

Adapting Dense Matching for Homography Estimation with Grid-based Acceleration

Kaining Zhang¹ Yuxin Deng¹ Jiayi Ma^{1*} Paolo Favaro²

¹Wuhan University, China ²University of Bern, Switzerland

{zkn707196, acuo.dyx, jyama2010}@gmail.com paolo.favaro@unibe.ch

Abstract

Current deep homography estimation methods are typically constrained to processing low-resolution image pairs due to network architecture and computational limitations. For high-resolution images, downsampling is often required, which can greatly degrade estimation accuracy. In contrast, image matching methods, which match pixels and compute homography from correspondences, provide greater resolution flexibility. So in this work, we revisit the traditional image matching paradigm for homography estimation and propose GFNet, a Grid Flow regression Network that adapts the high-accuracy dense matching framework for homography estimation while enhancing efficiency through a grid-based strategy—estimating flow only over a coarse grid by leveraging homography’s global smoothness. We demonstrate the effectiveness of GFNet on a wide range of experiments on multiple datasets, including the common scene MSCOCO, multimodal datasets VIS-IR and GoogleMap, and the dynamic scene VIRAT. Notably, on 448×448 GoogleMap, GFNet achieves an improvement of +13.5% in auc@3 while reducing MACs by ~47% compared to the SOTA dense matching method. Additionally, it shows a 1.8× improvement in auc@3 over the SOTA deep homography method. Code is available at <https://github.com/KN-Zhang/GFNet>.

1. Introduction

Homography estimation is the task of determining the transformation that aligns two planes. This is a basic low-level computer vision task widely used in various downstream applications, including image/video stitching [47], image fusion [38], GPS-denied UAV localization [36, 39], stereo vision [19], and planar object tracking [24].

Deep homography estimation methods, which typically parameterize homography using corner offsets or the homography matrix, have demonstrated high accuracy on low-

resolution images. However, their scalability to higher resolutions is constrained by limitations in (1) *network architecture* and (2) *computational capacity*. Specifically, 4-point-based methods [5, 6, 11, 20, 30, 49] extract matching information from image pairs and aggregate it to regress corner offsets, followed by homography estimation using DLT [17]. The aggregation process is typically implemented through a fixed number of large-kernel pooling operations. Consequently, the network’s downsampling factor is predefined and constrained by the kernel sizes, inherently limiting the input image resolution—typically to 128×128 . Matrix-based methods [7, 43–45] explicitly use the IC-LK iterator [1] with deep features to refine the homography matrix. This approach requires multiple iterations to achieve feature-metric alignment across all pixels, resulting in significant computational and memory overhead. As a result, current methods are forced to work with inputs at low resolution. Unfortunately, doing so has also a direct negative impact on the accuracy of the estimated homography.

Image matching methods, which first find correspondences and then estimate homography with robust estimators [16, 17], offer greater flexibility for high-resolution inputs. However, they are often criticized for their limited effectiveness in low-texture scenarios [23], which are common in homography applications. Recently, this issue has been well-addressed by the powerful dense matching methods [13, 14, 35], which find correspondences for every pixel. Yet, this strong performance comes with non-negligible computational costs. For example, training the SOTA dense matcher RoMa [14] on a 24GB GPU only supports a batch size of 1 for 560×560 images. This heavy resource demand arises from attempting to resolve every possible match between images, which is redundant for homography estimation, as it only has 8 degrees of freedom.

In this paper, we aim to solve the resolution issue of deep homography methods by introducing the dense matching framework, while tackling the computational challenges of dense matching by considering the globally smooth nature of homography transformations. This leads to Grid Flow regression Network (GFNet). As shown in Figure 1, GFNet

*Corresponding author.

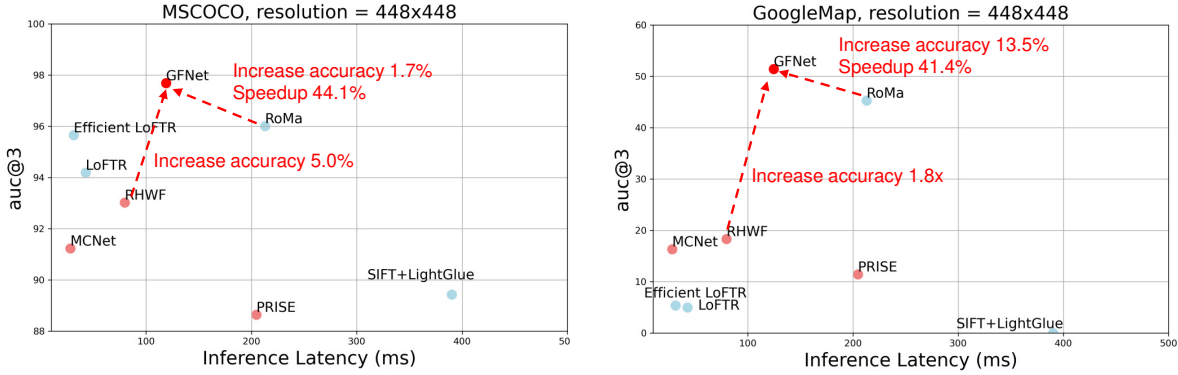


Figure 1. Homography estimation results of GFNet and existing deep homography estimation methods (red spot) and image matching methods (blue spot) on MSCOCO (natural) and GoogleMap (multimodal). We test on an RTX3090 with batchsize = 1.

achieves superior accuracy over deep homography estimation methods and outperforms the SOTA dense matching method RoMa with higher efficiency on both MSCOCO and GoogleMap. Key contributions include:

- We incorporate pre-trained self-supervised learning features of DINOv2 [27] with a lightweight feature pyramid network to construct robust multi-scale features. This fusion solves the low-resolution limitation of DINOv2, allowing our model to benefit from its strong cross-domain feature matching capabilities across multiple scales.
- We sparsify pixel-level dense matching to a coarse grid prediction, greatly reducing computational costs while keeping the high accuracy of dense matching. Compared to previous deep homography estimation methods, our grid-flow representation provides greater resolution flexibility and enhanced accuracy.
- We introduce a novel synthetic data generation method for self-supervised learning to improve generalization and an iterative flow regression approach to prevent suboptimal multi-scale flow optimization in challenging scenes.

