Motion Basis Learning for Unsupervised Deep Homography Estimation with Subspace Projection

Nianjin Ye[1] Chuan Wang[1] Haoqiang Fan[1] Shuaicheng Liu[2,1*]

[1]Megvii Technology
[2]University of Electronic Science and Technology of China

Abstract

In this paper, we introduce a new framework for unsupervised deep homography estimation. Our contributions are 3 folds. First, unlike previous methods that regress 4 offsets for a homography, we propose a homography flow representation, which can be estimated by a weighted sum of 8 pre-defined homography flow bases. Second, considering a homography contains 8 Degree-of-Freedoms (DOFs) that is much less than the rank of the network features, we propose a Low Rank Representation (LRR) block that reduces the feature rank, so that features corresponding to the dominant motions are retained while others are rejected. Last, we propose a Feature Identity Loss (FIL) to enforce the learned image feature warp-equivariant, meaning that the result should be identical if the order of warp operation and feature extraction is swapped. With this constraint, the unsupervised optimization is achieved more effectively and more stable features are learned. Extensive experiments are conducted to demonstrate the effectiveness of all the newly proposed components, and results show that our approach outperforms the state-of-the-art on the homography benchmark datasets both qualitatively and quantitatively. Code is available at https://github.com/megvii-research/BasesHomo

1. Introduction

Homography is a fundamental and important image alignment model that has been widely used for image registration [1]. A homography is a 3×3 matrix that contains 8 Degree-of-Freedoms (DOFs), with each 2 for scale, translation, rotation and perspective [9]. Traditionally, a homography is often estimated by detecting and matching image features [16, 20], and then solving a Direct Linear Transform (DLT) [9] with outlier removal [7]. In contrast, deep homography methods take two images as the network input, and directly output a homography matrix [5]. Compared with traditional methods that highly rely on the extracted feature matches, deep methods are more robust.

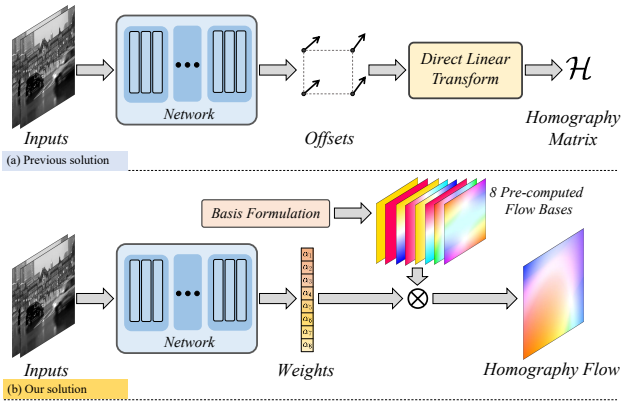

Figure 1. (a) Previous deep homography methods estimate 4 motion offsets and solve a DLT for the result. (b) We construct 8 flow motion bases by modifying matrix elements of a homography, and then regress 8 weights to combine the flow bases for the result.

Deep methods can be classified into two categories, supervised [5, 14] and unsupervised [28, 18]. The former one adopts synthesized examples with ground-truth labels to train the network while the latter one directly minimizes the photometric or feature differences between two images. As synthesized examples cannot reflect scene parallax and dynamic objects, unsupervised methods often generalize better than the supervised ones. For unsupervised methods, Nguyen *et al*. [18] minimized error over the entire image while Zhang *et al*. [28] proposed to learn a mask to skip outlier regions during the minimization.

It is not optimal to directly regress the elements of a homography matrix, as they are with different magnitudes. Current solution is to regress 4 offsets [5, 14, 28, 18], which is equivalent to a homography if feeds them to the DLT solver (Fig. 1(a)). In this work, we start from a new direction by proposing a "*homography flow*" representation (Fig. 1(b)). Specifically, we first generate 8 flow bases by modifying the entries of a homography matrix one at

*Corresponding author

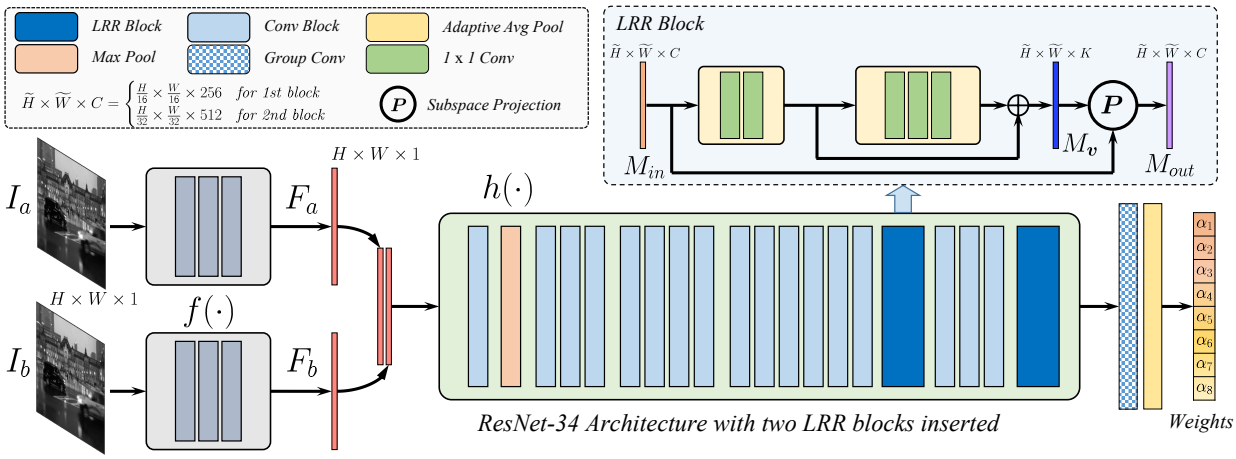

Figure 2. Our network pipeline takes grayscale image patches I_a and I_b as input, and produces 8 weights to combine 8 pre-defined homography bases to produce a homography flow as output. The network consists of two modules, a warp-equivariant feature extractor $f(\cdot)$ and a homography estimator $h(\cdot)$, with 2 inserted LRR blocks to reduce the rank of the motion features.

a time. As such, 8 homography matrices are obtained, each of which can be further translated into a flow map given the image coordinates, yielding 8 homography flow bases. In **small-baseline** scenarios, a homography flow can be reconstructed inside the space spanned by these flow bases by learning combinational weights.

