

Estimating 2D Camera Motion with Hybrid Motion Basis

Haipeng Li^{1*} Tianhao Zhou^{1*} Zhanglei Yang¹ Yi Wu² Yan Chen²
 Zijing Mao² Shen Cheng³ Bing Zeng¹ Shuaicheng Liu^{1†}
¹University of Electronic Science and Technology of China
²Xiaomi Corporation ³Dexmal

Abstract

*Estimating 2D camera motion is a fundamental computer vision task that models the projection of 3D camera movements onto the 2D image plane. Current methods rely on either homography-based approaches, limited to planar scenes, or meshflow techniques that use grid-based local homographies but struggle with complex non-linear transformations. We introduce **CamFlow**, a novel framework that represents camera motion using hybrid motion bases: physical bases derived from camera geometry and stochastic bases for complex scenarios. Our approach includes a hybrid probabilistic loss function based on the Laplace distribution that enhances training robustness. For evaluation, we create a new benchmark by masking dynamic objects in existing optical flow datasets to isolate pure camera motion. Experiments show CamFlow outperforms state-of-the-art methods across diverse scenarios, demonstrating superior robustness and generalization in zero-shot settings. Code and datasets are available at our project page: <https://lhaipp.github.io/CamFlow/>.*

1. Introduction

Estimating 2D camera motion, which involves recovering the projection of 3D rotation and translation onto 2D planes [14], is a cornerstone of computer vision. Given a 3D rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, the camera motion $\mathbf{M}$ can be expressed as:

$$\mathbf{M} = \mathbf{K} \left(\mathbf{R} + \mathbf{t} \frac{\mathbf{n}^T}{d} \right) \mathbf{K}^{-1}, \quad (1)$$

where $\mathbf{n}^T$ is the transpose of the normal vector to planes in the scene, d denotes the distance from the camera center to each plane, and $\mathbf{K}$ represents the camera intrinsic matrix. The resulting 2D camera motion is inherently non-linear due to its dependence on scene depth and plane geometry. In real-world images, scenes typically consist of

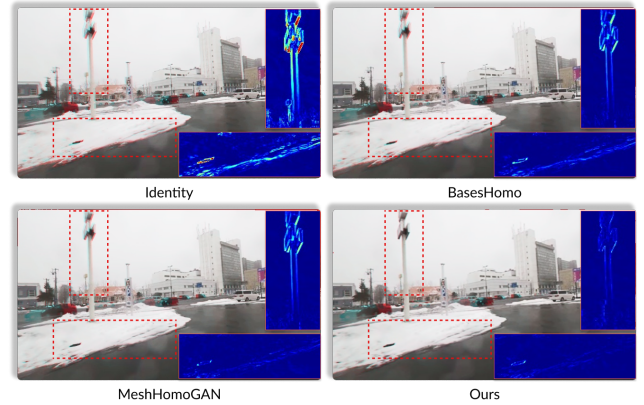


Figure 1. Comparison of camera motion estimation approaches in multi-plane scenes. Visualizations show warped source images overlaid on targets, with brighter areas in heatmaps indicating higher error. BasesHomo [44] captures only background motion, while MeshHomoGAN [31] improves accuracy through local grid-based homographies. Our CamFlow, with hybrid motion basis, achieves superior representation of complex camera motion.

multiple depths and planes. As a result, different regions of an image undergo distinct transformations, leading to complex, non-linear motion patterns. This task is essential for various computational imaging applications, such as digital video stabilization [26], where accurate representation of camera motion directly enhances performance. Existing methods for modeling camera motion primarily fall into two categories: homography and meshflow-based approaches.

Homography, a perspective transformation, aligns two views of a planar or nearly-planar 3D scene [14]. Traditional methods typically rely on artificially extracted and matched keypoints [33], excelling under standard conditions but often struggling in adverse scenarios (e.g., rain, snow, and low light) or when confronted with non-planar motion caused by depth parallax or dynamic objects. Recent deep learning approaches mitigate the reliance on keypoints, enabling direct estimation of homography from image pairs through robust, data-driven networks. A notable benchmark work, BasesHomo [44], reformulates the task

*Equal contribution.

†Corresponding author (liushuaicheng@uestc.edu.cn).

as learning a linear combination of an 8-dimensional motion basis, pioneering a new direction in motion estimation. Nevertheless, they are limited because: *a homography can only align a single plane*.

To address this limitation, MeshFlow [27] partitions the image into $N \times N$ grids, estimating a local homography for each cell and smoothing them to model non-linear camera motion. This approach performs well in scenes with multiple small-baseline depth variations and has become a popular choice for digital video stabilization [26, 27]. Furthermore, deep meshflow variants [28] demonstrate enhanced robustness under challenging conditions. However, a key limitation persists in current camera motion representation: increasing the number of grids enhances the ability to model non-linearity but raises optimization challenges [43].

In this work, we introduce **CamFlow**, a novel representation that models complex camera motion through hybrid motion bases. Our key insight is that the flow field resulting from the superposition of multiple homographies is inherently non-linear (as visualized in Fig. 3), enabling more sophisticated motion modeling beyond single-plane limitations. Building on this observation, we establish a comprehensive hybrid basis subspace comprising:

- **Physical Motion Bases:** Derived from Taylor expansion of homographic transformations up to second-order terms, our 12 physical bases model fundamental geometric transformations (rotation, translation, scaling, and perspective) that capture essential camera motion patterns;
 - **Noisy Motion Bases:** To model complex residual motion, we construct K orthogonal components through SVD decomposition of randomly sampled homographies from a Gaussian distribution, effectively complementing the physical bases by capturing higher-order motion patterns.
- To stabilize the training procedure and simplify complex loss designs, we propose a hybrid probabilistic loss function that assumes motion models follow a Laplace distribution [42], facilitating robust and efficient optimization.

In summary, CamFlow effectively represents complex 2D camera motion as illustrated in Fig. 1, capturing both background and foreground elements while accurately modeling depth-varying motion patterns that conventional approaches struggle to represent. To rigorously evaluate its performance, we introduce **GHOF-Cam**, a novel benchmark specifically designed for camera motion estimation by systematically masking dynamic objects and ill-posed occlusion regions in established optical flow datasets [24], thereby isolating pure camera-induced motion. Through comprehensive experiments across diverse datasets under both standard and challenging conditions, we demonstrate that CamFlow consistently outperforms state-of-the-art single-plane homography and multi-planar methods in both sparse and dense camera motion estimation tasks, exhibiting superior robustness and generalization capability in

real-world scenarios. Our main contributions are:

- A new hybrid motion representation that learns to model complex non-linear 2D camera motion through physically interpretable and stochastic motion bases.
- A novel probabilistic loss formulation based on Laplace distribution that simplifies training and stabilizes optimization without complex loss designs.
- A comprehensive benchmark for evaluating camera motion learning across diverse conditions. Experimental results confirm our approach’s effectiveness, robustness, and generalization ability across real-world scenarios.