2. Related Works

Deep homography estimation methods are generally categorized into 4-point-based and matrix-based approaches based on how they parameterize homography. 4-point-based methods employ neural networks to regress corner offsets and estimate homography with DLT. The first such approach, DHN [11], uses a VGG-style network to process a stacked image pair and output corner offsets. Subsequent methods improve accuracy by cascading multiple networks for progressively prediction [15, 20, 30]. More recently, end-to-end learning with a single network has become the preferred approach due to better accuracy, facilitated by techniques such as iterative estimation in IHN [5], the recurrence strategy in RHWF [6], and the multi-scale iterative framework in MCNet [49]. However, these 4-point-based methods are constrained to low-resolution in-

puts (typically 128×128) due to the network’s inherent inflexibility. Matrix-based methods employ neural networks to extract features and leverage non-linear optimization to refine the homography matrix, achieving feature-metric alignment. These methods enhance estimation accuracy either by ensuring a good initialization[43] or by improving the convergence basin for optimization[7, 44, 45]. These matrix-based methods are also limited to small-resolution images due to their high computational overhead. There are also alternative parameterizations, such as the homography flow proposed in BaseHomo [23, 40]. It does not impose a resolution limitation but is restricted to small-baseline scenes, which significantly limits its applicability.

Image matching methods can be categorized into sparse, semi-dense, and dense methods. Sparse methods rely on keypoint detection and description, followed by matching the descriptors [10, 12, 22, 28, 29]. While efficient, they often struggle with accuracy in textureless regions, which are common in homography estimation tasks [43]. Semi-dense methods degrade less in such areas by bypassing keypoint detection. Instead, they perform global matching at a coarse level, where initial coarse matches that pass the mutual nearest neighbor test are then refined at a finer level. Regarding the dense methods, they can handle textureless situations well by predicting matches for every pixel. Although they achieve impressive accuracy, they require substantial computational resources in terms of both time and memory, which limits their applicability in resource-constrained scenarios [13, 14, 34, 35].

Common challenges in homography estimation are twofold and arise from practical application requirements, according to recent literature [5, 20, 25, 30, 41, 43, 45]. The first one is photometric inconsistency caused by changes in illumination or modality, such as images taken at different times, with different sensors, or in different types [18, 48]. The second challenge is whether methods can handle violations of the homography assumption due to dynamic occlusions, which frequently occur in real-world scenarios.

3. Method

We aim to predict an accurate homography transformation \mathbf{H} that spatially aligns a source image I_S with a target image I_T . Rather than directly regressing the global homography, we follow the dense matching paradigm but estimate the flow over a regular grid and utilize the resulting correspondences to compute the homography. This process begins by extracting multi-scale features from both images, which are then used to predict grid flow across multiple scales.

3.1. Multi-scale Feature Extraction

We use DINOv2 to enhance the feature representation capability [42]. Specifically, DINOv2 divides the input image into 14×14 patches and generates a feature vector for each patch. As these patch descriptors are typically high-dimensional, we apply a linear projection layer to reduce their dimensionality, setting it to 64 in our implementation. To further enrich the features with cross-view information, which is crucial for matching, we add a normalized 2D positional encoding [4, 8] and pass the features through stacked cross-attention layers. The resulting feature is denoted as $\mathcal{F}^1 \in \mathbb{R}^{64 \times \lfloor \frac{H}{14} \rfloor \times \lfloor \frac{W}{14} \rfloor}$, where H and W are the height and width of the original image, $\lfloor \cdot \rfloor$ is the round down operation.

Since DINOv2 produces relatively low resolution features, relying solely on these features for homography estimation would lead to limited matching accuracy [3]. To address this issue, we introduce a Feature Pyramid Network (FPN) to construct multi-scale features, ranging from the original image size down to a $1/8$ scale. We keep the network lightweight by configuring the channel numbers of each layer to [8, 16, 32, 64]. Furthermore, to improve the representation capability of the lightweight FPN, we bilinearly upsample \mathcal{F}^1 to the $1/8$ scale and merge it with the $1/8$ scale features produced by the encoding stage of the FPN. The final output features used for homography estimation consist of five scales, denoted as $\{\mathcal{F}^l\}_{l=1,2,3,4,5}$, corresponding to spatial sizes of $1/14, 1/8, 1/4, 1/2$, and 1 relative to the original image, as illustrated in Figure 2.

3.2. Background: Dense Matching

We derive our grid flow regression based on the framework of the SOTA dense matcher RoMa [14]. Here is a brief introduction to it. RoMa aims to find pixel-wise correspondences between two images $I_S, I_T \in \mathbb{R}^{H \times W \times 3}$ by estimating a dense displacement field, or flow, $\mathbf{w} \in \mathbb{R}^{H \times W \times 2}$. Given the pixel coordinates in I_S as $\mathbf{x} \in \mathbb{R}^{H \times W \times 2}$, the relationship $\mathbf{y} = \mathbf{x} + \mathbf{w}$ denotes the corresponding pixel positions in I_T that align with those in I_S , representing the same physical locations in the scene. \mathbf{w} is predicted in a multi-scale manner, from the coarsest ($l = 1$) to the finest:

$$\mathbf{w}^l = \text{up}(\mathbf{w}^{l-1}) + \Delta \tilde{\mathbf{w}}^l, \Delta \tilde{\mathbf{w}}^l = \text{decoder}_{\theta}^l \left(\mathcal{F}_S^l, \mathcal{F}_T^l, \text{up}(\mathbf{w}^{l-1}) \right), \quad (1)$$

where θ represents learnable parameters, \mathbf{w}^0 is an all-zero field, and $\text{up}(\cdot)$ is the bilinear upsampling operation. There is a feature correlation layer included in each decoder $\text{decoder}_{\theta}^l(\cdot)$, which is computed as:

$$c(\mathcal{F}_S^l, \mathcal{F}_T^l; r^l) = \mathcal{F}_S^l[\mathbf{x}^l] \odot \mathcal{F}_T^l[\mathbf{x}^l + \text{up}(\mathbf{w}^{l-1}) + \delta], \quad (2)$$

where $\mathbf{x}^l \in \mathbb{R}^{H^l \times W^l \times 2}$ is the pixel coordinates in \mathcal{F}_S^l , $[\cdot]$ is bilinear interpolation, \odot is the dot product, and $\delta \in [-2r^l - 1, 2r^l + 1] \times [-2r^l - 1, 2r^l + 1]$ defines a local window with radius r^l . This operation leads to a 4D tensor of shape $\mathbb{R}^{H^l \times W^l \times (2r^l+1) \times (2r^l+1)}$, which captures visual similarity within a local neighborhood. While this correlation step is crucial for producing accurate results, it also introduces the main computational bottleneck in RoMa.