As homography has only 8 DOFs, the homography flow lies in a low-rank subspace. However, the rank of the motion features through a network are usually much higher than that of a homography. From this observation, we propose to decrease the rank of the features by projecting them into their subspaces. Specifically, the projection contains two steps, including discovering the subspace bases of the features maps and then transforming feature maps into the subspace. To achieve this projection, we propose a Low Rank Representation (**LRR**) block, that can be plugged into a normal CNN and be trained end-to-end for the feature rank reduction. When the rank is reduced, features corresponding to the dominant motions, *i.e.* motions that could be described by a homography, are often retained. Features induced from non-dominant motions, e.g., multi-depth and dynamic contents, are often removed or suppressed.

Besides, the triplet loss of previous method still introduces trivial solutions [28]. Specifically, the feature warp-equivariance cannot be well preserved during the unsupervised training, while this property should have held ideally, *i.e.* $f(\mathcal{W}(I)) = \mathcal{W}(f(I))$, where $\mathcal{W}, f(\cdot)$ represent the warp operation and feature extraction. The lack of feature warp-equivariance results in the incorrect optimization of triplet loss, whose convergence direction is dominated by the distances between target features (anchors) and source features (negatives). However, the closer distances between target features (anchors) and warped source features (positives) are more essential regarding the alignment task. To this end, we propose a "*Feature Identity Loss*" (**FIL**) to enforce the image feature to be warp-equivariant. It is demonstrated that with FIL, our model can achieve more effective unsupervised optimization and learn more stable features.

We demonstrate the effectiveness of all the newly proposed techniques and components by extensive experiments and ablation studies. The experimental results also verify that our method outperforms the state-of-the-arts on the public benchmark both qualitatively and quantitatively. To sum up, our contributions are as follows:

- We propose a new representation "homography flow" that assembles 8 pre-computed flow bases for unsupervised deep homography estimation.
- We propose a new LRR block that reduces motion feature rank so as to reject motion noises implicitly.
- We propose a new FIL loss that enforces the warp-equivariance of the learned image feature to facilitate a stable unsupervised optimization.

2. Related works

Traditional homography. A homography is often estimated by first detecting and matching image features, such as SIFT [16], ORB [20], SURF [3], LPM [17], GMS [4], SOSNet [24], LIFT [26], BEBLID [22] and OAN [27], after which two sets of point correspondences were established. Next, the false matches were rejected by RANSAC [7], IRLS [11], MAGSAC [2]. Finally, a DLT is solved for the homography [8]. Some deep approaches have been proposed to improve the feature detection, e.g., SuperPoint [6] or matching, e.g., SuperGlue [21].

Deep homography. The deep homography can be classified into supervised [5, 14] and unsupervised [18, 28] methods. Compared to supervised approaches which learn transformation from synthesized images that lack depth disparities, the unsupervised ones can be trained on real image

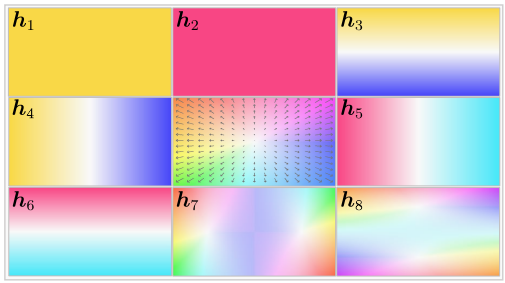

Figure 3. The visualization of the pre-computed 8 orthogonal and normalized flow bases $\boldsymbol{h}_1 \sim \boldsymbol{h}_8$ and the flow legend in the center.

pairs by minimizing the photometric loss between two images, with Spatial Transform Network (STN) [12] to warp the source to the target. Nguyen *et al.* [18] minimizes the photometric loss on the entire image while Zhang *et al.* [28] learns a mask to skip outlier regions.

Bases learning. Our method is also related to bases learning [15]. Tang *et al.* shows that there are subspaces that can be exploited for regularization in low-level vision problems [23]. PCAFlow learns flow bases from movies, showing that the flow estimation can be converted into the learning of the weighted sum of the learned flow bases [25]. Inspired by these works, we learn the coefficients to combine 8 pre-defined flow bases to estimate a homography flow.

3. Algorithm

3.1. Network Structure

Our method is built on convolutional neural networks, which takes two gray image patches I_a and I_b of size $H \times W$ as input, and produces a homography flow $\boldsymbol{H}_{ab}$ from I_a to I_b of the same size as output. The entire structure consists of two modules, a warp-equivariant feature extractor $f(\cdot)$ and a homography estimator $h(\cdot)$. $f(\cdot)$ is a fully convolutional network which accepts input of arbitrary sizes, and $h(\cdot)$ adapts a backbone of ResNet-34 [10] with our newly introduced LRR blocks and produces 8 weights, which are used to linearly combine 8 pre-computed flow bases to obtain $\boldsymbol{H}_{ab}$. Fig. 2 illustrates the network structure.

Homography flow and its basis formulation. A homography matrix has 8 DOFs and it is computed by solving a DLT after 4 corner offsets of an image are predicted as in [28, 18]. In this paper, we solve the problem from a new perspective. Specifically, our network learns a special optical flow of size $H \times W \times 2$ constrained by the homography, called "*homography flow*". Due to the constraint, homography flow falls into a 8-D subspace within the entire $2HW$-D space of a general optical flow. It can be represented by 8 orthogonal flow bases spanning the subspace, *i.e.*

$$\begin{gathered} \exists\, \{\boldsymbol{h}_i\} \ \text{ s.t. } \ \boldsymbol{h}_{ab} = \sum_i \alpha_i \boldsymbol{h}_i \ \ (i = 1, 2, ..., 8) \\ \text{where } \boldsymbol{h}_i \in \mathbb{R}^{2HW}, \ \boldsymbol{h}_i^T \boldsymbol{h}_j = 0 \end{gathered} \tag{1}$$

Here $\boldsymbol{h}_{ab}$ is the flattened version of $\boldsymbol{H}_{ab}$, and $\{\alpha_i\}$ are the coefficients for the flow bases.

