Iterative Deep Homography Estimation

Si-Yuan Cao, Jianxin Hu, Zehua Sheng, Hui-Liang Shen
Zhejiang University
karlcao@hotmail.com, {hujianxin,shengzehua,shenhl}@zju.edu.cn

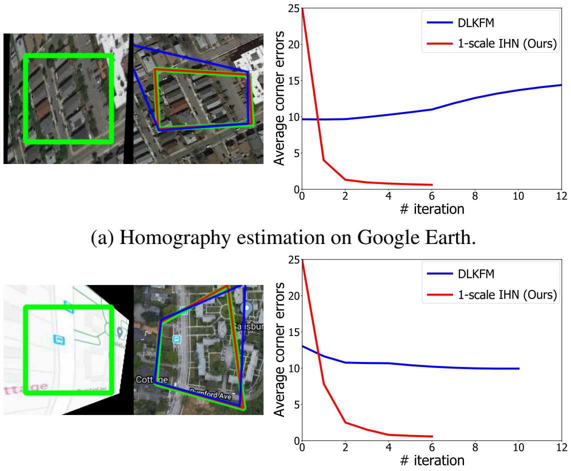
Abstract

We propose Iterative Homography Network, namely IHN, a new deep homography estimation architecture. Different from previous works that achieve iterative refinement by network cascading or untrainable IC-LK iterator, the iterator of IHN has tied weights and is completely trainable. IHN achieves state-of-the-art accuracy on several datasets including challenging scenes. We propose 2 versions of IHN: (1) IHN for static scenes, (2) IHN-mov for dynamic scenes with moving objects. Both versions can be arranged in 1-scale for efficiency or 2-scale for accuracy. We show that the basic 1-scale IHN already outperforms most of the existing methods. On a variety of datasets, the 2-scale IHN outperforms all competitors by a large gap. We introduce IHN-mov by producing an inlier mask to further improve the estimation accuracy of moving-objects scenes. We experimentally show that the iterative framework of IHN can achieve 95% error reduction while considerably saving network parameters. When processing sequential image pairs, IHN can achieve 32.7 fps, which is about 8× the speed of IC-LK iterator. Source code is available at https://github.com/imdump178/IHN.

1. Introduction

Homography estimation aims to find the global perspective transform between two images. It serves as a crucial step in a widely range of computer vision tasks such as image/video stitching [13, 32], video stabilization [16], SLAM [9, 24], augmented reality [29], GPS denied navigation [12, 40], and multimodal image fusion [37, 42].

The approaches in the literature can be roughly categorized into photometric-based and feature-based ones [32]. Photometric-based approaches aim to estimate homography from pixel intensities. The Lucas-Kanade (LK) algorithm [2, 19] is the most widely adopted photometric-based approach, which iteratively estimates the residual homography using a pre-computed iterator. Feature-based approaches usually consist of three steps: feature extraction, feature matching, and homography estimation [32]. The well-known feature extractors are SIFT [18], SURF [4], and ORB [4]. Homography estimation methods include RANSAC [11], DLT [8], and MAGSAC [3].



(a) Homography estimation on Google Earth.

(b) Homography estimation on Google Map & Satellite.

Figure 1. Visualization of homography estimation with average corner error (ACE) at each iteration of our IHN and DLKFM [40] (which use the traditional IC-LK iterator). Left 2 images: image pair for homography estimation with the source image I_{S} on the left and the target image I_{T} on the right. Green polygons denote the ground-truth position of I_{S} on I_{T}. Blue polygons denote the estimated position using MHN+DLKFM. Red polygons denote the estimated position using our IHN. Right plot: ACEs during first 12 iterations. IHN stops at iteration 6 while DLKFM has a dynamic stop criterion which iterates 21 times averagely.

Recently, deep homography methods have attracted increasing interest due to their superior performance. The VGG-style network, first proposed by DeTone *et al.* [6], is adopted to directly estimate homography. Based on this work, many recent methods have been introduced to boost the estimation accuracy by cascading multiple VGG-style networks [10, 14, 41]. The cascading is actually a way of iteration that can significantly improve the estimation performance. However, this way of iteration is limited to a fixed number of cascading, and more network cascading does not necessarily lead to better performance [14].

To further improve the homography estimation accuracy, some works [5, 40] take the Lucas-Kanade (LK) algorithm as an untrainable iterator and combine it with CNNs. However, the approximation of the Hessian matrix in the LK algorithm fails when the Jacobian matrix is rank-deficient [26]. Even worse, the training of the network is limited to the feature extractor, which means the above drawback is theoretically unavoidable.

To cope with these issues, we propose iterative homography network (IHN), which is completely trainable. We introduce 2 versions of IHN: (1) IHN for static scenes, (2) IHN-mov for dynamic scenes with moving objects. Both versions can be arranged in 1-scale for efficiency or 2-scale for accuracy. We show that the basic 1-scale IHN already outperforms most of the existing methods. On MSCOCO [15], the 2-scale IHN outperforms all competitors by a large gap, with more than 90% of the average corner errors (ACEs) lower than 0.1 pixels. On cross-modal datasets [40], IHN outperforms the state-of-the-art method MHN+DLKFM [40] that uses the traditional LK iterator in refining the deep homography estimation of MHN [14]. On moving-objects data [35], IHN and IHN-mov both surpass the competitors. IHN-mov gains further accuracy improvement by producing an inlier mask mimicking the essence of RANSAC [11].

