Content-Aware Unsupervised Deep Homography Estimation and its Extensions

Shuaicheng Liu, *Member, IEEE*, Nianjin Ye, Chuan Wang, Jirong Zhang,
Lanpeng Jia, Kunming Luo, Jue Wang, *Senior Member, IEEE*, and Jian Sun, *Senior Member, IEEE*

Abstract—Homography estimation is a basic image alignment method in many applications. It is usually done by extracting and matching sparse feature points, which are error-prone in low-light and low-texture images. On the other hand, previous deep homography approaches use either synthetic images for supervised learning or aerial images for unsupervised learning, both ignoring the importance of handling depth disparities and moving objects in real-world applications. To overcome these problems, in this work, we propose an unsupervised deep homography method with a new architecture design. In the spirit of the RANSAC procedure in traditional methods, we specifically learn an outlier mask to only select reliable regions for homography estimation. We calculate loss with respect to our learned deep features instead of directly comparing image content as did previously. To achieve the unsupervised training, we also formulate a novel triplet loss customized for our network. We verify our method by conducting comprehensive comparisons on a new dataset that covers a wide range of scenes with varying degrees of difficulties for the task. Experimental results reveal that our method outperforms the state-of-the-art, including deep solutions and feature-based solutions.

Index Terms—Homography, deep homography, image alignment, RANSAC

1 Introduction

Homography is the foundation of stereo vision [1]. It can align images taken from different perspectives if they approximately undergo a rotational motion or the scene is close to a planar surface. For scenes that satisfy the constraints, a homography can align them directly. For scenes that violate the constraints, e.g., a scene that consists of multiple planes or contains moving objects, homography usually serves as an initial alignment model before more advanced models such as mesh flow [2] and optical flow [3]. Most of the time, such a pre-alignment is crucial for the final quality. As a result, the homography has been widely applied in vision tasks such as multi-frame high dynamic ranging (HDR) imaging [4], multi-frame image super resolution [5], burst image denoising [6], video stabilization [7], image/video stitching [8], [9], SLAM [10], [11], augmented reality [12] and camera calibration [13].

Homography estimation by traditional approaches generally requires matched image feature points such as SIFT [14]. Specifically, after a set of feature correspondences are obtained, a homography matrix is estimated by Direct Linear Transformation (DLT) [1] with RANSAC outlier rejection [16]. Feature-based methods commonly could achieve good performance while they highly rely on the quality of image features. Estimation could be inaccurate due to the insufficient number of matched points or poor distribution of the features, which is a common case due to the existence of textureless regions (e.g., blue sky and white wall), repetitive patterns or illumination variations. Moreover, the rejection of outlier points, e.g., point matches that are located on the non-dominant planes or dynamic objects, is also important for the high-quality result. Consequently, feature-based traditional homography estimation is usually a challenging task for the non-regular scenes as above.

Due to the development of deep neural networks (DNN) in recent years, DNN-based solutions to homography estimation are gradually proposed, such as supervised [17] and unsupervised [15] ones. For the former solution, it requires homography as ground truth (GT) to supervise the training so that only synthetic target images warped by the GT homography could be generated. Although the synthetic image pairs can be produced on an arbitrary scale, they are far from real cases because real depth disparities are unavailable in the training data. As such, this method suffers from bad generalization to real images. To tackle this issue, Nguyen *et al.* proposed the latter unsupervised solution [15], which minimizes the photometric loss on real image pairs. However, this method has two main problems. One is that the loss calculated with respect to image intensity is less effective than that in the feature space, and the

- *Shuaicheng Liu is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China. E-mail: liushuaicheng@uestc.edu.cn.*
- *Nianjin Ye, Lanpeng Jia, and Kunming Luo are with Megvii Technology, Chengdu, Sichuan 610095, China. E-mail: yenianjin@gmail.com, jialanpeng.mz@outlook.com, luokunming@megvii.com.*
- *Chuan Wang and Jian Sun are with Megvii Technology, Beijing 100080, China. E-mail: cwang.hku@gmail.com, sunjian@megvii.com.*
- *Jirong Zhang is with the Beijing Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing 100094, China. E-mail: zhangjirong.dgt@gmail.com.*
- *Jue Wang is with Tencent AI Lab, Shenzhen 518054, China. E-mail: arphid@gmail.com.*