2. Related works

2.1. Homography Methods

The traditional homography estimation follows three stages: feature detection (e.g., SIFT [33], ORB [35]), correspondence matching [5], and outlier rejection (e.g., RANSAC [9], MAGSAC++ [2]). Learning-based methods like LIFT [45], SuperPoint [7], and SOSNet [40] improve robustness. Optimization-based approaches [4, 8] iteratively refine homography estimates, while deep learning methods span supervised [3, 6, 17, 21, 23, 37] and unsupervised [15, 19, 34, 44, 47] frameworks. Unsupervised models have proven effective for real-world scenarios. Notable examples include CAHomo [47] and BaseSHomo [44], which enhance feature extraction and motion constraints. HomoGAN [15] integrates GAN loss [12] and Transformers [32] for coarse-to-fine refinement. Despite this progress, homography remains a single-plane model, limiting its effectiveness in complex motion. In this paper, we introduce a hybrid motion-basis representation to model multi-plane, non-linear motion more effectively.

2.2. Mesh-based Methods

Mesh-based warping estimates local homographies per mesh cell, including dual-plane approaches [10], patch-wise mixtures of homography [13], flexible warping techniques like APAP [46], Bundled Paths [26], grid-based methods [27], and cascade residual homography [38]. Deep learning variants include deepMeshFlow [43], MeshCA-Homo [29], which merges multi-resolution meshes, and BasesMesh [30], which applies motion bases per grid. MeshHomoGAN [31] incorporates a planarity-aware mechanism for local homography estimation. However, these methods still assume that local regions fit the homography relationship, limiting their ability to represent complex motion. To address this limitation, we propose a novel motion representation that combines multiple non-linear motion bases, enhancing performance.

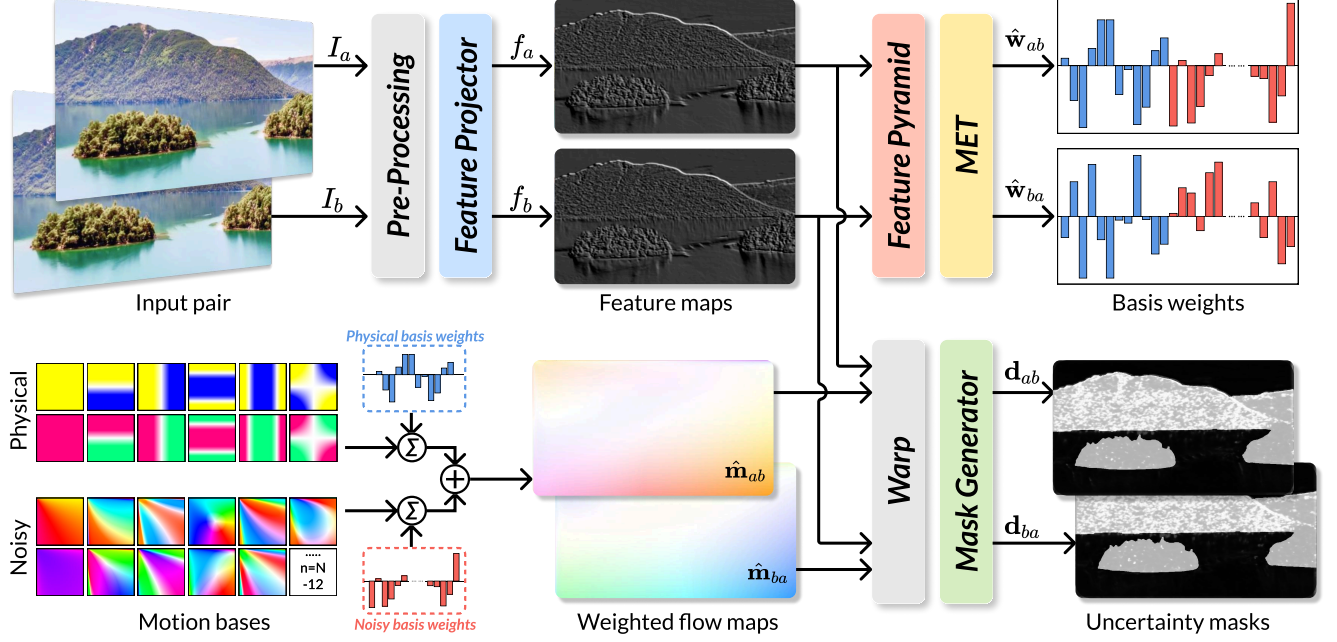


Figure 2. Our proposed motion estimation framework. Given image pair (I_a, I_b) , features are extracted through a multi-scale pyramid and processed by the motion estimation transformer (MET) to compute weights for physical (blue) and noisy (red) motion bases. These weights linearly combine predefined motion bases to generate flow maps for warping. A mask generator predicts uncertainty masks d_{ab} and d_{ba} to reject unreliable regions, enhancing estimation robustness.

3. Method

3.1. Motion Basis

In this work, we propose a novel motion representation through motion basis learning with deep networks to better represent the camera motion via Physical and Noisy bases.