3.3. Grid Flow Regression

Grid-based strategy for efficiency. To alleviate the computational burden of Eq. 2, previous works typically reduce the radius r as the scale increases [30, 34, 49]. However, this saving is limited for high-resolution images, where the first two dimensions of $c(\mathcal{F}_S^l, \mathcal{F}_T^l; r^l)$ dominate the computational load. So to further minimize computational costs, we leverage the global smooth pattern of homography transformations and focus on learning a sparser representation of \mathbf{w} to replace the original dense, pixel-wise one.

To achieve this, we sparsify \mathbf{w} through a grid-based strategy. We redefine $\mathbf{x} \in G \times G \times 2$ to represent the coordinates of a regular grid on I_S with $G \times G$ being the grid size. Our goal is to estimate a grid-based displacement field $\mathbf{w} \in G \times G \times 2$ and a confidence map $\mathbf{m} \in G \times G \times 1$ that reflects the reliability of each flow estimate. Our grid subsampling strategy is as follows: we use features predicted by the FPN to refine and proportionally scale the grid flow, with the patch matching results from DINOv2 serving as initialization. Under this strategy, the grid sizes from $l = 1$ to $l = 5$ are $\frac{H}{14} \times \frac{W}{14}, \frac{H}{14} \times \frac{W}{14}, \frac{H}{7} \times \frac{W}{7}, \frac{2H}{7} \times \frac{2W}{7}, \frac{4H}{7} \times \frac{4W}{7}$, respectively.

Using DINOv2 for initialization is crucial, as it excels at semantic matching, providing a strong starting point for refinement, as shown in our experiments. This implies that the core idea of our method is to refine and scale the patch matching results provided by the foundation model. Therefore, other visual foundation models with strong semantic information could also be considered for this purpose.

Direct regression vs. Iterative regression. The decoder $(\mathcal{F}_S^l, \mathcal{F}_T^l, \text{up}(\mathbf{w}^{l-1}))$, which estimates the flow only once at each scale (as shown in Eq. 1), is referred to as direct regression. This is the default regression paradigm in dense matching. However, as the convergence basin shrinks with increasing scale, obtaining an optimal estimation with this approach requires that the estimate from the previous level falls within the convergence basin of the next level. If this condition is not met, the result will be suboptimal, as

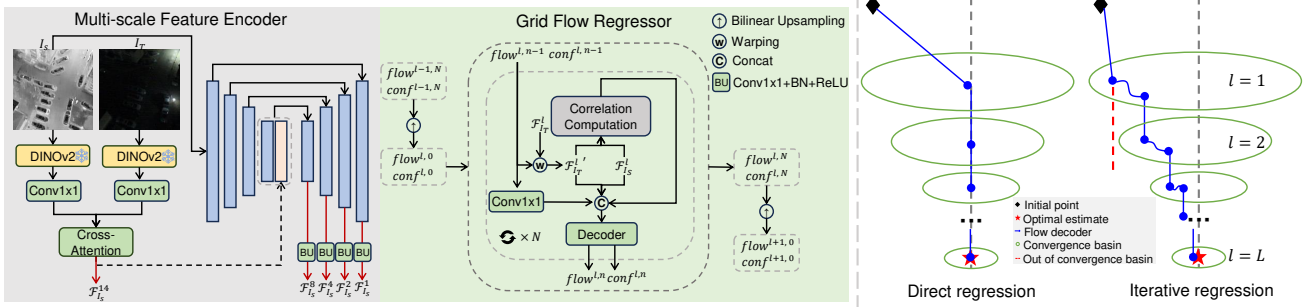


Figure 2. Left: the overview of GFNet. Using multi-scale features, GFNet regresses the grid flow and confidence map progressively from the coarsest scale to the finest. Right: flow optimization across multiple scales.

illustrated in the right plot of Figure 2. The precondition for obtaining a reliable estimate at each scale is that \mathcal{F}_S^l and \mathcal{F}_T^l must be well-aligned with $\text{up}(\mathbf{w}^{l-1})$. This alignment is relatively easy to achieve for images within the similar modality. But in cases where images have significant representational gaps, it becomes challenging for the decoder to make an accurate estimate in a single pass. To address this issue, we propose an iterative regression strategy to improve the reliability of each grid flow. Our formulation for estimating the grid flow and its confidence map is given by:

$$\begin{aligned} \mathbf{w}^{l,n} &= \mathbf{w}^{l,n-1} + \Delta \tilde{\mathbf{w}}^{l,n}, \\ \mathbf{m}^{l,n} &= \mathbf{m}^{l,n-1} + \Delta \tilde{\mathbf{m}}^{l,n}, \\ \Delta \tilde{\mathbf{w}}^{l,n}, \Delta \tilde{\mathbf{m}}^{l,n} &= \text{decoder}_{\theta}^l(\mathcal{F}_S^l, \mathcal{F}_T^l, \mathbf{w}^{l,n-1}), \end{aligned} \quad (3)$$

where $n \in [1, N]$ represents the iteration number, with N total iterations per scale. When $N = 1$, Eq. 3 reduces to direct regression. All predictions are made at the image scale, and the flow and confidence map are updated between scales as $\mathbf{w}^{l+1,0}, \mathbf{m}^{l+1,0} = \text{up}(\mathbf{w}^{l,N}), \text{up}(\mathbf{m}^{l,N})$.

Multiscale+iteration is a common way to learn iterative updates, mimicking first-order optimizers. The key difference between prior methods and GFNet lies in the variable being optimized. For example, RAFT [33] optimizes optical flow, IHN [5] and MCNet [49] optimize corner offsets, while GFNet optimizes grid flow. GFNet is more efficient than pixel-based flow for homography and more flexible than corner offsets across resolutions. Besides, the design of flow decoders may vary across different methods.