The motivation of IHN comes from the traditional IC-LK iterator [2]. Different from IC-LK, IHN is completely trainable and thus can learn prior information for the residual homography prediction directly from data. Fig. 1 illustrates the homography estimation results and average corner errors (ACEs) at each iteration for our IHN and DLKFM [40] that use the IC-LK iterator. The initial ACEs at iteration 0 differs because DLKFM uses the preliminary estimation from MHN. It is observed that our basic 1-scale IHN can produce an accurate homography estimation within 6 iterations, while DLKFM using the traditional IC-LK fails. Furthermore, IHN can process image pairs in sequence at 32.7 fps, which is about 8× the speed of the IC-LK iterator. We further explore the effectiveness of our iterative framework by replacing our global motion aggregator (GMA, which directly estimate the residual homography) with the network architectures in [6, 14, 39]. Experimental results show that compared to feature/image concatenation strategy in [6, 14, 25, 39, 41], our iterative framework achieves about 95% error reduction. Thanks to the iterative framework, our global motion aggregator can achieve comparable accuracy with much fewer parameters than previous architectures, *e.g.*, 36.4% of the architecture in [14] and 3.6% in [39]. The same deep iterative concept that significantly improves optical flow estimation accuracy has been proposed in RAFT [33], which inspires us to construct a completely trainable deep homography estimation network.

To summarize, the main contributions of this work are as follows:

- We propose an iterative homography network, namely IHN, which is completely trainable. IHN achieves state-of-the-art accuracy on several datasets including challenging scenes. The iteration of IHN is stable and doesn't require extra parameters.
- We show that the proposed iterative framework is vital for an accurate homography estimation, which can achieve 95% error reduction regardless of a specific network architecture design. The iterative framework also enables considerable parameter saving.
- We specially design a network architecture IHN-mov for the moving-objects scenes where homography assumption is violated. The network produces an inlier mask mimicking the essence of RANSAC that can further benefit the homography estimation.

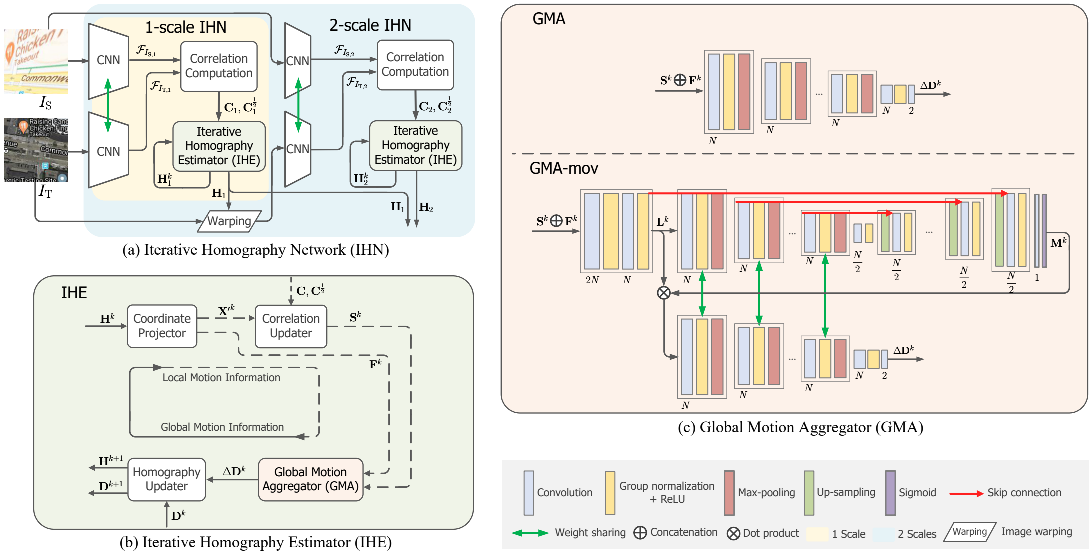
2. Related Work

We make a brief introduction of deep homography estimation, challenges in homography estimation, and iterative homography estimation that are most relevant to our method. For the basic knowledge of homography estimation, the readers are referred to [32, 43].

Deep Homography Estimation. Deep homography estimation is first proposed by DeTone *et al.* [6], who adopted a VGG-style network to directly predict the homography between the concatenated source and target images. Following this pioneering work, several works [10, 14, 41] proposed to cascade multiple VGG-style networks to improve the homography estimation accuracy. Nowruzi *et al.* [10] proposed to arrange similar stacked networks to successively refine the homography estimation. Le *et al.* [14] proposed to use multiscale VGG-style networks to iteratively estimate the residual homography. Nevertheless, compared to the Lucas-Kanade iterator [2, 19], the cascaded deep homography methods still lack accuracy [5, 40].

Another kind of works accomplishes iterative deep homography estimation by combining the LK algorithm with CNNs. Chang *et al.* [5] adopted the inverse compositional LK (IC-LK) iterator as an untrainable layer of the deep network. A CNN is employed to extract the feature maps that are optimal for the IC-LK iterator. Zhao *et al.* [40] proposed to construct a one-channel deep Lucas-Kanade feature map (DLKFM) using CNNs. The DLKFM is then sent into the IC-LK iterator. Similar approaches can be found in [12, 34]. However, the LK iterator is untrainable, and therefore theoretically the drawbacks such as rank-deficient Jacobian cannot be avoided.

Challenges in Homography Estimation. According to the recent works [14, 25, 28, 39, 40], there are mainly two challenges in homography estimation. The first challenge comes from the photometric inconsistency such as illumination change or modality variation. Nguyen *et al.* [25]



(a) Iterative Homography Network (IHN)

(b) Iterative Homography Estimator (IHE)

(c) Global Motion Aggregator (GMA)

Figure 2. Schematics and detailed architectures of iterative homography network (IHN). (a) Overall schematic of IHN, including the illustration of basic 1-scale IHN and 2-scale IHN. (b) Structure of the iterative homography estimator (IHE), which plays a leading role of realizing the iterative homography refinement. (c) Architecture of the global motion aggregator (GMA) that directly estimates residual homography with GMA designed for the static scenes and GMA-mov for the moving-objects scenes. Please refer to the text for details.

proposed an unsupervised learning method to improve the network capacity on illumination variation. Zhao *et al.* [40] proposed to extract intensity consistent DLKFM for cross-modal inputs. They estimated the homography using multiple cascaded VGG-style networks MHN [14] and refined it using the IC-LK iterator. The second challenge comes from the violation of the homography assumption. For example, in scenes with moving objects, the matching between the source and target images does not always satisfy a uniform homography. Zhang *et al.* [39] proposed a mask predictor network that weights the feature maps to achieve content-aware homography estimation. However, as the masks are separately computed on the source and target images, the method cannot reject matching outliers. Le *et al.* [14] proposed to teach the network a moving object mask produced by the PWC-Net [31] optical flow. However, the mask cannot be learned if the optical flow estimation fails.