Physical Basis. Consider a pixel with homogeneous coordinates $\mathbf{P}(x, y) = [x, y, 1]^T \in \mathbb{R}^3$. The physical motion $m = [\Delta x, \Delta y, 1]^T$ induced by a homography $\mathbf{H}$ is defined:

$$m = \mathbf{H} \cdot \mathbf{P}(x, y) - \mathbf{P}(x, y), \quad (2)$$

After normalization, the two-dimensional coordinates (x-axis and y-axis) of the motion become:

$$\begin{aligned} \Delta x &= \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + 1} - x, \\ \Delta y &= \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + 1} - y, \end{aligned} \quad (3)$$

where $h_1, \dots, h_8$ denote the elements of matrix $\mathbf{H}$, and h_9 is constrained to 1. By applying a Taylor expansion, this motion can be mapped to another subspace. For instance,

expanding Δx around the point $(x, y) = (0, 0)$ gives:

$$\Delta x = \frac{(h_1 - 1)x + h_2 y + h_3 - h_7 x^2 - h_8 xy}{h_7 x + h_8 y + 1} \quad (4)$$

$$\approx w_1 \cdot 1 + w_2 \cdot x + w_3 \cdot y + w_4 \cdot xy \quad (5)$$

$$+ w_5 \cdot x^2 + w_6 \cdot y^2 + \Delta, \quad (6)$$

where $w_i, i \in [1, 6]$ are coefficients, the basis functions are $b = [1, x, y, xy, x^2, y^2]$ and Δ denotes the higher-order infinitesimal. Similarly, Δy can be decomposed into this subspace. By combining the decompositions of both $\Delta x, \Delta y$, the motion space can be represented using the 12 bases:

$$\mathbf{F} = \{(b_i, 0) \mid b_i \in b\} \cup \{(0, b_i) \mid b_i \in b\}, \quad (7)$$

where b_i denotes the i -th element in b . Each basis can be transformed into an optical flow according to the image coordinate, as shown in Fig. 2 (Physical motion bases).

Non-Linearity of Adding Bases. A key finding is that combining flow fields derived from different homographies produces a non-linear motion field that cannot be represented by any single homography. This challenges the linear basis assumption in prior work [44]. As shown in Fig. 3, when two homography-derived flows (flow1 and flow2) are added together, the resulting flow (flow3) differs from the flow derived from multiplying the original homography matrices (homo3). Additionally, attempting to solve for a homography from points sampled on flow3 yields inconsistent

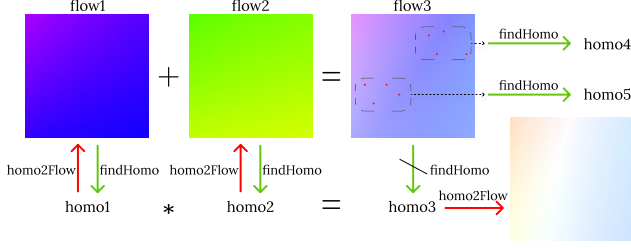


Figure 3. Non-linearity of flow addition. Two homography matrices generate flow1 and flow2. Adding these flows (flow3) differs from the flow derived by multiplying the original homography matrices (homo3). When sampling points from flow3 to solve for a homography, we get inconsistent solutions, proving that combined flow fields cannot be represented by a single homography.

solutions, confirming that the combined flow cannot be represented by any single homography. This observation motivates our approach, which leverages more bases to model complex camera motion.

Stochastic Basis. While the physical bases effectively capture fundamental motion patterns, the complete space of camera motions is infinitely dimensional when considering higher-order Taylor expansions. Exhaustively modeling all possible bases becomes computationally intractable. To address this challenge, we leverage the expressive power of random sampling to complement our physical bases. Specifically, we generate K random 3×3 matrices $\{\mathbf{H}^{(k)}\}_{k=1}^K \subseteq \mathbb{R}^{3 \times 3}$, where each matrix is formulated as:

$$\mathbf{H} = \{h_i\}_{i=1}^9, \text{ where } h_i = \begin{cases} \epsilon_i \sim \mathcal{N}(0, 1), & 1 \leq i \leq 8, \\ 1, & i = 9. \end{cases} \quad (8)$$

Following BasesHomo [44], we convert random matrices into flows and apply singular value decomposition (SVD) to extract principal components. This process yields $N - 12$ stochastic bases that capture diverse motion patterns beyond physical bases, as illustrated in Fig. 2 (Noisy motion bases). These stochastic bases, combined with the 12 physical basis vectors described earlier, form a comprehensive set of N motion bases that enhances our ability to model complex non-linear camera motions.

3.2. Network Structure

The network is illustrated in Fig. 2. Following practices from previous work [15, 47], we process video frames by: (1) random cropping to 320×576 patches, (2) converting to grayscale, (3) projecting into a shallow feature space to handle luminance variations, and (4) generating a 3-layer feature pyramid for multi-scale processing. Then we propose a Motion Estimation Transformer (MET). The MET employs a specialized architecture to separately predict weights for both physical bases (12 weights) and stochastic bases ($N - 12$ weights), yielding bidirectional weights $\hat{\mathbf{w}}_{ab}$ and $\hat{\mathbf{w}}_{ba}$. The final camera motion is computed through a

linear combination of the predicted weights and their corresponding motion bases, producing bidirectional dense motion fields $\hat{\mathbf{m}}_{ab}$ and $\hat{\mathbf{m}}_{ba}$.

It is noteworthy that the confidence masks $\mathbf{d}_{ab}$ and $\mathbf{d}_{ba}$ are also crucial in predicting camera motion, as they effectively filter out dynamic objects. First, we apply the predicted $\hat{\mathbf{m}}_{ab}$ and $\hat{\mathbf{m}}_{ba}$ to f_a and f_b , respectively, obtaining warped feature maps f'_a and f'_b . The mask network $\mathcal{M}$ [22] inputs the concatenated features to generate weighted maps that highlight well-aligned regions:

$$\mathbf{d}_{ab} = \mathcal{M}([f_a, f'_b]), \mathbf{d}_{ba} = \mathcal{M}([f_b, f'_a]). \quad (9)$$

3.3. Loss Function

Probabilistic Motion Modeling. Camera motion estimation faces a fundamental challenge: distinguishing between camera-induced and object motion in the scene. We therefore formulate our approach as a probabilistic model that explicitly accounts for uncertainty in motion estimation.

Building on previous findings that 2D motion follows a Laplace distribution [11, 16, 42], we model the conditional probability of the true camera motion given our prediction:

$$p(\mathbf{m}_{ab} | \hat{\mathbf{m}}_{ab}; \mathbf{d}_{ab}), \quad (10)$$

where $\hat{\mathbf{m}}_{ab}$ is our predicted motion field and $\mathbf{d}_{ab}$ represents our confidence in each pixel's motion estimate. Higher confidence values indicate pixels likely following camera motion, while lower values suggest pixels belonging to independently moving objects.

Laplace Distribution Model. We model the probability density using two conditionally independent Laplace distributions for the horizontal and vertical components:

$$\mathcal{L}(\mathbf{m}_{ab} | \hat{\mathbf{m}}_{ab}; \mathbf{d}_{ab}) = \prod \left(\frac{1}{\sqrt{2}\sigma^2} e^{-\sqrt{\frac{2}{\sigma^2}}|u-\mu_u|} \cdot \frac{1}{\sqrt{2}\sigma^2} e^{-\sqrt{\frac{2}{\sigma^2}}|v-\mu_v|} \right), \quad (11)$$

where (u, v) are the ground-truth motion components, (μ_u, μ_v) are the predicted motion components, and σ^2 is derived from our confidence mask $\mathbf{d}_{ab}$. This formulation allows our model to express both its prediction and its uncertainty about that prediction.