Adaptive resolution recurrence. During training, images are consistently resized to a fixed resolution before being fed into the network to maintain a uniform grid shape. To handle varying resolutions, we adopt a simple recurrent strategy similar to [34] during inference:

$$\begin{aligned} \mathbf{w}_1, \mathbf{m}_1 &= \text{GFNet}(I_S, I_T; \mathbf{w}_0), \\ \mathbf{w}_0, \mathbf{m}_0 &= \text{GFNet}(I'_S, I'_T; \mathbf{0}), \\ I'_S, I'_T &= \text{resize}(I_S, I_T), \end{aligned} \quad (4)$$

where I_S and I_T are the original images, I'_S and I'_T are resized to match the training resolution, and $\text{GFNet}(\cdot; \mathbf{w})$ denotes GFNet is initialized with \mathbf{w} . The linear combination of \mathbf{m}_0 and \mathbf{m}_1 is regressed as the final confidence map. For images with resolutions lower than the training resolution, we use the training resolution recurrently, while for higher-resolution images, we apply the recurrence at a higher resolution. Computing $\mathbf{w}_0, \mathbf{m}_0$ spans all scales, whereas the computation of $\mathbf{w}_1, \mathbf{m}_1$ begins at the 1/8 scale, skipping the global initialization at 1/14 scale.

Besides, a symmetric approach is used during inference. This involves swapping the input sequence of I_S and I_T to compute the flow for the regular grid on I_T . The resulting flows, $\mathbf{w}_{S \rightarrow T}$ and $\mathbf{w}_{T \rightarrow S}$, along with their corresponding confidence maps $\mathbf{m}_{S \rightarrow T}$ and $\mathbf{m}_{T \rightarrow S}$, are then combined to form the final matches. From these, K matches are selected by thresholding the confidence maps and sampling according to a uniform distribution rule [14]. Then RANSAC [16] is used to compute homography.

3.4. Loss Function

For flow estimation, we use L_2 loss between the ground truth grid flow \mathbf{w}_{gt} and the predicted flow \mathbf{w} at every scale $l \in [1, L]$ and iteration $n \in [1, N]$:

$$L_{flow} = \sum_{l=1}^L \frac{\mathbf{m}_{gt}^l}{G^l \times G^l} \sum_{n=1}^N \lambda^{(N-n)} \rho(\|\mathbf{w}^{l,n} - \mathbf{w}_{gt}^l\|_2), \quad (5)$$

where G^l represents the grid size at scale l , and $\lambda \in (0, 1)$ assigns higher weights to later iterations, similar to the ‘‘learning to optimize’’ approach [33]. ρ is a robust cost function that mitigates the impact of outliers [2], as employed in RoMa [14]. $\mathbf{m}_{gt} \in \{0, 1\}$ is a binary mask that indicates which pixels in I_S have correspondences in I_T .

For the confidence map, we learn it using a Binary Cross-Entropy (BCE) loss:

$$L_{conf} = \sum_{l=1}^L \frac{1}{G^l \times G^l} \sum_{n=1}^N \lambda^{(N-n)} \text{BCE}(\mathbf{m}^{l,n}, \mathbf{m}_{gt}^l). \quad (6)$$

$L = L_{flow} + \alpha L_{conf}$ is the total loss, where α is a hyper-parameter balancing the two terms. All predicted $\mathbf{w}^{l,n}$ are in image space, while $\mathbf{m}^{l,n}$ are in log space.

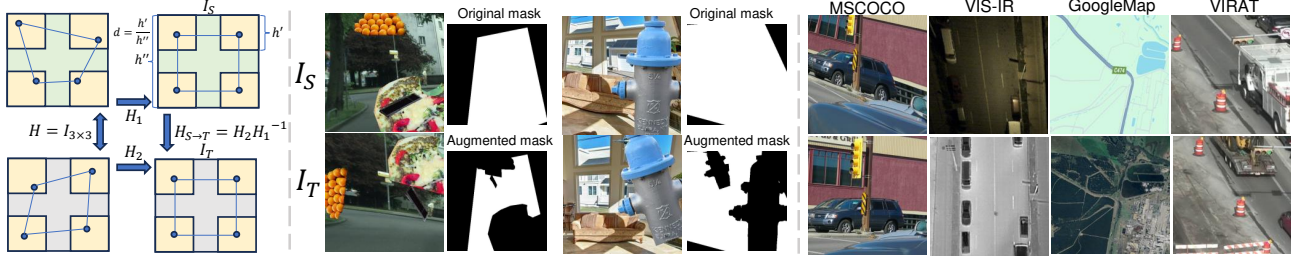


Figure 3. Left: data generation. Middle: training samples. Right: test samples. In the mask, white for 1, while black for 0.

3.5. Data Generation for Self-supervised Learning

Composite homography. Following prior works [5–7, 11, 15, 20, 30, 43–45], as shown in Figure 3, we generate synthetic homography datasets for both training and testing. Specifically, we first define a deformation area by selecting four squares at the image’s corners, with a deformation ratio $d \in (0, 0.5)$. Here, d is the ratio between the deformation area height and the image height. From each of these squares, four random points are selected and transformed to the center of the respective squares, yielding a homography matrix. This process is performed independently on each image in the pair, producing two homography matrices, \mathbf{H}_1 and \mathbf{H}_2 . The images I_S and I_T are then generated by cropping the deformed images, with the centers of the deformation areas serving as the corners of the squares. The homography between I_S and I_T is then computed as $\mathbf{H}_{S \rightarrow T} = \mathbf{H}_2 \mathbf{H}_1^{-1}$. Different from prior homography estimation approaches which use only a single homography (*i.e.*, setting $\mathbf{H}_2 = \mathbf{I}_{3 \times 3}$), we introduce two homographies and use their composite to generate training data. This modification is crucial for ensuring that our method remains invariant to the input sequence.

Augment with dynamic occlusions. To enable GFNet to handle real-world scenarios with dynamic occlusions, we follow [35], which augments training data with dynamic objects. Specifically, we treat the images generated in Figure 3 as background. Objects are first added to I_S , then transformed with new planar transformations and added to I_T as the foreground, simulating moving occlusions in real-world scenarios. In this case, the ground truth confidence map \mathbf{m}_{gt} is also updated, where the area occupied by the object in I_S and its reprojection from I_T to I_S are set to 0. Moving objects are from MSCOCO [21] and augmented training examples are shown in Figure 3.

4. Experiment

4.1. Datasets

Basic training. Following [34], we use the CityScapes [9] and ADE-20K [46] datasets, both containing images larger than 750×750 , to generate 33,398 training samples of-fline, as described in the pipeline in Sec. 3.5. The model

trained on this split serves as the basic model, which is subsequently fine-tuned for application on multimodal datasets.