Iterative Homography Estimation. The most widely used iterative homography estimation framework is the Lucas-Kanade (LK) algorithm [2, 19]. The objective function of the most widely adopted inverse compositional Lucas-Kanade (IC-LK) algorithm is

$$\min_{\Delta\mathbf{H}} \|I_{\mathrm{T}}(W(\mathbf{X};\Delta\mathbf{H}) - I_{\mathrm{S}}(W(\mathbf{X};\mathbf{H}))\|_2^2, \quad (1)$$

where $\Delta\mathbf{H}$ denotes the residual homography matrix, and W denote the coordinate warping operation. By conducting a first-order Taylor expansion on $I_{\mathrm{T}}(W(\mathbf{X};\Delta\mathbf{H})$, a closed-form solution of $\Delta\mathbf{H}$ can be computed as

$$\Delta\mathbf{H} = (\mathbf{J}^{\mathsf{T}}\mathbf{J})^{-1}\mathbf{J}^{\mathsf{T}}\mathbf{r}, \quad (2)$$

where $\mathbf{J}$ denotes the Jacobian matrix of I_{T}, $\mathbf{J}^{\mathsf{T}}\mathbf{J}$ is the approximation of the Hessian matrix. $\mathbf{r} = \mathrm{vec}(I_{\mathrm{T}} - I_{\mathrm{S}}(W(\mathbf{X};\mathbf{H})))$ represents the vectorized residual image that is updated in every iteration.

The IC-LK algorithm iteratively estimates homography in 3 steps: (1) updating $I_{\mathrm{S}}(W(\mathbf{X};\mathbf{H}))$ using local coordinates, and thus updating local information in $\mathbf{r}$; (2) aggregating residual global homography using Eq. (2); (3) projecting updated global homography into local coordinates for updating local information in the next iteration.

To the best of our knowledge, there isn't any completely iterative trainable deep homography estimation network before our IHN. The most related work is the deep iterative optical flow estimation proposed in RAFT [33].

3. Method

The overview of our iterative homography network (IHN) is illustrated in Fig. 2a. IHN takes in a pair of source image I_{S} and target image I_{T} and outputs the estimated homography matrix $\mathbf{H}_1$ (1 scale) or $\mathbf{H}_1$ and $\mathbf{H}_2$ (2 scales). The main steps of IHN include feature extraction using CNN, correlation computation, and recurrent homography estimation using an iterative homography estimator (IHE).

3.1. Feature Extraction

We extract the feature maps of the source and target images using a Siamese CNN. We take the combination of 1 max-pooling layer (stride 2) and 2 residual blocks as a basic unit. The images are first processed by 1 convolutional block with kernel size 7×7. We then add q basic units to produce the $1/2^q \times 1/2^q$ resolution feature maps. The feature maps are finally reprojected by 1 linear convolutional layer with kernel 1×1. Specifically, we set $q = 2$ in practical implementation. As illustrated in Fig. 2a, the 1-scale IHN uses the feature maps at $1/4 \times 1/4$ resolution, and the 2-scale IHN uses both feature maps at $1/4 \times 1/4$ and $1/2 \times 1/2$ resolution. For the 2-scale IHN, the first basic units of both resolution feature maps share the same weights. We found that the 1-scale IHN already achieves considerable homography estimation performance, while the 2-scale IHN can further improve the accuracy. The Siamese CNN is also used for cross-modal data. We will show in Section 4.4 that our IHN can produce promising homography estimation without a specific feature extractor setting (*e.g.*, the pseudo-Siamese network in [40] for cross-modal inputs).

3.2. Correlation

Different from most previous deep homography estimation works [6, 10, 14, 25, 41], we explicitly compute the correlation to enable iterative refinement. Let us denote the feature maps of the source and target images as $\mathcal{F}_{I_\mathrm{S}} \in \mathbb{R}^{D\times H\times W}$ and $\mathcal{F}_{I_\mathrm{T}} \in \mathbb{R}^{D\times H\times W}$. We set $D = 256$ for all feature maps. We compute the pairwise correlation, namely the correlation volume as

$$\mathbf{C}(\mathbf{x}_\mathrm{S}, \mathbf{x}_\mathrm{T}) = \mathrm{ReLU}(\mathcal{F}_{I_\mathrm{S}}(\mathbf{x}_\mathrm{S})^\intercal \mathcal{F}_{I_\mathrm{T}}(\mathbf{x}_\mathrm{T})), \tag{3}$$

where $\mathbf{x}_\mathrm{S}$ and $\mathbf{x}_\mathrm{T}$ denote the coordinate position of the source and target feature maps. The correlation volume $\mathbf{C}$ is of size $H\times W\times H\times W$. In one iteration, a fixed search window is sampled from $\mathbf{C}$ by the correlation updater, which is described in detail in Section 3.3. As mentioned in [33], the correlation volume can also be computed on demand during the iteration, which can reduce the space complexity.

Correlation Pooling. To enlarge the perception range within a feature scale, we conduct average pooling on $\mathbf{C}$ at stride 2 at the last 2 dimensions to form another correlation volume $\mathbf{C}^{\frac{1}{2}}$, which is of size $H\times W\times H/2\times W/2$. For both volumes, we use search windows of the same size, which means that the sampling operation on $\mathbf{C}^{\frac{1}{2}}$ has a 2×2 larger perception range compared to $\mathbf{C}$.