Hybrid Loss Strategy. Another challenge in camera motion estimation is the scarcity of ground-truth labels. To overcome this, we introduce a hybrid loss strategy with two components: 1) Motion supervision loss (ℓ_{NLL_m}): we generate pseudo-labels using existing methods and apply negative log-likelihood (NLL) loss in both forward and backward directions:

$$\ell_{NLL_m} = -\log p(\mathbf{m}_{ab} | \hat{\mathbf{m}}_{ab}; \mathbf{d}_{ab}) - \log p(\mathbf{m}_{ba} | \hat{\mathbf{m}}_{ba}; \mathbf{d}_{ba}). \quad (12)$$

2) Photometric loss (ℓ_{NLL_p}): We apply the same probabilistic framework to enforce consistency between warped features. Given image features f_a and f_b , we use our predicted



Figure 4. Using the GHOF benchmark, we leverage SAM to generate semantic maps and manually identify dynamic objects such as cars and people, producing corresponding masks. These masks are then dilated to encompass occlusion regions, reducing ill-posed artifacts. Finally, the masks are applied to both the images and optical flow, isolating camera-induced motion and constructing a camera-motion-only dataset.

motion to produce warped features f'_a and f'_b :

$$\ell_{NLL_p} = -\log p(f_a | f'_b; \mathbf{d}_{ab}) - \log p(f_b | f'_a; \mathbf{d}_{ba}). \quad (13)$$

Adaptive loss balancing: to ensure stable training despite the different scales of our loss components, we dynamically balance them using:

$$\ell_{overall} = \ell_{NLL_p} + \mathbf{w} \times \frac{|\ell_{NLL_p}|}{|\ell_{NLL_m}|} \cdot \ell_{NLL_m}, \quad (14)$$

where $\mathbf{w}$ is a predefined weight.

4. Experiments

4.1. Dataset

We evaluate our method on: CAHomo [47] and GHOF [24]. We train on CAHomo (460K training pairs, 4.2K test pairs across regular, low-texture, low-light, and foreground scenes) with additional generated samples [23], and conduct zero-shot testing on GHOF (256 test pairs in Regular, Foggy, Low-light, Rainy, and Snowy conditions).

GHOF-Cam Benchmark. To isolate camera motion from dynamic scene elements, we propose a camera-motion-specific benchmark derived from GHOF. We employ the Segment Anything Model (SAM) [20] to generate semantic maps, from which we identify dynamic objects (e.g., cars, people) and create corresponding masks. These masks are dilated to encompass occlusion regions, mitigating ill-posed artifacts at object boundaries. For edge occlusions not detected by semantic segmentation, we mainly utilize the ground-truth homography to identify black edge regions as additional masks. The combined masks are then applied to both input images and ground-truth optical flow, resulting in a benchmark that exclusively captures camera-induced motion, as illustrated in Fig. 4.

4.2. Comparison with Existing Methods

We assess CamFlow against a comprehensive set of methods across three primary categories. The first category encompasses feature-based methods: SIFT [33], ORB [35], SuperPoint[7] with SuperGlue (SPSG) [36], and LoFTR

	Methods	AVG	RE	LT	LL	SF	LF
1)	$\mathcal{I}_{3 \times 3}$	6.70	7.75	7.65	7.21	7.53	3.39
2)	SIFT[33] + RANSAC[9]	1.41	0.30	1.34	4.03	0.81	0.57
3)	SIFT[33] + MAGSAC[1]	1.34	0.31	1.72	3.39	0.80	0.47
4)	ORB[35] + RANSAC[9]	1.48	0.85	2.59	1.67	1.10	1.24
5)	ORB[35] + MAGSAC[1]	1.69	0.97	3.34	1.58	1.15	1.40
6)	SPSG[7, 36] + RANSAC[9]	0.71	0.41	0.87	0.72	0.80	0.75
7)	SPSG[7, 36] + MAGSAC[1]	0.63	0.36	0.79	0.70	0.71	0.70
8)	LoFTR[39] + RANSAC[9]	1.44	0.56	2.70	1.36	1.05	1.52
9)	LoFTR[39] + MAGSAC[1]	1.39	0.55	2.57	1.33	1.05	1.41
10)	DHN[6]	2.87	1.51	4.48	2.76	2.62	3.00
11)	LocalTrans[37]	4.21	4.09	4.84	4.55	5.30	2.25
12)	IHN[3]	4.67	4.85	5.54	5.10	5.04	2.84
13)	RealSH[17]	0.34	0.22	0.35	0.44	0.42	0.29
14)	DMHomo[23]	0.31	0.19	0.33	0.40	0.38	0.28
15)	CAHomo[47]	0.88	0.73	1.01	1.03	0.92	0.70
16)	BasesHomo[44]	0.50	0.29	0.54	0.65	0.61	0.41
17)	HomoGAN[15]	0.39	0.22	0.41	0.57	0.44	0.31
18)	Ours*	0.33	0.20	0.33	0.41	0.39	0.30
19)	Ours	0.32	0.19	0.32	0.39	0.39	0.31

Table 1. The benchmark consists of 5 distinct scenarios, namely regular (RE), low-texture (LT), low-light (LL), small foreground (SF), and large foreground (LF). The point matching errors (PME) on the test set of CAHomo [47] are presented.

1)	Methods	AVG	RE	FOG	LL	RAIN	SNOW
2)	$\mathcal{I}_{3 \times 3}$	5.22	3.65	6.69	5.88	4.90	4.96
3)	SIFT[33]	2.82	0.60	2.43	7.09	0.61	3.37
4)	SPSG[7, 36]	3.07	3.99	1.57	6.88	0.79	2.16
5)	CAHomo[47]	2.81	2.02	2.03	4.56	2.84	2.61
6)	BasesHomo[44]	1.74	1.39	0.97	4.12	0.66	1.58
7)	Meshflow[27]	2.15	1.09	2.21	5.57	0.44	1.69
8)	HM_Mix[13]	4.35	1.02	4.03	8.75	1.53	6.42
9)	RANSAC-F[38]	3.26	2.81	3.14	5.12	2.21	3.04
10)	Ours	1.10	1.08	0.74	2.15	0.46	1.05

Table 2. To evaluate generalizability, we compute the end point errors (EPE) of pre-trained models from Table 1 on our proposed GHOF-Cam benchmark.