Fine-tuning. Following [5, 6, 43–45], we select GoogleMap and VIS-IR as the multimodal scenarios for evaluation. For GoogleMap, we create 5,000 satellite–map pairs from the Google Static Map API, each with a resolution of 1280×1280 , sourced from various countries. For VIS-IR, we select 5,000 visible–infrared pairs with a resolution of 640×512 , captured by UAVs, from the training split provided by [32]. The final training samples for GoogleMap and VIS-IR fine-tuning are generated on-the-fly from the selected aligned image pairs.

Inference. We test on MSCOCO, VIS-IR, GoogleMap and VIRAT datasets, examples are shown in Figure 3. For VIS-IR, we generate test samples from its test split, and for GoogleMap, test samples are generated from regions not overlapping with the training data. VIS-IR and GoogleMap are used to assess the model’s ability to handle photometric inconsistencies caused by modality changes, while MSCOCO [21] serves as a standard unimodal dataset, commonly used in previous works for evaluation. For each of these three datasets, we generate 1,000 test samples. **Note that** our MSCOCO and GoogleMap test sets have a different resolution setting (448×448) from those (128×128) used in prior works [5, 6, 49]. Besides, scale information in the GoogleMap test set is also different.

VIRAT is a dataset featuring dynamic scenes captured in a surveillance context [26], which we use to evaluate performance on dynamic scenes with occlusions. To create the dataset, we utilize the provided object annotations to identify dynamic objects in the videos. Image pairs are generated by cropping patches centered on the dynamic objects and aligning them with the same patches from different timestamps where the dynamic objects are absent. After removing similar samples, 144 test samples are finally collected from various video scenarios.

4.2. Experimental Settings

Implementation details. We use a batch size of 16 on each GPU, with a learning rate of $2 \cdot 10^{-4}$ for training the basic model, and $1 \cdot 10^{-4}$ for fine-tuning. We use the AdamW optimizer with a weight-decay factor of 10^{-2} , and

Table 1. Comparative results with 448×448 resolution. **Bold**: best, underline: second. Group A: homography estimation. Group B: image matching. Relative improvement or decrease is shown for GFNet.

Group	Method	MSCOCO				VIRAT				
		auc@3	auc@5	auc@10	auc@20	auc@3	auc@5	auc@10	auc@20	
A	RHWF	<u>93.02</u>	<u>95.81</u>	<u>97.9</u>	<u>98.77</u>	<u>46.57</u>	<u>53.7</u>	<u>61.53</u>	<u>70.64</u>	
	MCNet	91.23	94.73	97.36	98.68	42.41	50.76	59.92	69.83	
	PRISE	88.64	92.83	96.23	97.14	41.44	49.36	59.12	69.09	
	GFNet (ours)	97.69 (+5.0%)	98.61 (+2.9%)	99.3 (+1.4%)	99.65 (+0.8%)	46.92 (+0.7%)	54.47(+1.4%)	61.73(+0.3%)	70.78(+0.1%)	
B	SIFT+LightGlue	89.43	93.48	96.64	98.27	38.25	47.15	57.67	68.14	
	LoFTR	94.19	96.51	98.25	99.12	42.29	50.61	59.54	69.26	
	Efficient LoFTR	95.65	97.39	98.69	99.34	43.34	51.17	60.13	69.83	
	RoMa	<u>96.01</u>	<u>97.61</u>	<u>98.8</u>	<u>99.4</u>	47.5	<u>54.39</u>	61.91	70.88	
	GFNet (ours)	97.69 (+1.7%)	98.61 (+1.0%)	99.3 (+0.5%)	99.65 (+0.2%)	<u>46.92 (-1.2%)</u>	54.47(+0.1%)	<u>61.73(-0.2%)</u>	<u>70.78(-0.1%)</u>	
Group	Method	VIS-IR				GoogleMap				
		auc@3	auc@5	auc@10	auc@20	auc@3	auc@5	auc@10	auc@20	
A	RHWF	<u>18.16</u>	<u>32.05</u>	<u>54.48</u>	<u>72.84</u>	<u>18.32</u>	<u>37.47</u>	<u>61.09</u>	<u>76.43</u>	
	MCNet	16.1	30.59	51.51	71.82	16.31	35.28	59.51	75.63	
	PRISE	13.28	24.2	48.6	66.6	11.41	31.68	54.78	70.54	
	GFNet (ours)	21.28(+17.1%)	35.72(+11.4%)	58.44(+7.2%)	76.23(+4.6%)	51.44(+180.7%)	66.34(+77.0%)	80.62(+31.9%)	89.66(+17.3%)	
B	SIFT+LightGlue	2.56	8.91	23.33	39.99	-	-	-	-	
	LoFTR	10.17	22.51	45.93	65.84	4.99	13.87	32.27	51.24	
	Efficient LoFTR	13.28	26.32	49.43	69.1	5.37	14.49	33.6	52.73	
	RoMa	25.74	39.79	62.49	80.14	45.29	62.18	78.63	88.23	
	GFNet (ours)	<u>21.28(-17.3%)</u>	<u>35.72(-10.2%)</u>	<u>58.44(-6.4%)</u>	<u>76.23(-4.8%)</u>	51.44(+13.5%)	66.34(+6.6%)	80.62(+2.5%)	89.66(+1.6%)	
		Group A			Group B				GFNet (ours)	GFNet (ours)
		RHWF	MCNet	PRISE	SIFT+LightGlue	LoFTR	Efficient LoFTR	RoMa	w/o iteration	w/ iteration
Learnable parameters		1.29M	0.85M	19.24M	11.88M	11.56M	16.02M	111.28M	3.86M	3.86M
MACs		23.43G	4.56G	55.25G	38.51G	249.72G	179.81G	2503.19G	1657.62G	1709.09G
Runtime		79.70ms	28.11ms	204.67ms	390.11ms	42.68ms	31.29ms	212.97ms	118.90ms	124.62ms

use CosineAnnealing to schedule the learning rate. Both the basic training and fine-tuning stages use 2,000,000 training samples drawn from their respective datasets. Each epoch processes 25,000 samples, resulting in a total of 80 epochs. We train at a resolution of 448×448 , with the higher resolution in the recurrence set to 560×560 . We test MSCOCO and VIRAT with the basic model, while VIS-IR and GoogleMap with the fine-tuned models. Trained on 2 24GB RTX 3090 GPUs; Xeon Silver 4210R CPU.