To obtain the orthogonal flow bases, we first generate 8 homography matrices by modifying each single entry h_i of an identity homography matrix, except the entry at the position of $(3, 3)$ which is always normalized to 1. Given the image coordinates, a homography matrix can be converted to a flow map by transforming the image coordinates and minus their original positions. Then, 8 homography flows are normalized by their largest flow magnitude followed by a QR decomposition. Mathematically, it is described as,

$$\boldsymbol{M} = \boldsymbol{Q} \cdot \boldsymbol{R} \quad (\boldsymbol{M}, \boldsymbol{Q} \in \mathbb{R}^{2HW \times 8}, \boldsymbol{R} \in \mathbb{R}^{8 \times 8}) \tag{2}$$

where each column of matrix $\boldsymbol{M}$ is the flattened normalized homography flow $\boldsymbol{H}_i$ as mentioned above. By QR decomposition, columns of $\boldsymbol{Q}$ are orthogonal and they naturally serve as the flow bases spanning the homography subspace, *i.e.* $\boldsymbol{Q} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_8]$. In other words, each flow basis is associated with a tangent space at the origin of the homography group. With the 8 bases formulated, a homography flow can be acquired by accurately predicting their weights $\{\alpha_i\}$. Considering the perspective transformation can be approximated well with linear model in small baseline tasks, we can use such a linear-weighted solution to approximate homography. We visualize the bases in Fig. 3.

Discussion. Compared with the aforementioned bases learning methods like PCAFlow [25], our method has similarity to them so that a set of homography bases could be potentially learned. However, our solution has its specificity because unlike PCAFlow [25] which predicts a general optical flow that requires more flexibility, we only deal with the background of *small-baseline* scenes. It means that less flexibility exists in the solution space so that an analytical derivation of "homography flow" becomes feasible and complicated learning tools like PCA [19] could be unemployed. So for simplicity, we just employ the pre-computed bases in this work, although there indeed exist types of homographies that cannot be represented precisely by them, such as those in large-baseline scenes.

Warp-equivariant feature extractor. Before [28], previous unsupervised DNN based methods commonly minimize the pixel intensity values for the registration. In [28], the authors proposed to minimize the difference of learned deep features instead of using the original images. In this paper, we similarly follow the idea of [28], but constrain the learned features with warp-equivariance, which means the results should be identical if we swap the order of warp operation $\mathcal{W}$ and feature extraction f given an input image I, *i.e.* $\mathcal{W}\big(f(I)\big) = f\big(\mathcal{W}(I)\big)$. For inputs I_a and I_b, the feature extractor shares weights and produces feature maps F_a and F_b. In practice, features with absolute warp-equivariance are rarely achieved. Thus we introduce a new

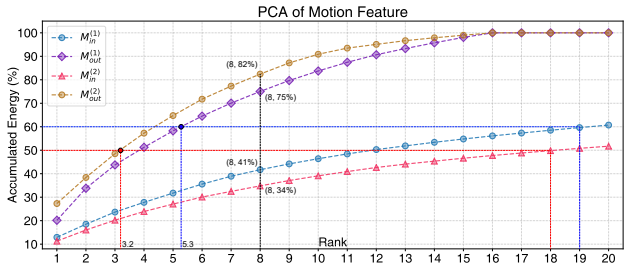

Figure 4. Accumulated energy of the principal components of the motion features, before and after the 1st and 2nd LRR blocks.

loss $\mathbf{L}_{\mathcal{W}} = \left|\mathcal{W}\big(f(I)\big) - f\big(\mathcal{W}(I)\big)\right|$ as a constraint to approximate this property, which is detailed in Sec. 3.3.

Homography estimator with LRR blocks. Given the feature maps F_a and F_b, we concatenate them to build a feature map $[F_a, F_b]$. Then, it is fed to the homography estimator network to produce 8 weights. These weights linearly combine $\{\boldsymbol{h}_i\}$ to produce the final homography flow $\boldsymbol{H}_{ab}$. We use $h(\cdot)$ to represent the whole process, *i.e.*

$$\boldsymbol{H}_{ab} = h([F_a, F_b]) \tag{3}$$

The backbone of $h(\cdot)$ generally follows a ResNet-34 [10] structure, except that two newly introduced LRR blocks are inserted at two layers. Each LRR block consists of shallow residual convolution layers and learns K bases for the input motion feature forwarded by the preceding layers. It then generates an output motion feature of rank at most K by subspace projection. Specifically, given an input motion feature $M_{in} \in \mathbb{R}^{H\times W\times C}$, the residual convolution layers convert it into a feature $M_{\boldsymbol{v}} \in \mathbb{R}^{H\times W\times K}$ of K channels. Then each channel serves as a feature basis $\boldsymbol{v}_k \in \mathbb{R}^{HW}, k = 1, 2, ..., K$ after being flattened. Finally, we project M_{in} into the subspace of the feature bases to obtain a low-rank motion feature $M_{out} \in \mathbb{R}^{H\times W\times C}$, *i.e.*

$$M_{out} = \boldsymbol{V}(\boldsymbol{V}^T\boldsymbol{V})^{-1}\boldsymbol{V}^T \cdot M_{in} \tag{4}$$

where $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_K] \in \mathbb{R}^{HW\times K}$. Note that the normalization term $(\boldsymbol{V}^T\boldsymbol{V})^{-1}$ is required since the feature bases $\{\boldsymbol{v}_k\}$ are not ensured orthogonal.