1)	Methods	AVG	RE	FOG	LL	RAIN	SNOW
2)	$\mathcal{I}_{3 \times 3}$	6.33	4.94	7.24	8.09	5.48	5.89
3)	SIFT[33]	4.80	0.59	4.47	12.10	0.62	6.20
4)	SPSG[7, 36]	4.47	3.54	2.21	10.66	0.83	5.10
5)	DHN[6]	6.61	6.04	6.02	7.68	6.99	6.32
6)	LocalTrans[37]	5.72	4.06	6.49	5.95	5.78	6.34
7)	IHN[3]	8.17	7.10	8.71	9.34	6.57	9.13
8)	RealSH[17]	1.72	1.60	0.88	4.42	0.43	1.28
9)	DMHomo[23]	1.75	0.64	0.85	4.16	0.39	2.74
10)	CAHomo[47]	3.87	4.10	3.84	6.99	1.27	3.17
11)	BasesHomo[44]	2.28	2.02	1.43	4.90	0.78	2.29
12)	HomoGAN[15]	1.95	1.73	0.60	3.95	0.47	3.02
13)	Ours	1.23	1.15	0.96	2.69	0.40	0.93

Table 3. To evaluate generalizability, we compute the point matching errors (PME) on the GHOF [24] test set using pre-trained models from Table 1.

[39], each evaluated with two outlier rejection techniques: RANSAC [9] and MAGSAC [1]. The second category includes supervised learning approaches: DHN [6], LocalTrans [37], IHN [3], RealSH [17], and DMHomo [23]. The third category comprises unsupervised methods: CAHomo

Method	AVG			RE			FOG			DARK			RAIN			SNOW		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
$\mathcal{I}_{3\times 3}$	24.05	0.7403	0.0836	21.06	0.6900	0.0750	26.57	0.7711	0.0821	25.70	0.8506	0.0785	21.53	0.5335	0.1411	25.37	0.8562	0.0412
GT-Homo	32.78	0.9187	0.0570	28.39	0.8697	0.0549	35.23	0.9508	0.0492	31.88	0.9405	0.0575	30.11	0.8511	0.1033	38.31	0.9814	0.0199
SIFT	28.44	0.9074	0.0781	29.23	0.9148	0.0545	29.42	0.9016	0.0768	27.37	0.9074	0.0982	30.00	0.8632	0.1055	26.16	0.9497	0.0558
SPSG	28.01	0.8697	0.0796	21.83	0.7593	0.0886	30.88	0.9049	0.0645	27.60	0.9019	0.0966	28.86	0.8270	0.1103	30.88	0.9556	0.0379
CAHomo	25.29	0.7837	0.0841	22.67	0.7341	0.0805	27.51	0.8048	0.0751	26.12	0.8743	0.0846	22.95	0.6130	0.1420	27.20	0.8924	0.0384
BasesHomo	29.61	0.9026	0.0672	25.08	0.8522	0.0666	31.06	0.9170	0.0627	30.05	0.9303	0.0702	29.58	0.8512	0.1071	32.30	0.9622	0.0292
MeshFlow	29.91	0.9239	0.0688	28.57	0.9216	0.0576	28.68	0.9280	0.0742	29.41	0.9254	0.0774	30.68	0.8747	0.1049	32.23	0.9700	0.0298
HM_Mix	25.77	0.8896	0.0882	26.09	0.8721	0.0596	26.56	0.8753	0.0882	26.43	0.9037	0.1002	28.20	0.8672	0.1107	21.58	0.9296	0.0820
RANSAC-F	26.04	0.8348	0.0890	26.09	0.8812	0.0665	29.22	0.8944	0.0801	27.29	0.9031	0.0923	21.68	0.5585	0.1495	25.90	0.9371	0.0566
Ours	32.09	0.9142	0.0575	27.08	0.8615	0.0558	34.17	0.9371	0.0512	32.36	0.9421	0.0565	30.52	0.8608	0.1021	36.35	0.9692	0.0218

Table 4. Quantitative comparison of different methods on various environmental conditions. We present three key perception metrics: PSNR (higher is better), SSIM (higher is better), and LPIPS (lower is better). Our method consistently outperforms existing approaches across all scenarios and metrics.

[47], BasesHomo [44], and HomoGAN [15].

For multi-plane camera motion modeling, we compare against both traditional approaches (MeshFlow [27], Homography Mixture [13], RANSAC-Flow [38]) and unsupervised deep methods (BasesMesh [30] and MeshHomoGAN [31]). Regarding pre-training, DHN, LocalTrans and IHN use the MS-COCO dataset [25], while other deep learning methods are pre-trained on CAHomo. Additional qualitative results and visual comparisons are available on: <https://lhaippp.github.io/CamFlow/>.

4.2.1. Quantitative Comparison.

Sparse camera motion. We evaluate using the points matching error (PME), which measures the average geometric distance between transformed source points and their corresponding ground-truth target points. Table 1 presents the quantitative performance of our method alongside various homography estimation approaches on the CAHomo test set, categorized as: feature-based methods (rows 2-9), supervised approaches (rows 10-14), and unsupervised ones (rows 15-17). The $\mathcal{I}_{3\times 3}$ baseline (row 1) represents the distance between point pairs without transformation.

Our CamFlow method achieves superior results across multiple categories, outperforming the leading unsupervised method, HomoGAN, by reducing PME by 17.95% (from 0.39 to 0.32). While feature-based methods like SIFT+RANSAC excel in regular (RE) scenes with abundant texture and keypoints, CamFlow surpasses them with a 36.67% improvement (reducing error by 0.11). To balance benchmark performance with generalization capability, we include Ours* (row 18), which represents an early-stopped training model that delivers better generalizability across datasets, even though it slightly underperforms our fully-trained model (row 19) on the CAHomo benchmark.

In small foreground (SF) and large foreground (LF) scenarios, where dynamic objects disrupt camera motion estimation, our probabilistic loss and confidence masking yield lower PMEs compared to methods like CAHomo and HomoGAN, which employ explicit outlier rejection masks for robustness. In low-texture (LT) and low-light (LL) conditions, learning-based approaches generally exhibit superior

resilience due to their keypoint-free strategies, particularly among unsupervised techniques. However, these scenes often contain homogeneous regions [41], occupying large image areas and reducing photometric differences, which limit unsupervised methods’ effectiveness. By contrast, CamFlow excels in LT and LL.

