion

We presented FlowSeek, a novel architecture for optical flow that combines the latest architectural developments in the field with cutting-edge depth foundation models and low-dimensional flow parametrization from classical computer vision. The synergy of the three makes FlowSeek a practical model capable of achieving state-of-the-art zero-shot generalization, even when trained with as few as a single consumer-grade GPU. We believe FlowSeek can inspire further attempts to design new models trainable with minimal hardware budget in adjacent fields of computer vision.

Limitations. The possibility of training FlowSeek comes from the availability of large, pre-trained foundation models which were likely trained with much higher hardware budget on web-scale data. However, we hope our work encourages the community to avoid training new architectures from scratch at prohibitive costs and instead reuse existing models when possible.

Future Work. Training data remains another significant bottleneck for flow models. Future research will focus on this aspect, attempting to emulate the successful strategies in the depth estimation literature [92, 93].

Acknowledgment. We thank Sadra Safadoust and Fatma Güney for the insightful discussion about motion bases.

References

- [1] Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning optical flow from still images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15201–15211, 2021. 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 (CVPR)*, pages 1013–1027, 2025. 2, 3
- [3] Michael J Black and Padmanabhan Anandan. A framework for the robust estimation of optical flow. In *1993 (4th) International Conference on Computer Vision*, pages 231–236. IEEE, 1993. 2
- [4] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 3
- [5] Richard Strong Bowen, Richard Tucker, Ramin Zabih, and Noah Snavely. Dimensions of motion: Monocular prediction through flow subspaces. In *2022 International Conference on 3D Vision (3DV)*, pages 454–464, 2022. 2, 4
- [6] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004. 1
- [7] Thomas Brox, Christoph Bregler, and Jitendra Malik. Large displacement optical flow. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48. IEEE, 2009. 1
- [8] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 6, 7
- [9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3
- [10] Qifeng Chen and Vladlen Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4706–4714, 2016. 1
- [11] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2, 3
- [12] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Flowtrack: Revisiting optical flow for long-range dense tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19268–19277, 2024. 2
- [13] Changxing Deng, Ao Luo, Haibin Huang, Shaodan Ma, Jiangyu Liu, and Shuaicheng Liu. Explicit motion disentangling for efficient optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9521–9530, 2023. 2, 6
- [14] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Rethinking optical flow from geometric matching consistent perspective. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 1337–1347, 2023. 2, 8
- [15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1, 2, 6, 8
- [16] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 1
- [17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6
- [18] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2
- [19] Yunhui Han, Kunming Luo, Ao Luo, Jiangyu Liu, Haoqiang Fan, Guiming Luo, and Shuaicheng Liu. RealFlow: Embased realistic optical flow dataset generation from videos. In *European conference on computer vision*, pages 288–305. Springer, 2022. 2
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 5
- [21] David J Heeger and Allan D Jepson. Subspace methods for recovering rigid motion i: Algorithm and implementation. *International Journal of Computer Vision*, 7:95–117, 1992. 2, 4
- [22] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 1, 2
- [23] Yinlin Hu, Rui Song, and Yunsong Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5704–5712, 2016. 1
- [24] Yinlin Hu, Yunsong Li, and Rui Song. Robust interpolation of correspondences for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1
- [25] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Rife: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2020. 1

- [26] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European conference on computer vision*, pages 668–685. Springer, 2022. 1, 2, 6, 7, 8
- [27] Tak-Wai Hui and Chen Change Loy. LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1
- [28] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8981–8989, 2018. 2
- [29] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow cnn - revisiting data fidelity and regularization. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2
- [30] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 8
- [31] Sergio Izquierdo, Mohamed Sayed, Michael Firman, Guillermo Garcia-Hernando, Daniyar Turmukhambetov, Javier Civera, Oisín Mac Aodha, Gabriel J. Brostow, and Jamie Watson. MVSAnywhere: Zero shot multi-view stereo. In *CVPR*, 2025. 2
- [32] Azin Jahedi, Maximilian Luz, Marc Rivinius, and Andrés Bruhn. Ccmr: High resolution optical flow estimation via coarse-to-fine context-guided motion reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6899–6908, 2024. 2
- [33] Azin Jahedi, Maximilian Luz, Marc Rivinius, Lukas Mehl, and Andrés Bruhn. Ms-raft+: high resolution multi-scale raft. *International Journal of Computer Vision*, 132(5): 1835–1856, 2024. 2, 8
- [34] Jisoo Jeong, Hong Cai, Rishiek Garrepalli, and Fatih Porikli. Distractflow: Improving optical flow estimation via realistic distractions and pseudo-labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13691–13700, 2023. 2
- [35] Hualie Jiang, Zhiqiang Lou, Laiyan Ding, Rui Xu, Minglang Tan, Wenjie Jiang, and Rui Huang. Defom-stereo: Depth foundation model based stereo matching. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 4
- [36] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9772–9781, 2021. 2, 6, 8
- [37] Yang Jiao, Trac D Tran, and Guangming Shi. Effiscene: Efficient per-pixel rigidity inference for unsupervised joint learning of optical flow, depth, camera pose and motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5538–5547, 2021. 2
- [38] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 557–572. Springer, 2020. 2
- [39] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 2, 3
- [40] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–28, 2016. 6
- [41] Marius Leordeanu, Andrei Zanfir, and Cristian Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1721–1728, 2013. 1
- [42] Vincent Leroy, Jerome Revaud, Thomas Lucas, and Philippe Weinzaepfel. Win-win: Training high-resolution vision transformers from two windows. *arXiv preprint arXiv:2310.00632*, 2023. 2
- [43] Yu Li, Dongbo Min, Minh N Do, and Jiangbo Lu. Fast guided global interpolation for depth and motion. In *European Conference on Computer Vision*, pages 717–733. Springer, 2016. 1
- [44] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 8
- [45] Chuang-Wei Liu, Qijun Chen, and Rui Fan. Playing to vision foundation model’s strengths in stereo matching. *IEEE Transactions on Intelligent Vehicles*, 2024. DOI:10.1109/TIV.2024.3467287. 3
- [46] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6489–6498, 2020. 2
- [47] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Flow2stereo: Effective self-supervised learning of optical flow and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6648–6657, 2020. 2
- [48] Shuaicheng Liu, Yuhang Lu, Hai Jiang, Nianjin Ye, Chuan Wang, and Bing Zeng. Unsupervised global and local homography estimation with motion basis learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7885–7899, 2022. 2