After the 2nd LRR block, the motion feature computed is forwarded to a group convolution and an adaptive pooling layer, to produce the final 8 weights $\{\alpha_i\}$ for homography flow bases combination. We illustrate the structure of LRR block in Fig. 2, and K is set to 16 in all of our experiments.

3.2. Robust Homography Estimation by LRR

As indicated in Sec. 3.1, a homography flow is of low rank, which means the rank of the motion features through various layers in $h(\cdot)$ should be reduced. Our observation is that, if the rank of the motion feature is reduced, the latent weights for the homography flow bases could be predicted more accurately and easily, where motion outliers are excluded during the rank reduction.

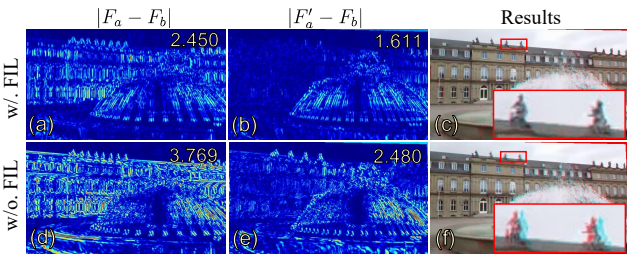

Figure 5. The comparison between with and without the warp-equivariance by FIL. Please refer the text for more details.

The motion outliers are motions caused by dynamic contents or non-planar depth variations outside the solution space of a single homography. Traditionally, motion outliers are often rejected by RANSAC [16]. In DNN, Zhang *et al.*'s [28] predicted a mask to skip the motion outliers. In this paper, LRR block serves this purpose. It reduces the rank of the motion feature, during which the feature ranks corresponding to the outliers are reduced. Because we enforce the network to learn motions spanned by the homography bases, any motions outside the subspace are treated as motion noise. In this way, the non-homography motion can be eliminated automatically during the rank reduction. As a result, a mask predictor as is no longer needed [28].

As seen in Table 1, with LRR block inserted, our network produces a lower error and performs much better compared to [28]. We also analyze the accumulated energy of the principal components for the motion feature M_{in} and M_{out}. As seen in Fig. 4, after the 1st LRR block, the number of principal components (NPC) are reduced from 19 to 5.3 in terms of accumulated energy being 60%. The effectiveness of the 2nd LRR block is more obvious, reducing the NPC from 18 to 3.2 in terms of 50% energy, reflecting the rank of the motion feature is highly reduced.

3.3. Triplet Loss with Feature Warp-Equivariance

With the homography flow $\boldsymbol{H}_{ab}$ estimated, we warp feature map F_a to F'_a and formulate a triplet loss without involving an attention mask as in [28], *i.e.*

$$\mathbf{L}_{\mathcal{T}}(I_a, I_b) = \mathbf{L}_{\mathcal{T}}^{ab} = |F'_a - F_b|_1 - |F_a - F_b|_1 \tag{5}$$

The original idea on the triplet loss tries to learn a discriminative feature and an accurate homography simultaneously to well align the input images. Even though it has been demonstrated successful in most cases in [28], it still has the potential to be incorrectly optimized so that $|F_a - F_b|$ is over-maximized while $|F'_a - F_b|$ is still under-minimized, due to the enough DOFs of $f(\cdot)$. To this end, we add a new constraint named "*Feature Identity Loss*" (FIL) to preserve the warp-equivariance of the learned feature, meaning that the final feature should be approximately identical if we swap the order of warp operation and feature extraction, *i.e.*

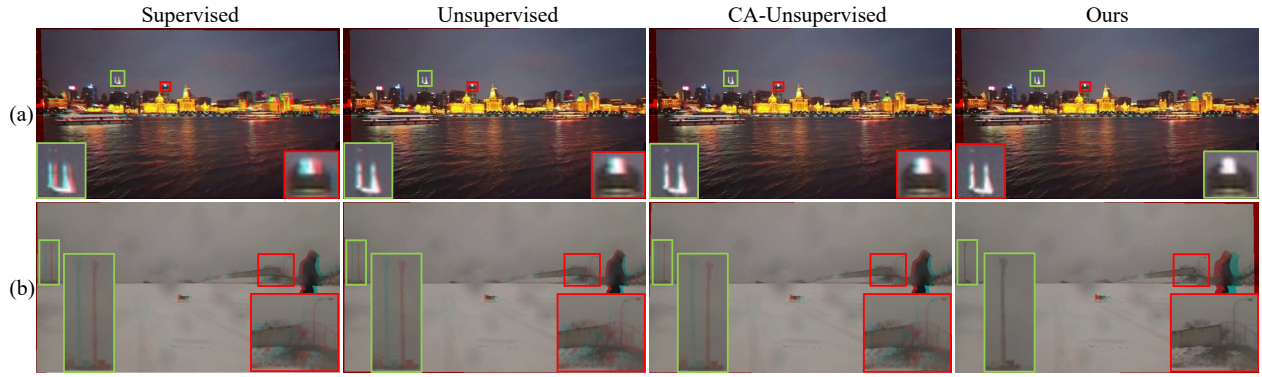

Figure 6. Qualitative comparison with recent DNN-based approaches. Columns $1 \sim 4$ are results of supervised [5], Unsupervised [18], CA-Unsupervised [28] and ours. The alignment difficulty of Rows 2 is greater than Row 1.

$$\mathbf{L}_{\mathcal{W}}(I_a, f, \mathcal{W}_{ab}) = \mathbf{L}_{\mathcal{W}}^{ab} = |\mathcal{W}_{ab}(f(I_a)) - f(\mathcal{W}_{ab}(I_a))|_1 \quad (6)$$

where $\mathcal{W}_{ab}$ is the warp operation by homography flow $\boldsymbol{H}_{ab}$.