3.3. Iterative Homography Estimator

We design our iterative homography estimator (IHE) under the inspiration of the IC-LK iterator. IHE plays a leading role in the realization of the iterative homography refinement. As illustrated in Fig. 2a and 2b, IHE takes in the correlation volume $\mathbf{C}, \mathbf{C}^{\frac{1}{2}}$ and outputs the estimated homography $\mathbf{H}$. From the coordinate projector to the global motion aggregator, local motion information is aggregated into global homography estimation. From the global motion aggregator back to the coordinate projector (in the next iteration), global homography estimation is converted to local coordinates for local information update. Similar essence can be found in the IC-LK iterator.

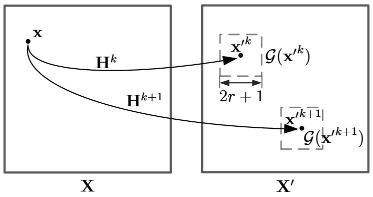


Figure 3. Illustration of the iterative process of coordinate projector and correlation updater. Left: coordinate of $\mathcal{F}_{I_\mathrm{S}}$, denoted as $\mathbf{X}$. Right: coordinate of $\mathcal{F}_{I_\mathrm{T}}$, denoted as $\mathbf{X'}$.

Coordinate Projector. Taking iteration k for example, the point-wise correspondence between the source feature map $\mathcal{F}_{I_\mathrm{S}}$ and the target feature map $\mathcal{F}_{I_\mathrm{T}}$ is mapped by the present homography matrix $\mathbf{H}^k$. Let's denote $\mathbf{X}$ as the meshgrid coordinate set of $\mathcal{F}_{I_\mathrm{S}}$, and its corresponding meshgrid coordinate set in $\mathcal{F}_{I_\mathrm{T}}$ as $\mathbf{X'}$. For each coordinate position, we denote $\mathbf{x} = (u, v), \mathbf{x} \in \mathbf{X}$ and $\mathbf{x} = (u', v'), \mathbf{x} \in \mathbf{X'}$. The point-wise correspondence of $\mathbf{X}$ and $\mathbf{X'}^k$ is projected by $\mathbf{H}^k$ using

$$\begin{bmatrix} u'^k \\ v'^k \\ 1 \end{bmatrix} \sim \begin{bmatrix} \mathbf{H}^k_{11} & \mathbf{H}^k_{12} & \mathbf{H}^k_{13} \\ \mathbf{H}^k_{21} & \mathbf{H}^k_{22} & \mathbf{H}^k_{23} \\ \mathbf{H}^k_{31} & \mathbf{H}^k_{32} & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{4}$$

We illustrate the coordinate projection in iteration k and $k+1$ in Fig. 3. To further facilitate the learning of local motion information, we also compute the homography flow $\mathbf{F}^k$ as

$$\mathbf{F}^k = \mathbf{X'}^k - \mathbf{X}. \tag{5}$$

$\mathbf{F}^k$ is then sent to the global motion aggregator.

Correlation Updater. The correlation updater samples the correlation volume $\mathbf{C}$ using homography projected coordinate $\mathbf{X'}^k$ and outputs updated correlation slice $\mathbf{S}^k$. The sampling process can be expressed as

$$\mathbf{S}^k(\mathbf{x}) = \mathbf{C}(\mathbf{x}, \mathcal{G}_r(\mathbf{x'}^k)), \tag{6}$$

where $\mathcal{G}_r(\mathbf{x'}^k)$ denotes a local square grid of fixed search radius r. The square grid sampling is illustrated in Fig. 3. Note that the correlation slice is also sampled on the pooled correlation volume $\mathbf{C}^{\frac{1}{2}}$ to make $\mathbf{S}^{\frac{1}{2},k}$ that have 2×2 larger perception range.

Global Motion Aggregator. The residual homography is estimated by the global motion aggregator, with the homography matrix parameterized by the displacement vectors of the 4 corner points as in [6, 14, 28].

As illustrated in Fig. 2c, we design the GMA for static scenes and the GMA-mov for moving-objects scenes. N denotes the number of filters of convolutional kernels.

GMA is mainly composed of multiple basic units. Each basic unit includes a 3×3 convolutional block, 1 group normalization [36] + ReLU [22], and 1 max-pooling layer (stride 2). We continually add the basic unit until the spatial resolution of the feature map reaches 2×2. Then a convolutional block projects the feature map into a $2 \times 2 \times 2$ cube

$\Delta\mathbf{D}^k$, which is the estimated residual displacement vectors of the 4 corner points. In iteration k, GMA takes the concatenated correlation slice $\mathbf{S}^k$ and homography flow $\mathbf{F}^k$ as input.

GMA-mov is specifically designed for scenes with moving objects. GMA-mov explicitly produces a mask $\mathbf{M}^k$ to weight the matching inliers that meet the homography transform assumption. It is worth noting that, different from [14] that needs additional optical flow supervision or [39] that relies on image content, our GMA-mov can produce an inlier mask based on the combined local and global motion information as RANSAC. The mask can be generated without supervision, yet can improve the homography estimation accuracy. As illustrated in Fig. 2c, GMA-mov explicitly encodes the point-wise local motion information into feature map $\mathbf{L}^k$ using 1 convolutional block. $\mathbf{L}^k$ is then sent into multiple basic units as in GMA to preliminarily extract the $N \times 2 \times 2$ feature map containing the global motion information. Different from GMA, the extracted global motion information is not directly used for the residual homography estimation, but the inlier mask prediction. The latter half part of GMA-mov progressively upsamples the feature map containing the global motion information and combines it with the local motion information by skip connections. The inlier mask $\mathbf{M}^k$ with the same size as $\mathbf{L}^k$ is predicted by a sigmoid function. $\mathbf{M}^k$ and $\mathbf{L}^k$ are taken dot product and sent into the same structure as GMA to produce the residual homography estimation. We note that the basic units used for extracting the preliminary global motion information and residual homography estimation share the same weights.