Manuscript received 27 Feb. 2021; revised 23 Jan. 2022; accepted 30 Apr. 2022. Date of publication 10 May 2022; date of current version 3 Feb. 2023.
This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61872067 and 61720106004.
(Corresponding author: Chuan Wang.)
Recommended for acceptance by H. Li.
Digital Object Identifier no. 10.1109/TPAMI.2022.3174130

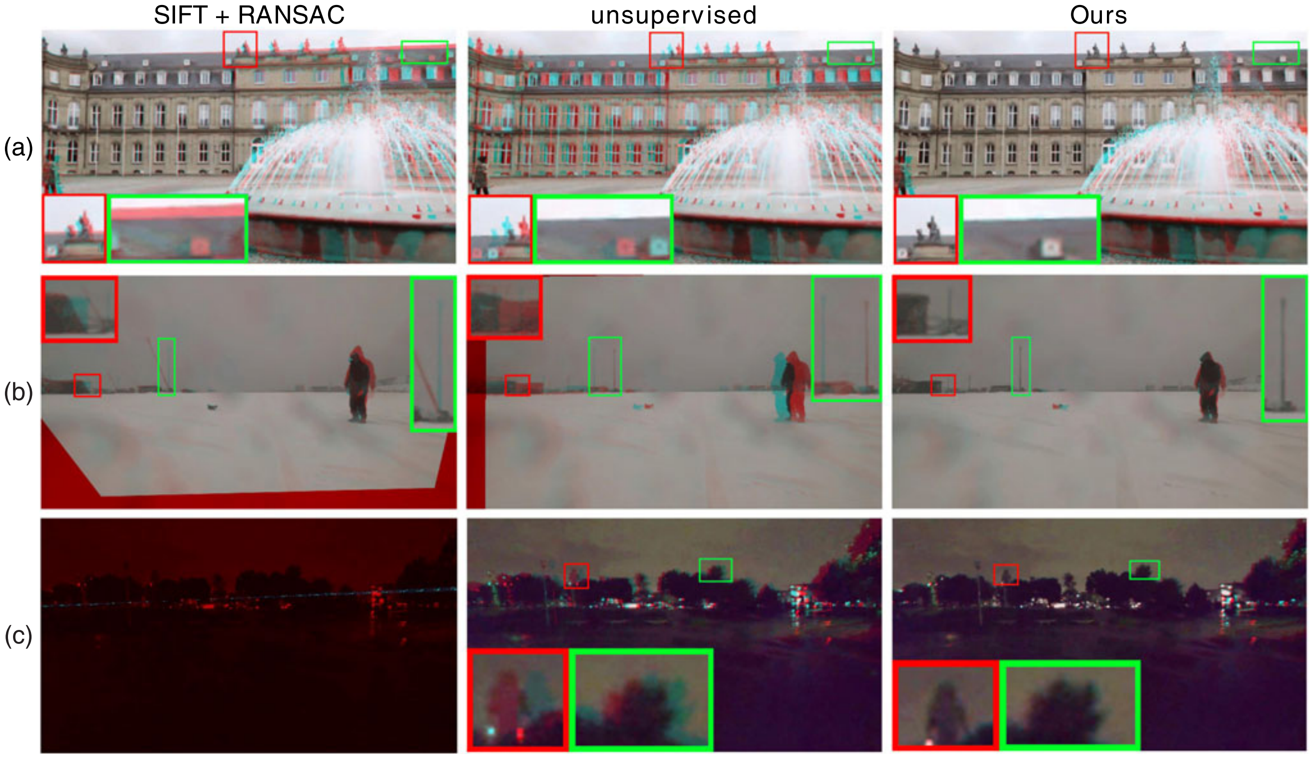


Fig. 1. Our deep homography estimation on challenging cases, compared with one traditional feature-based, i.e., SIFT [14] + RANSAC and one unsupervised DNN-based method [15]. (a) An example with dynamic foreground. (b) A low texture example. (c) A low light example. We mix the blue and green channels of the warped image and the red channel of the target image to obtain the visualization results as above, where the misaligned pixels appear as red or green ghosts. The same visualization method is applied for the rest of this paper.

loss is calculated uniformly in the entire image ignoring the RANSAC-like process. As a result, this method cannot exclude the moving or non-planar objects from the final loss, so as to potentially decrease the estimation accuracy. To avoid the above phenomenons, Nguyen *et al.* [15] has to work on aerial images that are far away from the camera to minimize the influence of depth variations of parallax.