We observe that with this constraint, the optimization of $f(\cdot)$ can be stabilized, improving the accuracy of estimated homography flow. We visualize an example in Fig. 5, where without FIL involved, the triplet loss $\mathbf{L}_{\mathcal{T}}(I_a, I_b)$ is less than the one optimized with FIL, even though $|F'_a - F_b|$ becomes larger. The reason behind is that the term $|F_a - F_b|$ is maximized excessively without FIL. As a result, the alignment accuracy is down-graded.

In practice, we also swap the order of I_a and I_b to compute symmetric losses $\mathbf{L}(I_b, I_a)$ and $\mathbf{L}_{\mathcal{W}}(I_b, f, \mathcal{W}_{ab})$, and add a constraint to enforce the homograph flows $\boldsymbol{H}_{ba}$ and $\boldsymbol{H}_{ab}$ to be inverse. So, the energy is written as follows,

$$\min_{f,h} \ (\mathbf{L}_{\mathcal{T}}^{ab} + \mathbf{L}_{\mathcal{T}}^{ba}) + \lambda(\mathbf{L}_{\mathcal{W}}^{ab} + \mathbf{L}_{\mathcal{W}}^{ba}) + \mu|\boldsymbol{H}_{ab} + \boldsymbol{H}_{ba}|_2^2 \quad (7)$$

where λ and μ are set to 1.0 and 0.001 in our experiments.

4. Experiment

4.1. Dataset and Implementation Details

We evaluate our method using the same dataset as in Zhang *et al.*'s [28], *i.e.* the CA-Unsupervised. The training set consists of 5 categories of small-baseline image pairs in real scenes, including regular (RE), low-texture (LT), low-light (LL), small-foregrounds (SF), and large-foregrounds (LF). Except for the RE, the other 4 scenes are challenging for the homography estimation. A randomly selected subset of $4.2k$ samples is used as the test set, each sample of which contains 6 pairs of labeled matching points for evaluation.

Our network is trained with $360k$ iterations by the Adam optimizer [13], with parameters $l_r = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The batch size is set to 16 and the l_r is reduced by 20% every epoch. The implementation is based on PyTorch and the network training is performed on one NVIDIA RTX 2080 Ti. To alleviate the impact of empty boundaries in the warped image, we randomly crop patches of size 320×576 from the original images to serve as input.

4.2. Comparison to Existing Methods

We compare our method with two groups of homography estimation approaches, the DNN-based ones and the feature-based ones. The former group includes Supervised [5], Unsupervised [18] and CA-Unsupervised [28], and the latter group includes 14 methods, including 12 combinations of 6 types of features (3 traditional: SIFT [16] / ORB [20] / BEBLID [22] and 3 DNN-based: LIFT [26] / SOSNet [24] / SuperPoint [6]) and 2 outlier rejection algorithms (RANSAC [7] / MAGSAC [2]), as well as 2 additional customized descriptor matching approaches SuperGlue [21] specifically for SuperPoint [6] only.

Qualitative comparison. Fig. 6 shows the comparison with DNN-based methods. Fig. 6(a) is from the LL category with repetitive dynamic textures at the river, where the Supervised [5] fails because it is trained on synthetic data without dynamic contents. The results of the Unsupervised [5] and the CA-Unsupervised [28] contain some small errors since both methods cannot reject the dynamic flowing rivers precisely. In contrast, our outlier rejection is achieved via LRR, obtaining accurate principal feature attention compared to the mask of CA-Unsupervised [28].

Our method shows superiority in Fig. 6(b). This scene is from the LT category, where texture quality is extremely poor as in the snow and the sky. The walking man in the foreground making the task much more challenging. Other methods try to align the moving man since he has more textures than his surrounding areas, while only our method successfully aligns the scene without paying attention to the man, demonstrated by the highlighted pole region.

Fig. 7 further compares our method with all the aforementioned feature-based solutions. In Fig. 7(a), (d) and (g), we validate various feature-based methods in a snow scene, where all of the feature-based methods fail to produce satisfying results, due to either the feature extraction or the foreground interferences. In contrast, our method aligns this