Generalization experiment. Evaluating cross-dataset generalization presents a significant challenge in motion estimation, particularly for applications where real-world video often differs substantially from training data. We assess CamFlow’s generalization capabilities in two ways: (1) for dense camera motion, we evaluate against traditional single-plane, multi-planar, and deep learning approaches using our proposed GHOF-Cam benchmark (Table 2 and Table 4); and (2) for sparse camera motion, we compare against traditional and homography-based methods on the original GHOF benchmark [24] (Table 3).

Table 2 presents End Point Error (EPE) results on the GHOF-Cam benchmark, which represents ground-truth camera motion. We evaluate against homography-based methods (rows 3-5) and non-single-plane approaches (rows 6-9). As illustrated in Fig. 3, we classify BasesHomo [44] as a multi-plane rather than single-homography method. The results demonstrate that CamFlow outperforms competing methods across nearly all categories, showcasing exceptional zero-shot capability in capturing non-linear camera motion patterns. Additionally, Table 4 presents perceptual quality metrics (PSNR, SSIM, and LPIPS) across various conditions. We also include ground-truth homography results for reference (GT-Homo). Our method, CamFlow, achieves state-of-the-art performance across multiple categories, consistently outperforming all baselines, particularly in challenging conditions such as snowy scenes. Notably, CamFlow approaches the performance of ground-truth homography, with only a 0.69 dB difference in PSNR, 0.0045 in SSIM and 0.0005 in LPIPS, while significantly surpassing the second-best method, MeshFlow. These results support CamFlow’s potential for applications requiring high-quality visual alignment.

Table 3 presents compelling evidence of CamFlow’s generalization capability on the GHOF benchmark. Our

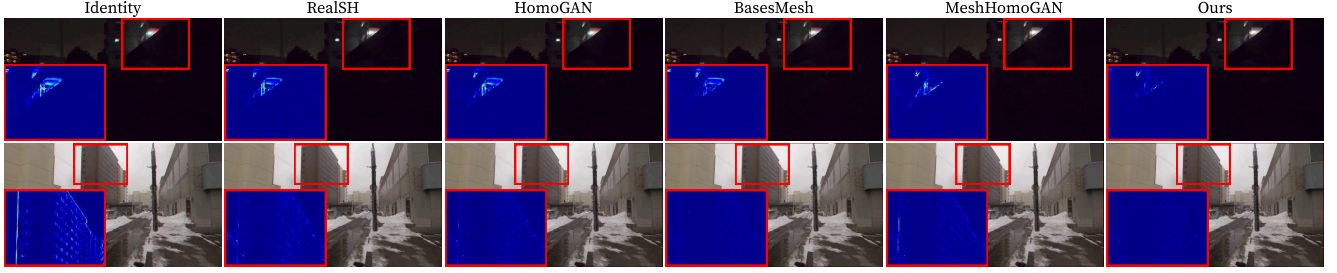


Figure 5. Qualitative results of CamFlow and methods from each category (i.e., supervised, unsupervised, and multi-homography) on the CAHomo testset [47]. The images are generated by superimposing the warped source images on the target image. Error-prone regions are highlighted with red boxes, which are further converted into alignment heatmaps for better distinction when zoomed in.

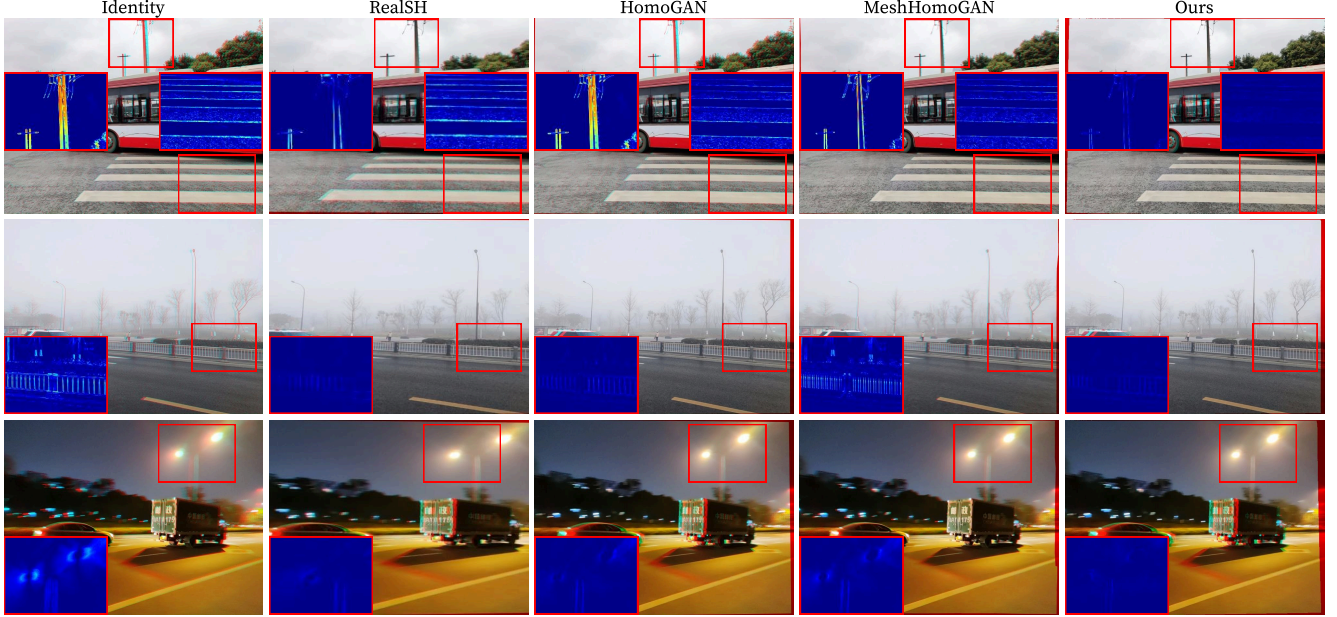


Figure 6. Qualitative results of our method and the best-performing generalizable methods from learning-based categories (i.e., supervised, unsupervised, and multi-homography) on the GHOF testset [24]. The examples are arranged from top to bottom and include RE, Fog, and Dark. The images are generated by superimposing the warped source images on the target image. Error-prone regions are highlighted with red boxes, which are further converted into alignment heatmaps for better distinction when zoomed in.

method achieves the lowest average PME (1.23), representing a 28.5% improvement over the previous best supervised method (RealSH at 1.72) and a 36.9% improvement over the leading unsupervised approach (HomoGAN at 1.95). This performance advantage extends across all environmental conditions, with particularly significant gains in challenging scenarios. In low-light (LL) conditions, CamFlow reduces error by 39.1% compared to RealSH (from 4.42 to 2.69), while in snowy (SNOW) environments, it achieves a remarkable 69.2% error reduction compared to HomoGAN (from 3.02 to 0.93). These results confirm that CamFlow enables robust zero-shot transfer to unseen datasets, even under diverse and challenging environmental conditions.