For data generation, we set $d = 0.3$ following prior work, and use composite homography during training only. The model adopts DINOv2-large with 4 cross-attention layers, and each decoder uses 8 stacked depthwise conv blocks as in [13, 14]. The local radius from coarse to fine is [7, 6, 4, 2, 0]. We set $N = 2$ iterations (only for multimodal cases) to balance performance and efficiency. For loss, we use $\lambda = 0.85$, $\alpha = 0.01$. For estimation, $K = 5,000$ matches are selected, and RANSAC is applied using the default `cv2.findHomography` setting.

Metrics. We evaluate accuracy using the Average Corner Error (ACE, truncated at 70 pixels) and the AUC of ACE at thresholds of 3, 5, 10, and 20 pixels [5, 6, 29]. Efficiency is measured by runtime and Multiply-Accumulate Operations (MACs).

4.3. Comparative Results

Baselines. Comparative methods are divided into two groups: homography estimation methods (Group A) and image matching methods (Group B). In Group A,

we compare with RHWF [6], PRISE [44], and MCNet [49]. RHWF and MCNet are 4-point-based methods, whereas PRISE is a matrix-based method, initialized with MHN [15]. In Group B, we compare with the sparse matcher SIFT+LightGlue [22], the semi-dense matchers LoFTR [31] and Efficient LoFTR [37], and the dense matcher RoMa [14]. For a fair comparison, all methods in Group A are trained under the same settings as ours. For methods in Group B (excluding SIFT+LightGlue), we fine-tune their official outdoor models using our training data. During evaluation, we use RANSAC with consistent settings to compute the homography, limiting correspondences to 2,048 for sparse and semi-dense matchers.

Evaluation on MSCOCO. As shown in Table 1, GFNet outperforms all comparative methods on MSCOCO. Compared to methods in Group A, GFNet’s ability to train at higher resolutions, beyond the 128×128 limitation of Group A methods, leads to improved accuracy. Compared to methods in Group B, GFNet benefits from the accuracy of the dense matching framework, while the grid-based strategy is highly effective for homography estimation.

Evaluation on multimodal datasets. GFNet demonstrates significant improvements over SOTA homography estimation methods, with a 17.1% auc@3 increase on VIS-IR and a $1.8 \times$ auc@3 improvement on GoogleMap. Beyond the advantage of high-resolution training, this improvement is largely due to the integration of DINOv2, which excels in cross-domain feature matching, boosting performance in multimodal scenarios.

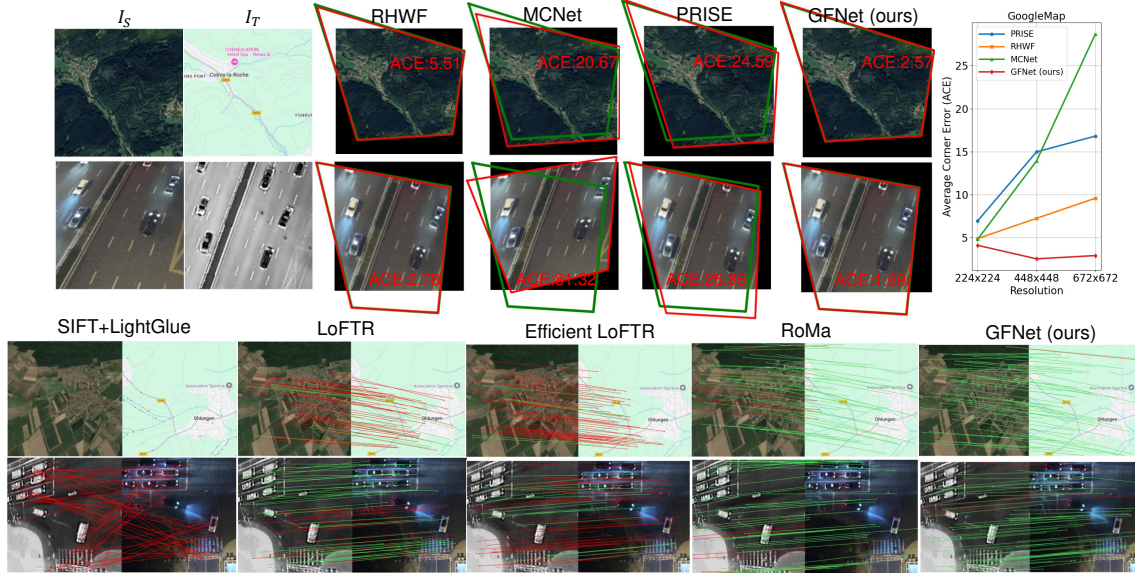


Figure 4. Visualization results on GoogleMap and VIS-IR. Top: homography estimation results. The green polygon is the ground-truth location of I_S on I_T , while the red one is the predicted location. The right plot shows ACE at different resolutions on GoogleMap. Bottom: image matching results. We randomly visualize 50 predicted matches. Matches with reprojection error lower than 3 pixels are drawn in green, otherwise red. Only keypoints will be displayed when no matches are predicted.

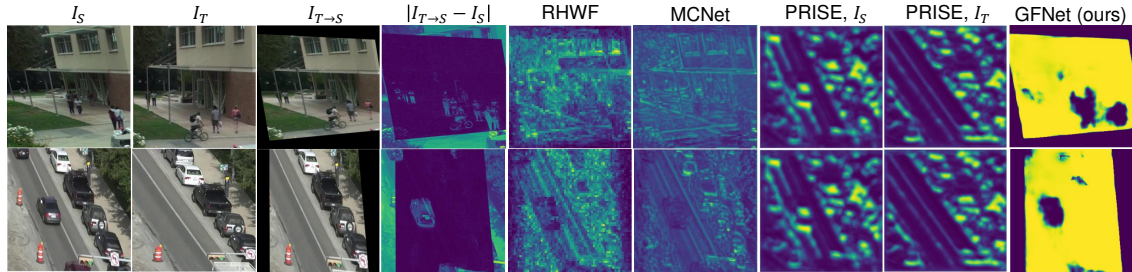


Figure 5. Visualization of inlier information in VIRAT. Darker indicates lower values, and $1 - |I_{T \rightarrow S} - I_S|$ is GT. While other methods recognize occlusion implicitly, GFNet handles it explicitly.