Homography Updater. As in [6, 14, 28], we parameterize the homography matrix using the displacement vectors of the 4 corner points of an image, namely the displacement cube $\mathbf{D}$. In iteration k, $\mathbf{D}$ is updated as

$$\mathbf{D}^{k+1} = \mathbf{D}^k + \Delta\mathbf{D}^k. \tag{7}$$

With $\mathbf{D}^{k+1}$, it is convenient to get the homography matrix $\mathbf{H}^{k+1}$ by the least square method, the direct linear transform [1] or other methods. The updated $\mathbf{H}^{k+1}$ will be sent into the coordinate projector in the next iteration. The initial displacement cube is set to $\mathbf{D}^0 = \mathbf{0}$, which means identical transform $\mathbf{H}$.

3.4. Multiscale Strategy

We introduce an optional multiscale strategy that can further improve the homography estimation accuracy. We note that, according to our experiments, the 1-scale IHN can outperform most existing homography estimation methods. As illustrated in Fig. 2a, another scale of IHN with its correlation volume computed on the $1/2 \times 1/2$ resolution feature maps is attached to the 1-scale IHN. The target image I_T is warped using the estimated homography $\mathbf{H}_1$ at $1/4 \times 1/4$ resolution. The bottom right subscript, namely 1 or 2, denotes the computed result at the $1/4 \times 1/4$ resolution or the $1/2 \times 1/2$ resolution. The homography matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ from the 2 scales are composed to produce the final homography estimation for the 2-scale IHN as in [14, 28]. In Section 4, we show that IHN only needs 2 scales to achieve very high accuracy than the 3 or 4 scales IC-LK based methods [5, 40].

3.5. Loss Function

We apply supervision on the L_1 distance between the ground-truth displacement $\mathbf{D}_\mathrm{gt}$ and the estimated displacement $\mathbf{D}$ at each iteration. A weighted sum of all the iteration is computed as the loss function

$$L = \sum_{k=0}^{K-1} \alpha^{(K-k-1)} |\mathbf{D}^{k+1} - \mathbf{D}_\mathrm{gt}|, \tag{8}$$

where K denotes the total iteration at one resolution, and k ranges from 0 to $K-1$. If the multiscale strategy is applied, the loss of both resolutions is separately computed and summed to make the final loss.

4. Experiments

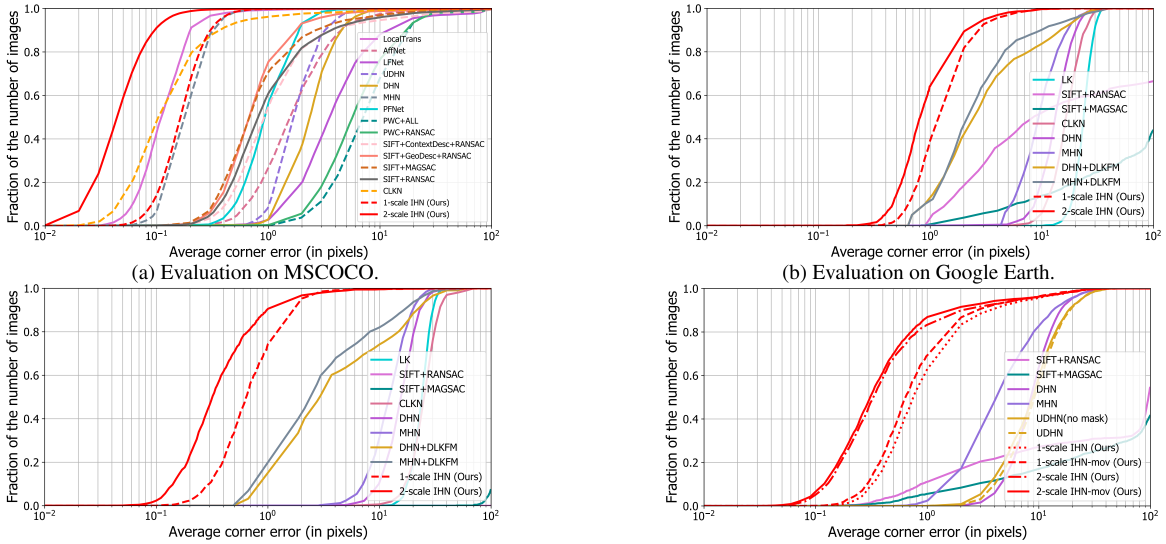
4.1. Implementation Details

We implement our network using PyTorch. The network is trained using AdamW [17] optimizer and the maximum learning rate is set to 2.5×10^{-4}. The network is trained with a batch size of 16 and a training iteration of 120000. The total iteration K at one resolution for training is set to 6. We set the search radius of correlation updater $r = 4$ and set $\alpha = 0.85$ in the loss function. We set $N = 128$ for the GMA in the basic 1-scale IHN, and set $N = 80$ for the GMA in the extra scale of the 2-scale IHN. For the mentioned different network structures and scales, we employ the same hyper-parameters.

4.2. Datasets

We evaluate IHN on both common and challenging datasets. We test IHN for static scenes on common MSCOCO dataset [15] as in [5, 6, 14, 28, 40]. We also evaluate IHN on the cross-modal Google Earth and Google Map & Satellite datasets [40]. For dynamic scenes, we test IHN and IHN-mov on a challenging moving-objects scene generated based on the SPID surveillance dataset [35].

Part of the images in SPID are discarded due to camera shake or overly low-quality. The images are processed in two different ways: (1) For scenes with low resolution, the original images are used directly. We randomly select two images within the same scene to make an image pair. (2) For scenes with high resolution, we randomly select one image first, and then use the provided pedestrian annotation



(a) Evaluation on MSCOCO.

(b) Evaluation on Google Earth.

(c) Evaluation on Google Map & Satellite.

(d) Evaluation on SPID.

Figure 4. Evaluation of homography estimation methods on MSCOCO, Google Earth, Google Map & Satellite, and SPID datasets. MSCOCO contains common RGB images. Google Earth and Google Map & Satellite are cross-modal datasets. SPID dataset provides surveillance images with foreground moving objects.