4.3. Qualitative Comparison

Fig. 5 and 6 present qualitative results of CamFlow alongside competing methods on the CAHomo and GHOF test

sets. Dynamic visualizations are available on our project page. For visualization, we employ red-blue ghosting and alignment heat maps following [18]. After transforming the source frame using estimated camera flow, misaligned regions appear as red and blue ghosting. We highlight specific areas with red boxes and generate alignment heat maps where brighter regions indicate poorer alignment.

In both figures, the “Identity” column shows source and target images overlaid without warping. For CAHomo (Fig. 5), we focus on evaluating background camera motion modeling capabilities. We selected challenging test cases featuring extremely low-light conditions (first row) and sophisticated motion patterns (parallax and depth variation), comparing against leading methods from three categories: supervised (RealSH), unsupervised (HomoGAN), and deep meshflow (BasesMesh, MeshHomoGAN). CamFlow achieves superior alignment in background regions.

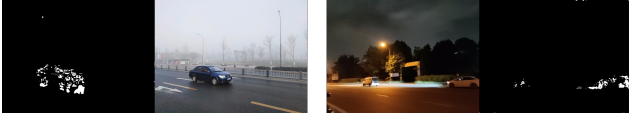


Figure 7. Uncertainty masks and corresponding original images. The brighter the mask, the higher the uncertainty. The masks effectively highlight dynamic objects, indicating regions where camera motion estimation is less reliable.

The GHOF benchmark (Fig. 6) highlights CamFlow’s advantages in handling complex camera motion and its zero-shot generalization capabilities. The first row shows a scene with parallax and foreground motion that challenges homography-based methods and reduces meshflow accuracy. The second row demonstrates CamFlow’s generalization to foggy conditions unseen in training data, where depth variation complicates motion estimation. The last row presents an extreme case with motion blur, large parallax, and dynamic objects. In all scenarios, CamFlow delivers robust results where competing methods struggle, confirming its effectiveness in modeling complex camera motion patterns across diverse environmental conditions.

Foreground Masks. Fig. 7 illustrates the uncertainty masks. We observe that the network effectively identifies dynamic objects and assigns higher uncertainty to regions containing these objects. This design helps focus the learning process on areas where motion is most relevant, while reducing noise from dynamic object regions, thereby enhancing the accuracy of camera motion estimation.

4.4. Ablation Studies

4.4.1. Motion Basis

	CAHomo	GHOF	GHOF-Cam	Params	Inference Time
8 Bases	0.37	1.68	1.45	2.658M	76.42ms
12 Bases	0.36	1.54	1.23	2.658M	75.38ms
24 Bases	0.33	1.23	1.10	2.660M	79.63ms
200 Bases	0.33	1.27	1.07	2.677M	99.28ms

Table 5. Performance comparison under different numbers of motion bases on three benchmarks. Results demonstrate the effectiveness of combining physical and stochastic motion bases, with 24 bases providing optimal balance between accuracy and computational efficiency.

In Table 5, we evaluate with varying numbers of motion bases across three benchmarks: CAHomo (trained) and GHOF/GHOF-Cam (zero-shot). The average results demonstrate that: 1) Increasing physical bases from 8 to 12 improves performance across all benchmarks; 2) Introducing additional hybrid bases (24 total) yields further enhancements, particularly for generalization (GHOF and GHOF-Cam). Notably, while expanding to 200 bases provides marginal improvements on some benchmarks, it increases inference time by 24.7%. We therefore adopt 24 bases in our final model as the optimal balance between ac-

curacy and computational efficiency.

4.4.2. Hybrid Probabilistic Loss

ℓ_{NLL_m}	ℓ_{NLL_p}	CAHomo	GHOF	GHOF-Cam
✓		0.41	2.21	2.13
	✓	0.36	1.58	1.42
✓	✓	0.33	1.23	1.10

Table 6. Ablation study of different loss function combinations across three benchmarks. Results demonstrate that the hybrid approach combining motion loss and photometric loss achieves superior performance compared to individual loss components.

Table 6 evaluates our probabilistic loss components across three benchmarks. Using only motion loss yields limited generalization performance, because pseudo motion labels provide approximate supervision. The photometric loss alone performs substantially better, particularly on zero-shot datasets, consistent with findings from prior unsupervised methods. However, our hybrid approach combining both losses achieves the best results across all benchmarks, with improvements on GHOF-Cam (22.5% error reduction compared to photometric-only). We believe motion labels provide coarse guidance while photometric loss enables fine-grained refinement, resulting in more accurate and generalizable camera motion estimation.

5. Conclusion

CamFlow presents a novel motion representation for modeling 2D camera motion using a hybrid motion basis approach. We identify a fundamental issue: superposing homographies by simply adding flow fields introduces non-linear interactions, contradicting the assumption that homographies can be expressed as linear combinations of 8 basis flows. By expanding the previous 8-dimensional motion basis into a higher-dimensional space with both physical and stochastic motion bases, CamFlow effectively captures complex, non-linear motion patterns. The proposed probabilistic loss function enhances training stability. Our newly introduced GHOF-Cam benchmark demonstrates that CamFlow surpasses state-of-the-art homography and meshflow methods, showcasing superior robustness and generalization. We hope this work opens new avenues for camera motion modeling in video processing applications. Code-base, pre-trained models, and benchmark data are released at <https://lhaippp.github.io/CamFlow/>.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants No. 62372091 and No. 62031009, and in part by the Hainan Province Key R&D Program under grant No. ZDYF2024(LALH)001. We would also like to thank Hao Xu, Mingbo Hong, Hai Jiang, Xinglong Luo, Nianjin Ye, and the anonymous reviewers for their valuable discussions, constructive suggestions, and helpful feedback.