GFNet outperforms both sparse and semi-dense matchers on the two multimodal datasets. As shown in Figure 4, SIFT+LightGlue struggles with large modality gaps, while LoFTR-based methods yield low accuracy. Compared to RoMa, GFNet achieves a 13.5% gain in AUC@3 on GoogleMap but performs worse on VIS-IR, likely due to low-light image degradation where RoMa’s high-dimensional features offer better robustness.

Evaluation on datasets with dynamic occlusions. Generally, image matching methods are trained on the 3D datasets to handle two- or multi-view matching, which gives them an inherent ability to address dynamic occlusion. Thus here we mainly discuss how homography estimation methods, which only involve planar information, learn to manage dynamic occlusion. As shown in Figure 5, RHWF and MCNet aggregate matching information to estimate homography, with occluded areas being implicitly filtered during this process. PRISE aligns the feature maps of two images to

find the homography, also learning occlusion implicitly. In contrast, GFNet explicitly predicts a mask to identify occlusion, providing more precise occlusion information, which results in better performance on VIRAT compared to methods that implicitly predict occlusion (see Table 1).

Computational analysis. From the bottom table in Table 1, we observe that although GFNet is under the framework of dense matching, it reduces the MACs by 33.7% without iteration and by 31.7% with iteration compared to the SOTA dense matcher RoMa. We attribute it to our smaller network architecture and the grid-based strategy.

Robustness to resolution. In addition to the 448×448 GoogleMap test set presented in Table 1, we create two more test sets, each containing 1000 image pairs, with resolutions of 224×224 and 672×672 . The results are presented in Figure 4. Since homography estimation methods in Group A only accept 128×128 inputs, errors are calculated at this scale and rescaled to the original resolu-

Table 2. Grid size ablation. MSCOCO: w/o iteration. GoogleMap: w/ iteration.

Dataset	Model	DINOv2 grid size ($l = 1$)	FPN grid size (from $l = 2$ to $l = 5$)	Training costs on 1 GPU	Runtime (ms) ↓	MACE (pixel) ↓
MSCOCO	Half	1/14	1/28, 2/28, 4/28, 8/28	bs=12, 10.9GB, 16h5m	115.3	0.068
	Full		1/8, 1/4, 1/2, 1	bs=12, 22.3GB, 22h23m	134.2	0.083
	Ours		1/14, 2/14, 4/14, 8/14	bs=12, 14.0GB, 17h18m	118.9	0.063
GoogleMap	Half	1/14	1/28, 2/28, 4/28, 8/28	bs=7, 8.3GB, 24h2m	117.9	3.31
	Full		1/8, 1/4, 1/2, 1	bs=7, 23.6GB, 33h42m	146.3	4.02
	Ours		1/14, 2/14, 4/14, 8/14	bs=7, 12.0GB, 25h20m	124.6	2.57

Table 3. Ablation on the iterative strategy.

N	MSCOCO		GoogleMap	
	MACE (pixel) ↓	Runtime (ms) ↓	MACE (pixel) ↓	Runtime (ms) ↓
1	0.063	118.9	3.01	109.6
2	0.135	134.2	2.57	124.6
4	0.207	163.3	2.62	153.6

Table 4. Ablation on the network architecture and evaluation protocol. In w/o DINOv2, we use a trainable pre-trained ResNet50.

		w/o DINOv2	w/o Cross-Att.	w/o symmetric	w/o recurrence	Baseline
MSCOCO	MACE(pixel) ↓	0.056	0.071	0.069	0.048	0.063
	Runtime(ms) ↓	91.8	117.2	119.3	70.8	118.9
GoogleMap	MACE(pixel) ↓	8.35	2.97	2.95	3.19	2.57
	Runtime(ms) ↓	99.6	121.5	119.0	70.5	124.6

tion, leading to a near-linear increase in error as resolution grows. Moreover, the resizing process leads to loss of image details, which can further amplify the error beyond the expected near-linear growth. In contrast, GFNet does not impose input resolution constraints, making it more robust to resolution changes.

4.4. Ablation Studies

Ablation experiments are conducted on MSCOCO (natural images) and GoogleMap (multimodal images). We mainly ablate our grid-based strategy, network architecture, iterative strategy, evaluation strategy, and data generation.

Grid size. We fix the initial grid size based on DINOv2 and ablate the following 4 scales (Table 2). Grid-based flow reduces costs without sacrificing accuracy, validating that pixel-based flow is redundant for homography. However, too small a grid may slightly decrease accuracy. Our subsampling strategy—initial patch matching with DINOv2 followed by refinement and scaling—offers an intuitive balance between accuracy and efficiency.

Iteration. Results in Table 3 demonstrate that iteration is helpful for multimodal data but not for natural images. We attribute this to the challenge of precisely aligning the same physical positions in multimodal image pairs due to photometric inconsistencies, making only approximate alignment possible. The iterative strategy helps learn the process of optimization by mimicking this approximate alignment. However, for natural images where exact alignment is achievable, the iterative approach, which is designed for approximate alignment, can reduce accuracy. Setting $N = 2$ strikes a balance between accuracy and efficiency.

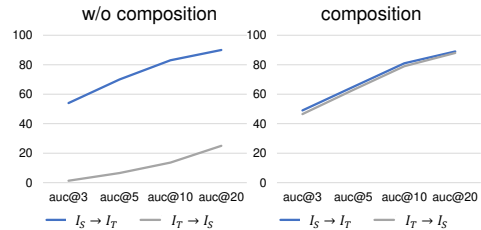


Figure 6. Impact of data generation during training.

Data generation. As illustrated in Figure 6, if we only apply the deformation to I_S during training (w/o dynamic occlusions), the network tends to overfit to the input sequence (I_S, I_T) and fails to work on the reversed sequence (I_T, I_S). However, by incorporating the composition homography during training, the network learns to generalize to both input sequences.

Network architecture. As shown in Table 4, replacing DINOv2 initialization with the 1/16 feature map from ResNet50 leads to a significant performance drop on GoogleMap. This highlights the effectiveness of DINOv2, which excels in semantic matching, as a reliable choice for grid flow initialization in multimodal scenarios. Regarding cross attention layers, it provides a stable accuracy improvement on MSCOCO and GoogleMap.