Table 1. Comparison of concatenation and iterative framework on MACE when using various network architectures.

	DHN [6]	MHN [14]	UDHN [39]	GMA
Concatenation	3.54	4.01	3.47	3.54
Iterative framework	0.19	0.20	0.20	0.19
Parameters	2.8 M	2.2 M	22.3 M	0.8 M

to determine the position of foreground moving objects. We expand the pedestrian annotation to make the crop area contain both object and background. We then randomly select another image within the same scene, and crop at the same position to obtain an image pair that contains different foreground moving objects but the same background. We use 80% images as the training data and 20% the test data.

For a fair comparison, all methods included for evaluation are trained and tested by the exact same corresponding training and test splits within each dataset.

4.3. Evaluation and Ablation Study on MSCOCO

We evaluate our IHN on the MS-COCO dataset [15] with LocalTrans [28], AffNet [23], CLKN [5], LFNet [27], DHN [6], UDHN [39], MHN [14], PFNet [38], PWC [31], SIFT+ContextDesc+RANSAC [20], SIFT+GeoDesc+RANSAC [21], SIFT+MAGSAC [3], and SIFT+RANSAC [18]. The experiment settings are the same as in most deep homography estimation methods [5, 6, 14, 40]. The corners of 128 × 128 images are randomly shifted in the range of $[-32, 32]$ pixels to make the deformed image. Similar to most deep homography works [5, 6, 14, 28, 40], we employ the average corner error (ACE) as the evaluation metric. An ablation study of our proposed iterative framework together with our network architecture is then conducted. All ablation study are conducted on the basic 1-scale IHN if not otherwise specifically mentioned.

Table 2. Ablation study on the settings of IHN.

Experiment	Setting	MACE	Parameters
Correlation pooling	No pooling	0.23	1.2 M
	Pooling	0.19	1.3 M
Parameterization	Homography	∞	1.3 M
	Displacement	0.19	1.3 M
Homography flow	No flow	0.21	1.3 M
	Flow	0.19	1.3 M
Scale	**1 scale**	0.19	1.3 M
	2 scales	0.06	1.7 M
Iteration	1	3.15	1.3 M
	6	0.19	1.3 M
	12	0.19	1.3 M
	100	0.19	1.3 M

Evaluation on MSCOCO. The statistical results on MSCOCO are illustrated in Fig. 4a. It is observed that the basic 1-scale IHN already outperforms most competitors except CLKN and LocalTrans. We note that MHN is conducted on 3 scales, which is surpassed by our basic 1-scale IHN. Our 2-scale IHN outperforms all other homography estimation methods by a large gap. The 2-scale IHN produces over 90% ACEs lower than 0.1 pixels, which significantly outperforms CLKN that adopt the traditional IC-LK iterator.

Ablation Study on the Iterative Framework. To reveal the effectiveness of the iterative framework in IHE, we specifically conduct a homography estimation performance comparison on our iterative framework and the feature/image concatenation strategies in [6, 14, 25, 39, 41]. We switch the global motion aggregator among the deep homography estimation architectures introduced in previous works including DHN [6], MHN [14], UDHN [39]. To avoid the influence of the feature extractor, we uniformly use the feature extractor introduced in Section 3.1. The extracted feature maps are concatenated in the channel dimension to achieve the feature/image concatenation strategy in [6, 14, 39]. Table 1 lists the mean average corner error (MACE) and parameters of network using different architectures. It is observed that compared to the feature/image concatenation strategy, our iterative framework in IHE significantly boosts the estimation accuracy by about 95% error reduction with the same network design. Also benefit from our iterative framework, our GMA can achieve comparable high accuracy with much fewer parameters compared to previous homography estimation architectures, *e.g.*, 36.4% of MHN [14] and 3.6% of UDHN [39].

Ablation Study on Settings of IHN. Table 2 lists the mean average corner error (MACE) and parameters of IHN under different settings. The parameter counting involves the feature extractor introduced in Section 3.1, which differs from Table 1. The symbol ∞ denotes the case that the training doesn't converge and the MACE falls into infinity. It is observed that the correlation pooling, the parameterization using displacement, and the addition of homography flow improve the accuracy with very few parameter costs. The employment of another scale can improve the accuracy, despite that the 1-scale version already achieves relatively high accuracy. We further test the influence of inference iteration under the training iteration of 6. It is observed that the network owns a significantly better performance at 6 and 12 iterations than 1 iteration, once again indicating that iteration is vital for a high precision. Another interesting phenomenon is that when we raise the iteration to 100, IHN is still stable and doesn't get into divergence. The bold setting options in Table 2 are employed in the rest experiments.

4.4. Evaluation on Cross-Modal Datasets

We evaluate our IHN on cross-modal datasets [40], including Google Earth of season change and Google Map & Satellite images with large modality difference. It is worth noting that we don't specifically alter IHN for cross-modal data as in [40], in which two separate feature extractors (namely pseudo-Siamese) are adopted. The original LK [2], SIFT+RANSAC [18], SIFT+MAGSAC [3], CLKN [5], DHN [6], MHN [14], DHN+DLKFM [40], and MHN+DLKFM [40] are included for comparison. As illustrated in Fig. 4b and 4c, our 1-scale and 2-scale IHN outperform the competitors by a large gap. We note that the recent MHN+DLKFM uses the combination of 3-scale VGG-style networks and 3-scales LK iterator. Our superiority over MHN+DLKFM further reveals the potential of the deep iterative framework.