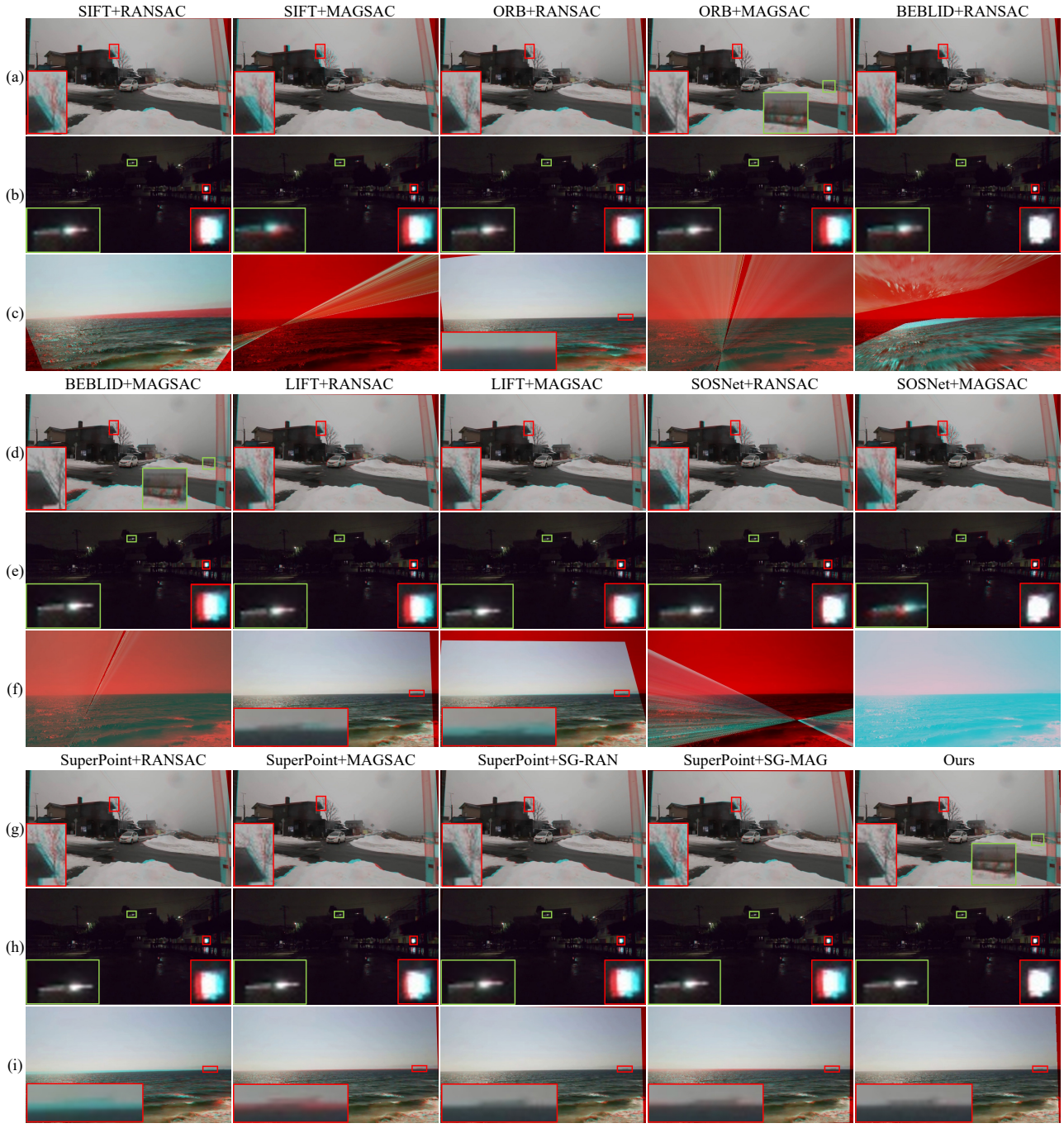

Figure 7. Qualitative comparison with with Feature-based approaches on 3 examples. The combination of various descriptors and outlier elimination methods produced a total of 14 approaches.

scene more accurately. As for the latter two examples including a low-light and a low-texture scene, both of them challenge the feature detection and matching. For instance, in the low-light scene (Fig. 7(b), (e) and (h)), only a small portion of the image contains salient regions. In the sea example (Fig. 7(c), (f) and (i)), it is tough to obtain reliable feature matches on the sea textures. In contrast, our method is naturally more adaptable to the case of insufficient features, benefiting from the pursuit of low-rank features.

Quantitative comparison. As seen in Table 1, for each pair of test images, the average l_2 distances between the warped source points and the target points are considered as the error metric. We report the errors with respect to each scene category. Specifically, Row 3 $\sim$ 5 are deep homography methods; Row 6 $\sim$ 11 are traditional feature-based methods and Row 12 $\sim$ 19 are DNN-based feature meth-

1)		RE	LT	LL	SF	LF	Avg
2)	$\mathcal{I}_{3\times3}$	7.75(+2483.33%)	7.65(+868.35%)	7.21(+930.00%)	7.53(+960.56%)	3.39(+621.28%)	6.70(+963.49%)
3)	Supervised [5]	1.51(+403.33%)	4.48(+467.09%)	2.76(+294.29%)	2.62(+269.01%)	3.00(+538.30%)	2.87(+355.56%)
4)	Unsupervised [18]	0.79(+163.33%)	2.45(+210.13%)	1.48(+111.43%)	1.11(+56.34%)	1.10(+134.04%)	1.39(+120.63%)
5)	CA-Unsupervised [28]	0.73(+143.33%)	1.01(+27.85%)	1.03(+47.14%)	0.92(+29.58%)	0.70(+48.94%)	0.88(+39.68%)
6)	SIFT [16] + RANSAC [7]	0.30(+0.00%)	1.34(+69.62%)	4.03(+475.71%)	0.81(+14.08%)	0.57(+21.28%)	1.41(+123.81%)
7)	SIFT [16] + MAGSAC [2]	0.31(+3.33%)	1.72(+117.72%)	3.39(+384.29%)	0.80(+14.08%)	0.47(+0.00%)	1.34(+112.70%)
8)	ORB [20] + RANSAC [7]	0.85(+183.33%)	2.59(+227.85%)	1.67(+138.57%)	1.10(+54.03%)	1.24(+163.83%)	1.48(+134.92%)
9)	ORB [20] + MAGSAC [2]	0.97(+223.33%)	3.34(+322.78%)	1.58(+125.71%)	1.15(+61.97%)	1.40(+197.87%)	1.69(+168.25%)
10)	BEBLID [22] + RANSAC [7]	0.78(+160.00%)	2.83(+258.23%)	1.38(+97.14%)	1.04(+46.48%)	1.33(+182.98%)	1.47(+133.33%)
11)	BEBLID [22] + MAGSAC [2]	0.94(+213.33%)	3.73(+372.15%)	3.49(+398.57%)	1.17(+64.79%)	1.25(+165.96%)	2.12(+236.51%)
12)	LIFT [26] + RANSAC [7]	0.40(+33.33%)	2.01(+154.43%)	1.14(+62.86%)	0.77(+8.45%)	0.68(+44.68%)	1.00(+58.73%)
13)	LIFT [26] + MAGSAC [2]	0.35(+16.67%)	1.85(+134.18%)	0.96(+37.14%)	0.72(+1.41%)	0.50(+6.38%)	0.88(+39.68%)
14)	SOSNet [24] + RANSAC [7]	0.29(-3.33%)	2.42(+206.33%)	3.71(+430.00%)	0.77(+8.45%)	0.59(+25.53%)	1.56(+147.62%)
15)	SOSNet [24] + MAGSAC [2]	0.30(+0.00%)	3.00(+279.75%)	3.66(+422.86%)	0.87(+22.54%)	0.49(+4.26%)	1.67(+165.08%)
16)	SuperPoint [6] + RANSAC [7]	0.43(+43.33%)	0.85(+7.59%)	0.77(+10.00%)	0.84(+18.31%)	0.80(+70.21%)	0.74(+17.46%)
17)	SuperPoint [6] + MAGSAC [2]	0.45(+50.00%)	0.90(+13.92%)	0.77(+10.00%)	0.76(+7.04%)	0.67(+42.55%)	0.71(+12.70%)
18)	SuperPoint [6] + SG-RAN [21] [7]	0.41(+36.67%)	0.87(+10.13%)	0.72(+2.86%)	0.80(+12.68%)	0.75(+59.57%)	0.71(+12.70%)
19)	SuperPoint [6] + SG-MAG [21] [2]	0.36(+20.00%)	0.79(+0.00%)	0.70(+0.00%)	0.71(+0.00%)	0.70(+48.94%)	0.63(+0.00%)
20)	Ours	0.29(-3.33%)	0.54(-31.65%)	0.65(-7.14%)	0.61(-14.08%)	0.41(-12.77%)	0.50(-20.63%)