Evaluation protocol. As shown in Table 4, symmetry and adaptive recurrence have a limited impact on MSCOCO, as the matches predicted in the initial stage are already sufficiently accurate. However, they prove effective on the multimodal case, *i.e.*, GoogleMap.

5. Conclusions

This paper introduces GFNet, a method designed to deliver high-accuracy homography estimation across varying image resolutions, overcoming the resolution limitations of current approaches. This strong performance stems from the dense matching paradigm and several key components, including the integration of DINOv2, a grid-based strategy, iterative regression, and adaptive resolution recurrence. Experiments show that GFNet outperforms existing homography estimation methods on challenging datasets, including multimodal and dynamic scenes, and achieves comparable results to the SOTA dense matcher with higher efficiency.

Acknowledgments

This work was supported by NSFC (62276192).

References

- [1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56:221–255, 2004. 1
- [2] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019. 4
- [3] Axel Barroso-Laguna, Sowmya Munukutla, Victor Adrian Prisacariu, and Eric Brachmann. Matching 2d images in 3d: Metric relative pose from metric correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4852–4863, 2024. 3
- [4] Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer++: Revealing the devil in transformer’s details for multi-view stereo. In *Proceedings of the International Conference on Learning Representations*, 2024. 3
- [5] Si-Yuan Cao, Jianxin Hu, Zehua Sheng, and Hui-Liang Shen. Iterative deep homography estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1879–1888, 2022. 1, 2, 4, 5, 6
- [6] Si-Yuan Cao, Runmin Zhang, Lun Luo, Beinan Yu, Zehua Sheng, Junwei Li, and Hui-Liang Shen. Recurrent homography estimation using homography-guided image warping and focus transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9833–9842, 2023. 1, 2, 5, 6
- [7] Che-Han Chang, Chun-Nan Chou, and Edward Y Chang. Clkn: Cascaded lucas-kanade networks for image alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2213–2221, 2017. 1, 2, 5
- [8] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *Proceedings of the European Conference on Computer Vision*, pages 20–36, 2022. 3
- [9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 5
- [10] Yuxin Deng and Jiayi Ma. Redfeat: Recoupling detection and description for multimodal feature learning. *IEEE Transactions on Image Processing*, 32:591–602, 2022. 2
- [11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 1, 2, 5
- [12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 2
- [13] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023. 1, 2, 6
- [14] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024. 1, 2, 3, 4, 6
- [15] Farzan Erlik Nowruzi, Robert Laganieri, and Nathalie Japkowicz. Homography estimation from image pairs with hierarchical convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 913–920, 2017. 2, 5, 6
- [16] 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. 1, 4
- [17] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1
- [18] Xingyu Jiang, Jiayi Ma, Guobao Xiao, Zhenfeng Shao, and Xiaojie Guo. A review of multimodal image matching: Methods and applications. *Information Fusion*, 73:22–71, 2021. 2
- [19] Anush Kumar, Fahim Mannan, Omid Hosseini Jafari, Shile Li, and Felix Heide. Flow-guided online stereo rectification for wide baseline stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15375–15385, 2024. 1
- [20] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7652–7661, 2020. 1, 2, 5
- [21] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 5
- [22] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 2, 6
- [23] 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. 1, 2
- [24] Xinran Liu, Xiaoqiong Liu, Ziruo Yi, Xin Zhou, Thanh Le, Libo Zhang, Yan Huang, Qing Yang, and Heng Fan. Planartrack: A large-scale challenging benchmark for planar object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20449–20458, 2023. 1
- [25] 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*, 3(3):2346–2353, 2018. 2
- [26] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3153–3160, 2011. 5
- [27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafranec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2
- [28] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2682–2691, 2024. 2
- [29] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. 2, 6
- [30] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14890–14899, 2021. 1, 2, 3, 5
- [31] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021. 6
- [32] Yiming Sun, Bing Cao, Pengfei Zhu, and Qinghua Hu. Drone-based rgb-infrared cross-modality vehicle detection via uncertainty-aware learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):6700–6713, 2022. 5
- [33] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419, 2020. 4
- [34] Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6258–6268, 2020. 2, 3, 4, 5
- [35] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10247–10266, 2023. 1, 2, 5
- [36] Xiaolong Wang, Runsen Xu, Zhuofan Cui, Zeyu Wan, and Yu Zhang. Fine-grained cross-view geo-localization using a correlation-aware homography estimator. *Advances in Neural Information Processing Systems*, 36, 2024. 1
- [37] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21666–21675, 2024. 6
- [38] Han Xu, Jiteng Yuan, and Jiayi Ma. Murf: Mutually reinforcing multi-modal image registration and fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12148–12166, 2023. 1
- [39] Tongyao Yang and Fengbao Yang. A high-efficiency two-layer path planning method for uavs in vast airspace. *Chinese Journal of Information Fusion*, 1(2):109–125, 2024. 1
- [40] 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
- [41] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *Proceedings of the European Conference on Computer Vision*, pages 653–669, 2020. 2
- [42] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [43] Kaining Zhang and Jiayi Ma. Sparse-to-dense multimodal image registration via multi-task learning. In *Proceedings of the International Conference on Machine Learning*, pages 59490–59504, 2024. 1, 2, 5
- [44] Yiqing Zhang, Xinming Huang, and Ziming Zhang. Prise: Demystifying deep lucas-kanade with strongly star-convex constraints for multimodal image alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13187–13197, 2023. 2, 6
- [45] Yiming Zhao, Xinming Huang, and Ziming Zhang. Deep lucas-kanade homography for multimodal image alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15950–15959, 2021. 1, 2, 5
- [46] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 5
- [47] Tianhao Zhou, Haipeng Li, Ziyi Wang, Ao Luo, Chen-Lin Zhang, Jiajun Li, Bing Zeng, and Shuaicheng Liu. Recdiffusion: Rectangling for image stitching with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2692–2701, 2024. 1
- [48] B Zhu and YX Ye. Multimodal remote sensing image registration: A survey. *Journal of Image and Graphics*, 29(08):2137–2161, 2024. 2
- [49] Haokai Zhu, Si-Yuan Cao, Jianxin Hu, Sitong Zuo, Beinan Yu, Jiacheng Ying, Junwei Li, and Hui-Liang Shen. Mcnet: Rethinking the core ingredients for accurate and efficient homography estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25932–25941, 2024. 1, 2, 3, 4, 5, 6