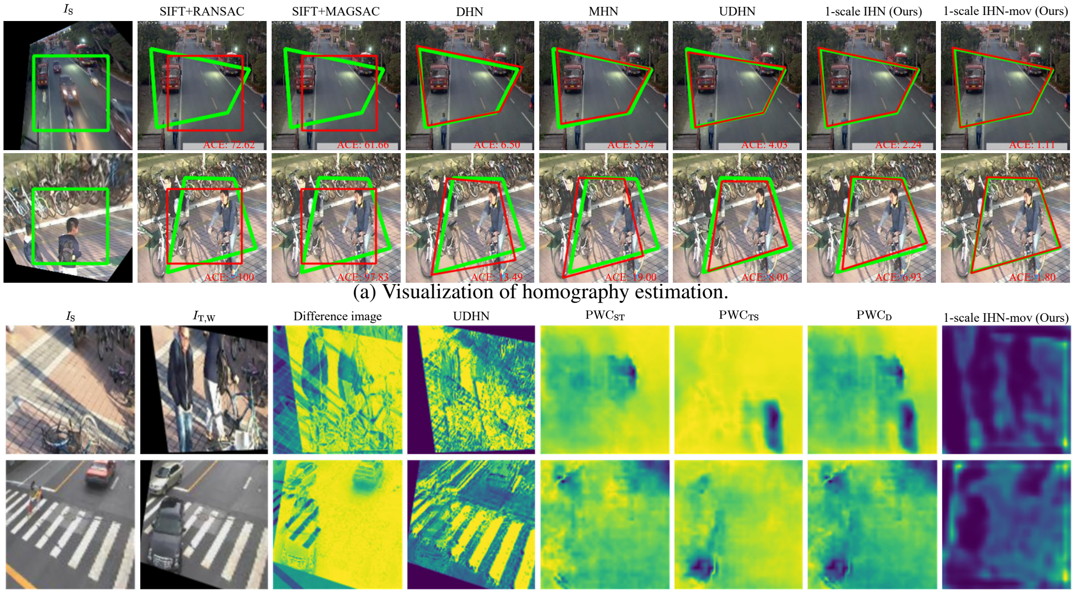
4.5. Evaluation on Moving-Objects Dataset

We further conduct an evaluation on the SPID dataset that contains foreground moving objects. We compare our IHN with SIFT+RANSAC [18], SIFT+MAGSAC [3], DHN [6], MHN [14], and UDHN [39]. The foreground moving objects usually occlude the background that satisfies the homography assumption, which makes accurate homography estimation difficult. We specifically propose an architecture for this scenario, called IHN-mov, which can produce a mask that explicitly weights the matching inliers to improve the estimation accuracy. We make UDHN into supervised for a fair comparison as its unsupervised training on SPID doesn't converge. As illustrated in Fig. 4d, 2 versions of IHN with different scales all outperform other competitors. 1-scale IHN-mov surpasses 1-scale IHN, with the fraction of ACE less than 1 pixel increased by around 6%. We further visualize the homography estimation of the above methods in Fig. 5a. It is observed that SIFT+RANSAC and SIFT+MAGSAC fail. The deep homography methods DHN, MHN, UDHN, and basic 1-scale IHN are affected by foreground moving objects, while 1-scale IHN-mov produces relatively more accurate homography estimations. The 2-scale IHN ourperforms 1-scale IHN-mov, meaning that the addition of extra scale can boost the estimation accuracy more. On the other side, the 2-scale IHN-mov exceeds 2-scale IHN, indicating that the weighting mask can boost the estimation within the same scale.

We further visualize the inlier mask produced by UDHN [39], PWC-Net [31] optical flow, and our 1-scale IHN-mov in Fig. 5b. As the ground-truth inlier mask is unavailable, we compute the difference of the source image I_{S} and the warped target image $I_{\mathrm{T,W}}$ and reverse its intensity to roughly illustrate the matching inliers. The mask of UDHN is obtained by dot product the separate UDHN masks of I_{S} and $I_{\mathrm{T,W}}$. We separately demonstrate the optical flow masks taking I_{S} as reference $\mathrm{PWC_{ST}}$, taking I_{T} as reference $\mathrm{PWC_{TS}}$ and their dot product $\mathrm{PWC_{D}}$. We reverse the normalized magnitude of the optical flow masks to show the weight of inliers. It is observed that the masks produced by UDHN are affected by image content. The shadow area in the left scene background is assigned with a very low weight, but it should belong to the matching inliers. As for the optical flow masks, the estimation fails as the moving objects vanish. On the contrary, our 1-scale IHN-mov produces the inlier masks directly from the local and global motion information like RANSAC, and thus is more reasonable. It's worth noting that we trained a version of UDHN without the weight mask, namely UDHN(no mask), the accuracy raises a little as shown in Fig. 4d.

4.6. Cross Dataset Evaluation

We conduct cross dataset evaluation of 1-scale IHN and DLKFM [40] that uses untrainable IC-LK, with results



(a) Visualization of homography estimation.

(b) Visualization of inlier mask.

Figure 5. Visualization of results on the SPID dataset with moving objects. In (a), green polygons denote the ground-truth position of I_S on the target images. Red polygons denote the estimated position using different algorithms on the target images. The closer the 2 colors of polygons are, the better estimation accuracy (also indicated by a lower ACE). In (b), the target images are warped to better illustrate the inlier masks. The mask produced by UDHN [39] detects the image content but not the needed matching inliers. The 3 masks produced by PWC-Net [31] fail to predict the correct inliers. Our IHN-mov produces a more reasonable inlier mask by excluding the moving objects.

Table 3. Cross dataset evaluation of 1-scale IHN and DLKFM.

Test / Train	MSCOCO		Google Earth		Google Map	
	IHN	DLKFM	IHN	DLKFM	IHN	DLKFM
MSCOCO	**0.19**	0.55	**2.41**	4.68	**24.21**	Fail
Google Earth	**2.42**	10.16	**1.60**	3.88	**13.30**	39.85
Google Map	**1.72**	37.30	**3.06**	39.72	**0.92**	4.41

listed in Table 3. It's shown that the generalization ability of IHN is superior than the IC-LK based DLKFM. For IHN, the model trained on hard datasets (*e.g*. Google map) generalizes better.