Table 1. Comparison of point matching errors between ours and all other methods. SG-RAN and SG-MAG are SuperGlue [21] + RANSAC [7] and SuperGlue [21] + MAGSAC [2] respectively. The percentage in the bracket indicates the improvements over the second best results. Red indicates the best performance and Blue refers to the second best result.

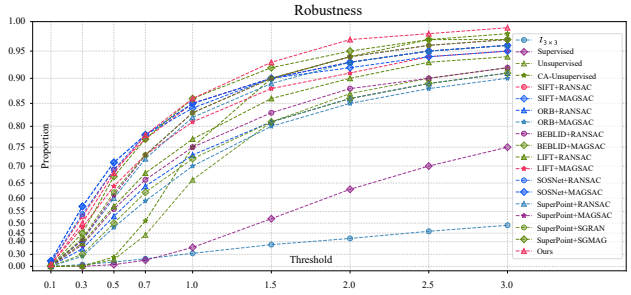

Figure 8. The proportion of inliers in all matching point pairs under various thresholds for all methods. Higher position of curves represent higher robustness.

ods. $\mathcal{I}_{3\times3}$ refers to an identity homography, which reflects the original distance between point pairs.

As for the RE scene, the abundant texture provides sufficient high-quality features for homography estimation. So that the feature-based solutions show obvious advantages in this category. Nevertheless, our method and the combination of SOSNet [24] and RANSAC [7] outperform the others and achieve the lowest error of 0.29.

For the scenes of LT and LL, most of the feature-based solutions become less robust due to the difficulty in extracting effective features. In contrast, our method consistently works well. In particular, the results of the 2nd best method which is constituted by the 3 latest algorithms (SuperPoint [6], SuperGlue [21] and MAGSAC [2]) achieve strong performance. In the scenes containing small (SF) and large (LF) foreground, although sufficient texture features are available, dynamic objects and multi-plane occlusions cause troubles for the outlier removal. In our method, objects and multi-plane depth tend to introduce high-rank features in the encoding space, which are abandoned by our low-rank property of LRR blocks. Therefore, our method achieves at least 14.08% and 12.77% lower errors than others on SF and LF, respectively. On the whole, the combination of SuperPoint [6], SuperGlue [21] and MAGSAC [2] produces rather competitive results for all the scenes, but their average errors are still higher than ours by 20.63%.

Robustness evaluation. Furthermore, we evaluate the robustness of all the methods by setting a threshold to judge if a homography matches the marked points. Points with errors lower than the threshold are considered matched inliers, otherwise they are judged as outliers. As such, we report a percentage of matched points overall marked points given a homography estimation method and a threshold, so that a series of curves are reported in Fig. 8 by setting threshold to 0.1 to 3.0. As seen, our method (the red curve) achieves the highest inlier percentage if the threshold is greater than 0.8, indicating our method can handle intractable cases better than the others. It draws a similar conclusion as Table 1 that our method outperforms others in challenging scenarios, such as LT and LF scenes.

4.3. Ablation Studies

We verify the effectiveness of all three contributions by ablation studies, and report the results in Table 2.

Homography flow vs. offsets. The 2nd row of Table 2 shows the result of CA-Unsupervised [28] that adopts offset representation with mask outlier removal. We replace its regression target from the offset to the weights of our homography flow bases, and observe a lower average error and comparable errors in LT, LL and SF scenes being achieved in Row 3. It demonstrates that even for a network not specially designed, the homography flow representation outperforms the old offset solution.

1)	Mask/LRR	Offset/Basis	FIL	RE	LT	LL	SF	LF	Avg
2)	Mask	Offset		0.73(+151.72%)	1.01(+87.04%)	1.03(+58.46%)	0.92(+50.82%)	0.70(+70.73%)	0.88(+76.00%)
3)	Mask	Basis		0.53(+82.76%)	1.05(+94.44%)	1.04(+60.00%)	0.95(+55.74%)	0.56(+36.59%)	0.83(+66.00%)
4)	Mask	Basis	✓	0.45(+55.17%)	1.02(+88.89%)	0.93(+43.08%)	0.96(+57.38%)	0.50(+21.95%)	0.77(+54.00%)
5)	LRR	Basis		0.37(+27.59%)	0.69(+27.78%)	0.75(+15.38%)	0.75(+22.95%)	0.45(+9.76%)	0.60(+20.00%)
6)	LRR	Basis	✓	0.29(+0.00%)	0.54(+0.00%)	0.65(+0.00%)	0.61(+0.00%)	0.41(+0.00%)	0.50(+0.00%)

Table 2. The homography representation solution is chosen in Offset and Basis (homography flow), as well as the outlier rejection module is chosen as Mask [28] or LRR. The FIL as an optional promotion to the effect of the model.

	0 layer	1 layer	2 layers	3 layers	3 layers*
FIL	0.00	0.14	0.31	0.35	0.11
MSE	0.76	0.69	0.61	0.60	0.50

Table 3. Comparison of point matching errors when $f(\cdot)$ contains a different number of convolution layers. 0 layer means network without $f(\cdot)$. * indicates FIL is involved in training.