- [49] Xiaozhang Liu, Hui Liu, and Yuxiu Lin. Video frame interpolation via optical flow estimation with image inpainting. *International Journal of Intelligent Systems*, 35(12):2087–2102, 2020. 1
- [50] Yawen Lu, Qifan Wang, Siqi Ma, Tong Geng, Yingjie Victor Chen, Huaijin Chen, and Dongfang Liu. Transflow: Transformer as flow learner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18063–18073, 2023. 1
- [51] Ao Luo, Fan Yang, Xin Li, and Shuaicheng Liu. Learning optical flow with kernel patch attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8906–8915, 2022. 2
- [52] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 2, 6
- [53] Lukas Mehl, Jenny Schmalfluss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4991, 2023. 2, 6, 7
- [54] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 6, 7
- [55] Henrique Morimitsu, Xiaobin Zhu, Xiangyang Ji, and Xu-Cheng Yin. Recurrent partial kernel network for efficient optical flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4278–4286, 2024. 2, 6, 8
- [56] OpenAI. Hello GPT-4o, 2024. 2
- [57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2
- [58] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 3
- [59] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 3, 4, 6
- [60] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017. 1, 2
- [61] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1164–1172, 2015. 1
- [62] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222, 2017. 8
- [63] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2
- [64] Sadra Safadoust and Fatma Güney. Multi-object discovery by low-dimensional object motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 4
- [65] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *Advances in Neural Information Processing Systems*, 36, 2023. 2, 6, 7
- [66] Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12469–12480, 2023. 2
- [67] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1599–1610, 2023. 1, 2, 8
- [68] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformer for robust optical flow. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 17602–17611, 2022. 1, 2, 6, 7, 8
- [69] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2432–2439. IEEE, 2010. 1
- [70] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 1, 2, 6, 8
- [71] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1408–1423, 2019. 1, 2
- [72] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10093–10102, 2021. 2, 7
- [73] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J Fleet, and William T Freeman. Disentan-

- gling architecture and training for optical flow. In *European Conference on Computer Vision*, pages 165–182. Springer, 2022. 2
- [74] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1390–1399, 2018. 1
- [75] Shangkun Sun, Yuanqi Chen, Yu Zhu, Guodong Guo, and Ge Li. Skflow: Learning optical flow with super kernels. *Advances in Neural Information Processing Systems*, 35: 11313–11326, 2022. 2, 6, 8
- [76] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 1, 2, 3, 4, 6, 8
- [77] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. *Advances in Neural Information Processing Systems*, 33, 2020. 1
- [78] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19795–19806, 2023. 2
- [79] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3
- [80] Wenshan Wang, DeLong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 2, 5, 6
- [81] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pages 36–54. Springer, 2024. 1, 2, 3, 5, 6, 7, 8
- [82] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024. 3
- [83] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. *arXiv*, 2025. 2, 3, 4
- [84] Hongyu Wen, Erich Liang, and Jia Deng. Layeredflow: A real-world benchmark for non-lambertian multi-layer optical flow. In *European Conference on Computer Vision*. Springer, 2024. 6, 8
- [85] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 1
- [86] Jonas Wulff and Michael J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*, 2015. 2
- [87] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezaatofghi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022. 1, 2, 6, 8
- [88] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezaatofghi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023. 2, 8
- [89] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 8
- [90] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances in Neural Information Processing Systems*, 32, 2019. 1
- [91] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. *Advances in neural information processing systems*, 32, 2019. 1, 2
- [92] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 2, 3, 4, 6, 8
- [93] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 2, 3, 4, 5, 6, 8
- [94] Nianjin Ye, Chuan Wang, Haoqiang Fan, and Shuaicheng Liu. Motion basis learning for unsupervised deep homography estimation with subspace projection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13117–13125, 2021. 2
- [95] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007. Proceedings 29*, pages 214–223. Springer, 2007. 2
- [96] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 3
- [97] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17592–17601, 2022. 1, 2, 6
- [98] Zihua Zheng, Ni Nie, Zhi Ling, Pengfei Xiong, Jiangyu Liu, Hao Wang, and Jiankun Li. Dip: Deep inverse patch-match for high-resolution optical flow. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8925–8934, 2022. [2](#), [6](#), [8](#)

- [99] 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 39th Annual AAAI Conference on Artificial Intelligence (AAAI 2025)*, Philadelphia, Pennsylvania, USA, 2025. [3](#)