We evaluate our 1-scale IHN (trained on SPID) on a SfM dataset EPFL [30]. We include MHN [14] and DPCP [7] (a method designed for estimating camera pose in the SfM pipeline). The mean errors of rotation and translation of camera pose are reported in Table 4. It is observed that IHN owns the best accuracy for a good generalization ability.

4.7. Inference Time

Table 5 lists the inference time of 1-scale IHN, 2-scale IHN, 2-scale IHN-mov, conventional IC-LK iterator in DLKFM [40], and MHN+DLKFM. The comparison is conducted on an Intel Xeon Silver 4210R CPU @ 2.40GHz with 64GB memory and an NVIDIA Quadro RTX 8000.

Table 4. Cross dataset evaluation of 1-scale IHN, MHN, and DPCP on the EPFL SfM dataset.

	1-scale IHN	MHN	DPCP
Rotation	**4.38**	19.54	5.44
Translation	**13.54**	33.38	17.28

Table 5. Inference time comparison (in milliseconds) of different types of IHN, DLKFM and MHN+DLKFM. ”s.” denotes scale.

1-s. IHN	2-s. IHN	2-s. IHN-mov	DLKFM	MHN+DLKFM
30.6	60.4	93.6	253.7	380.9

It is observed that the 1-scale IHN takes significantly less time. The speed of 1-scale IHN is 8× that of IC-LK iterator. Furthermore, 1-scale IHN is able to process image pairs in sequence at 32.7 fps.

5. Conclusions

We have proposed a new iterative deep homography estimation architecture named IHN. Different from the iterative framework in the traditional IC-LK iterator, IHN is completely trainable. IHN achieves state-of-the-art performance on several challenging datasets, including cross-modal and moving-objects scenes. We experimentally show that the iterative framework of IHN is vital for the error reduction and parameter saving. IHN also has an advantage in speed over the traditional IC-LK algorithm.

Limitations. The limitations of IHN are mainly 2 folds. First, the correlation volume raises the demand for GPU memory. Second, the resolution of the inlier mask produced by IHN-mov is limited by the feature map size and thus cannot be very clear.

References

[1] Youssef I Abdel-Aziz, HM Karara, and Michael Hauck. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogrammetric Engineering & Remote Sensing*, 81(2):103–107, 2015. 5

[2] Simon Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004. 1, 2, 3, 7

[3] Daniel Barath, Jiri Matas, and Jana Noskova. MAGSAC: marginalizing sample consensus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10197–10205, 2019. 1, 6, 7

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006. 1

[5] 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, 6, 7

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 1, 2, 4, 5, 6, 7

[7] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6080–6089, 2020. 8

[8] Elan Dubrofsky. Homography estimation. *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, 5, 2009. 1

[9] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014. 1

[10] Farzan Erlik Nowruzi, Robert Laganiere, 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. 1, 2, 4

[11] 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, 2

[12] Hunter Goforth and Simon Lucey. GPS-denied UAV localization using pre-existing satellite imagery. In *2019 International Conference on Robotics and Automation*, pages 2974–2980. IEEE, 2019. 1, 2

[13] Heng Guo, Shuaicheng Liu, Tong He, Shuyuan Zhu, Bing Zeng, and Moncef Gabbouj. Joint video stitching and stabilization from moving cameras. *IEEE Transactions on Image Processing*, 25(11):5491–5503, 2016. 1

[14] 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, 3, 4, 5, 6, 7, 8

[15] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 2, 5, 6

[16] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics*, 32(4):1–10, 2013. 1

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[18] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 6, 7

[19] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial intelligence*. Vancouver, British Columbia, 1981. 1, 2, 3

[20] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2527–2536, 2019. 6

[21] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *Proceedings of the European Conference on Computer Vision*, pages 168–183, 2018. 6

[22] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of International Conference on Machine Learning*, volume 30, page 3. Citeseer, 2013. 4

[23] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision*, pages 284–300, 2018. 6

[24] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 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, 4, 6

[26] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006. 2

[27] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. *arXiv preprint arXiv:1805.09662*, 2018. 6

[28] 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. 2, 4, 5, 6

[29] Gilles Simon, Andrew W Fitzgibbon, and Andrew Zisserman. Markerless tracking using planar structures in the scene. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pages 120–128. IEEE, 2000. 1

[30] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 8

[31] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 3, 6, 7, 8

[32] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006. 1, 2

[33] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 2, 3, 4

[34] Chaoyang Wang, Hamed Kiani Galoogahi, Chen-Hsuan Lin, and Simon Lucey. Deep-LK for efficient adaptive object tracking. In *2018 IEEE International Conference on Robotics and Automation*, pages 627–634. IEEE, 2018. 2

[35] Dan Wang, Chongyang Zhang, Hao Cheng, Yanfeng Shang, and Lin Mei. SPID: surveillance pedestrian image dataset and performance evaluation for pedestrian detection. In *Asian Conference on Computer Vision*, pages 463–477. Springer, 2016. 2, 5

[36] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, pages 3–19, 2018. 4

[37] Jiacheng Ying, Hui-Liang Shen, and Si-Yuan Cao. Unaligned hyperspectral image fusion via registration and interpolation modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 2021. 1

[38] Rui Zeng, Simon Denman, Sridha Sridharan, and Clinton Fookes. Rethinking planar homography estimation using perspective fields. In *Asian Conference on Computer Vision*, pages 571–586. Springer, 2018. 6

[39] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *European Conference on Computer Vision*, pages 653–669. Springer, 2020. 2, 3, 5, 6, 7, 8

[40] 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, 3, 4, 5, 6, 7, 8

[41] Qiang Zhou and Xin Li. STN-homography: Direct estimation of homography parameters for image pairs. *Applied Sciences*, 9(23):5187, 2019. 1, 2, 4, 6

[42] Yuan Zhou, Anand Rangarajan, and Paul D Gader. An integrated approach to registration and fusion of hyperspectral and multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3020–3033, 2019. 1

[43] Barbara Zitova and Jan Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21(11):977–1000, 2003. 2