LRR blocks. We modify the network in Row 3 to our structure (Row 5) by removing the mask module and inserting LRR blocks into the homography estimator $h(\cdot)$. By this replacement, the combination of LRR blocks and homography flow produces a notable advantage, reducing the error by at least about **20%** in all scenarios. A reasonable explanation is that the LRR blocks encourage the utilization of low-rank features for homography estimation, which is conducive to features extraction and outlier rejection.

FIL for warp-equivariance. We also verify the effectiveness of FIL in two structures, *i.e.* the modified CA-Unsupervised [28] one as in Row 3 and ours as in Row 5, in Table 2. By comparing Row 3 with 4, and Row 5 with 6, we can observe that errors are reduced from 0.83 to 0.77 (-8%) and from 0.60 to 0.50 (-17%). Especially in the scenes LL and LF, the error reduction is more significant.

To further investigate how FIL improves the optimization, we conduct another experiment by modifying the number of convolutional layers in $f(\cdot)$, from 0 to 3. As seen in Table 3, from $f(\cdot)$ containing 0 to 2 layers without FIL involved in the optimization, the Mean Square Error (MSE) is gradually reduced while the FIL is increased, reflecting the warp-equivariance is damaged. Here "0 layers" means $f(\cdot)$ is removed and photometric loss is used instead. If we continue adding layer to 3, the MSE cannot be decreased showing that the network cannot be consistently optimized, while FIL keeps increasing. If we add FIL to the optimization goal, we can see the MSE is reduced from 0.6 to 0.5 and FIL is decreased significantly from 0.35 to 0.11. This phenomenon reflects that with warp-equivariance preserved, the optimization becomes more stable so that a higher performance can be achieved.

4.4. Generalization

As the fixed bases are obtained via mathematical derivation, its generation is independent of images, meaning that unseen images could be also covered as long as they are in small-baseline scenes. Fig. 9 depicts the alignment results of unseen photos taken with a mobile phone in scenes of low light and low texture, etc.

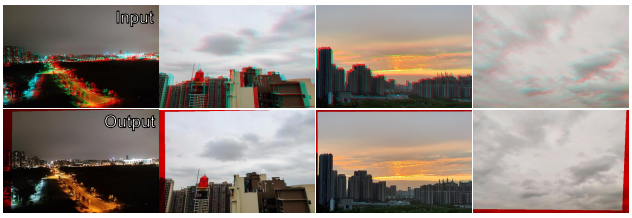

Figure 9. Photos taken by mobile phone.

4.5. Failure cases

The predefined flow bases is friendly to small-baseline scenarios. By contrast, it may introduce error when applied to the large baseline cases as bases h_7 and h_8 are a linear approximate representation of small range perspective transformation in the Cartesian coordinates.

5. Conclusion

We have presented a new deep solution for homography estimation, involving 3 new components to improve the performance of previous methods: a new representation called homography flow, a LRR block to reduce rank of features and a feature identity loss to stabilize the optimization process. Extensive experiments demonstrate the effectiveness of all the newly introduced components and the superior performance over the previous methods. Notwithstanding, our method has its limitations including that it may fail in large-baseline scenes, its single homography output may be insufficient for a real scene, and its fixed bases may limit its wider applications. We consider to extend the idea to mesh-based multi-homographies and explore the superiority of learned bases as our future work.

Acknowledgement

This research was supported in part by National Key R&D Plan of the Ministry of Science and Technology (Project No. 2020AAA0104400), in part by National Natural Science Foundation of China (NSFC) under grants No.61872067 and No.61720106004, in part by Research Programs of Science and Technology in Sichuan Province under grant No.2019YFH0016.

References

[1] Alex, M., and Andrew. Multiple view geometry in computer vision. *Kybernetes*, 1972. 1

[2] Daniel Barath, Jiri Matas, and Jana Noskova. MAGSAC: marginalizing sample consensus. In *Proc. CVPR*, pages 10197–10205, 2019. 2, 5, 7

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: speeded up robust features. In *Proc. ECCV*, pages 404–417, 2006. 2

[4] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proc. CVPR*, pages 4181–4190, 2017. 2

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

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proc. CVPRW*, pages 224–236, 2018. 2, 5, 7

[7] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 1, 2, 5, 7

[8] Andrew Harltey and Andrew Zisserman. *Multiple view geometry in computer vision (2. ed.)*. 2006. 2

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proc. ECCV*, pages 630–645, 2016. 3, 4

[11] Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics*, 6(9):813–827, 1977. 2

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proc. NeurIPS*, pages 2017–2025, 2015. 3

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 5

[14] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In *Proc. CVPR*, pages 7649–7658, 2020. 1, 2

[15] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2012. 3

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004. 1, 2, 4, 5, 7

[17] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *International Journal of Computer Vision*, 127(5):512–531, 2019. 2

[18] 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 Autom. Lett.*, 3(3):2346–2353, 2018. 1, 2, 3, 5, 7

[19] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. 3

[20] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. ICCV*, pages 2564–2571, 2011. 1, 2, 5, 7

[21] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proc. CVPR*, pages 4938–4947, 2020. 2, 5, 7

[22] Iago Suárez, Ghesn Sfeir, José M Buenaposada, and Luis Baumela. Beblid: Boosted efficient binary local image descriptor. *Pattern Recognition Letters*, 133:366–372, 2020. 2, 5, 7

[23] Chengzhou Tang, Lu Yuan, and Ping Tan. Lsm: Learning subspace minimization for low-level vision. In *Proc. CVPR*, pages 6235–6246, 2020. 3

[24] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proc. CVPR*, pages 11016–11025, 2019. 2, 5, 7

[25] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proc. CVPR*, pages 120–130, 2015. 3

[26] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: learned invariant feature transform. In *Proc. ECCV*, volume 9910, pages 467–483, 2016. 2, 5, 7

[27] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proc. ICCV*, pages 5845–5854, 2019. 2

[28] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *Proc. ECCV*, pages 653–669, 2020. 1, 2, 3, 4, 5, 7, 8