References

- [1] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: marginalizing sample consensus. In *Proc. CVPR*, pages 10197–10205, 2019. 5
- [2] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proc. CVPR*, pages 1304–1312, 2020. 2
- [3] Si-Yuan Cao, Jianxin Hu, Zehua Sheng, and Hui-Liang Shen. Iterative deep homography estimation. In *Proc. CVPR*, pages 1879–1888, 2022. 2, 5
- [4] Che-Han Chang, Chun-Nan Chou, and Edward Y Chang. Clkn: Cascaded lucas-kanade networks for image alignment. In *Proc. CVPR*, pages 2213–2221, 2017. 2
- [5] Padraig Cunningham and Sarah Jane Delany. K-nearest neighbour classifiers-a tutorial. *ACM Computing Surveys*, 54(6):1–25, 2021. 2
- [6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 2, 5
- [7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proc. CVPRW*, pages 224–236, 2018. 2, 5
- [8] Tianjiao Ding, Yunchen Yang, Zhihui Zhu, Daniel P Robinson, René Vidal, Laurent Kneip, and Manolis C Tsakiris. Robust homography estimation via dual principal component pursuit. In *Proc. CVPR*, pages 6080–6089, 2020. 2
- [9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 5
- [10] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *Proc. CVPR*, pages 49–56, 2011. 2
- [11] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proc. CVPR*, pages 3369–3378, 2018. 4
- [12] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Proc. NIPS*, 27, 2014. 2
- [13] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *Proc. ICCP*, pages 1–8, 2012. 2, 5, 6
- [14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 1
- [15] Mingbo Hong, Yuhang Lu, Nianjin Ye, Chunyu Lin, Qijun Zhao, and Shuaicheng Liu. Unsupervised homography estimation with coplanarity-aware gan. In *Proc. CVPR*, pages 17663–17672, 2022. 2, 4, 5, 6
- [16] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proc. ECCV*, pages 652–667, 2018. 4
- [17] Hai Jiang, Haipeng Li, Songchen Han, Haoqiang Fan, Bing Zeng, and Shuaicheng Liu. Supervised homography learning with realistic dataset generation. In *Proc. ICCV*, pages 9806–9815, 2023. 2, 5
- [18] Hai Jiang, Haipeng Li, Yuhang Lu, Songchen Han, and Shuaicheng Liu. Semi-supervised deep large-baseline homography estimation with progressive equivalence constraint. In *Proc. AAAI*, pages 1024–1032, 2023. 7
- [19] Dewi Endah Kharismawati, Hadi Ali Akbarpour, Rumana Aktar, Filiz Bunyak, Kannappan Palaniappan, and Toni Kazic. Cornet: Unsupervised deep homography estimation for agricultural aerial imagery. In *Proc. ECCV*, pages 400–417, 2020. 2
- [20] 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 *Proc. ICCV*, pages 4015–4026, 2023. 5
- [21] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In *Proc. CVPR*, pages 7652–7661, 2020. 2
- [22] Haipeng Li, Kunming Luo, and Shuaicheng Liu. Gyroflow: Gyroscope-guided unsupervised optical flow learning. In *Proc. ICCV*, pages 12869–12878, 2021. 4
- [23] Haipeng Li, Hai Jiang, Ao Luo, Ping Tan, Haoqiang Fan, Bing Zeng, and Shuaicheng Liu. Dnmomo: Learning homography with diffusion models. *ACM Transactions on Graphics (TOG)*, 43(3):1–16, 2024. 2, 5
- [24] Haipeng Li, Kunming Luo, Bing Zeng, and Shuaicheng Liu. Gyroflow+: Gyroscope-guided unsupervised deep homography and optical flow learning. *International Journal of Computer Vision (IJCV)*, 132(6):2331–2349, 2024. 2, 5, 6, 7
- [25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, pages 740–755, 2014. 6
- [26] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013. 1, 2
- [27] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *Proc. ECCV*, pages 800–815, 2016. 2, 5, 6
- [28] 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 (PAMI)*, pages 1–14, 2022. 2
- [29] Shuaicheng Liu, Nianjin Ye, Chuan Wang, Jirong Zhang, Lanpeng Jia, Kunming Luo, Jue Wang, and Jian Sun. Content-aware unsupervised deep homography estimation and its extensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 45(3):2849–2863, 2022. 2
- [30] 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 (PAMI)*, 45(6):7885–7899, 2023. 2, 6
- [31] Shuaicheng Liu, Mingbo Hong, Yuhang Lu, Nianjin Ye, Chunyu Lin, and Bing Zeng. Unsupervised global and local

- homography estimation with coplanarity-aware gan. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2024. 1, 2, 6
- [32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. ICCV*, pages 10012–10022, 2021. 2
- [33] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 1, 2, 5
- [34] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters (RA-L)*, 3(3): 2346–2353, 2018. 2
- [35] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proc. ICCV*, pages 2564–2571, 2011. 2, 5
- [36] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proc. CVPR*, pages 4938–4947, 2020. 5
- [37] Ruizhi Shao, Gaochang Wu, Yuemei Zhou, Ying Fu, Lu Fang, and Yebin Liu. Localtrans: A multiscale local transformer network for cross-resolution homography estimation. In *Proc. ICCV*, pages 14890–14899, 2021. 2, 5
- [38] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. In *Proc. ECCV*, pages 618–637, 2020. 2, 5, 6
- [39] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proc. CVPR*, pages 8922–8931, 2021. 5
- [40] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proc. CVPR*, pages 11016–11025, 2019. 2
- [41] Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and correspondences. In *Proc. CVPR*, pages 6258–6268, 2020. 6
- [42] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *Proc. CVPR*, pages 5714–5724, 2021. 2, 4
- [43] Nianjin Ye, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Jue Wang, and Yongqing Cui. Deepmeshflow: Content adaptive mesh deformation for robust image registration. *arXiv preprint arXiv:1912.05131*, 2019. 2
- [44] Nianjin Ye, Chuan Wang, Haoqiang Fan, and Shuaicheng Liu. Motion basis learning for unsupervised deep homography estimation with subspace projection. In *Proc. ICCV*, pages 13117–13125, 2021. 1, 2, 3, 4, 5, 6
- [45] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Proc. ECCV*, pages 467–483, 2016. 2
- [46] Julio Zaragoza, Tat-Jun Chin, Michael S Brown, and David Suter. As-projective-as-possible image stitching with moving dlt. In *Proc. CVPR*, pages 2339–2346, 2013. 2
- [47] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *Proc. ECCV*, pages 653–669, 2020. 2, 4, 5, 6, 7