To tackle the aforementioned issues, we propose an unsupervised solution to homography estimation by a new architecture with content-awareness learning. It is specially designed for image pairs with a small baseline, as this case is commonly applicable for consecutive video frames, burst image capturing or photos captured by a dual-camera cellphone. In particular, to robustly optimize homography, our network implicitly learns a deep feature for alignment and a content-aware mask to reject outlier regions simultaneously. The learned feature is used for loss calculation instead of using photometric loss as in [17], and learning the content-aware mask actually acts as a neural RANSAC to mimic the traditional RANSAC procedure. We further formulate a novel triplet loss to optimize the network so that the unsupervised learning could be achieved.

Experimental results demonstrate the effectiveness of all the newly involved techniques for our network. The qualitative and quantitative evaluations also show that our network outperforms the state-of-the-art (SOTA) as shown in Figs. 1, 8 and 9 for homography estimation task. We also introduce a comprehensive image pair dataset,[1] which has been divided into 5 categories of scenes and contains human-labeled GT point correspondences for quantitative evaluations of its validation set (Fig. 7).

In addition, with the proposed unsupervised homography pipeline, we show that it is possible to be extended for mesh-based registrations, which we call Deep Meshflow. On the one hand, our deep mesh-based registration shares the advantages of our deep homography framework. On the other hand, it can deliver flexibilities for alignments of scenes with large depth variations that are beyond the capability of a single homography. We show that our Deep Meshflow solution outperforms the mesh-based state-of-the-arts. To summarize, our main contributions are:

- A novel network structure that enables content-aware robust homography estimation from two images with small baseline.
- A triplet loss designed for unsupervised training, so that an optimal homography could be produced as an output, together with a deep feature map for alignment and a mask highlighting the alignment inliers being implicitly learned as intermediate results.
- A comprehensive dataset covers various scenes for unsupervised training of image alignment models, including but not limited to homography, mesh warps or optical flow.

2 Related Work

Traditional Homography. A homography is a 3 × 3 matrix that compensates plane motions between two images. It consists of 8 ° of freedom (DOF), with each 2 for scale, translation, rotation and perspective [1] respectively. To solve a homography, traditional approaches often detect and match image features, such as classic SIFT [14], SURF [18], ORB [19] and

[1] https://github.com/JirongZhang/DeepHomography

neural network pipeline LIFT [20]. Two sets of correspondences were established between two images, following which robust estimation is adopted, such as the classic RANSAC [16], IRLS [21] and recent MAGSAC [22] that no longer requires a user-defined inlier-outlier threshold, for the outlier rejection during the model estimation.

A homography can also be solved directly without image features. The direct methods, such as the seminal Lucas-Kanade algorithm [23], calculate the sum of squared differences (SSD) between pixels from two images. The differences guide the shift and warp of the images, yielding homography updates. A randomly initialized homography is optimized in this way iteratively [24]. Moreover, the SSD can be replaced with an enhanced correlation coefficient (ECC) for the robustness [25].

Deep Homography. Following the success of various deep image alignment methods such as optical flow [3], [26], dense matching [27], learned descriptors [28] and deep features [29], a deep homography solution was first proposed by [17] in 2016. The network takes source and target images as input and produces 4 corner displacement vectors, so as to generate the homography. It used GT homography to supervise the training. However, the training images with GT homography are generated from a single image without depth disparities. Recently, Le *et al.* [30] proposed another supervised homography network with emphasis on the dynamic contents. It further estimated a dynamic mask during homography regression supervised by GT dynamic masks calculated from optical flows. Similarly, this approach also suffers from the depth disparity problem as it generates GT homography similar to [17]. Note that, as [30] is not open-sourced, in the experiments, we mainly compare with [17] for the supervised homography.

To overcome the limitation of supervised homography, Nguyen *et al.* [15] proposed an unsupervised approach that computed photometric loss between two images and adopted Spatial Transform Network (STN) [31] for image warping. As mentioned above, loss defined on intensity instead of feature space may mislead the alignment evaluation, and the loss is calculated uniformly on the image plane so that the outlier regions contribute the same as the inliers, so as to introduce error when optimizing the homography.

Mesh Warping. To solve the depth parallax issue, mesh-based image warping is often adopted. Liu *et al.* proposed Content Preserving Warp (CPW) to encourage mesh cells to undergo a rigid motion [32]. Li *et al.* proposed a duel-feature warping by considering not only image features but also line segments for the warping in low-textured regions [33]. Lin *et al.* incorporated a curve preserving term to preserve curve structures [34]. Liu *et al.* introduced Meshflow, a non-parametric warping method for video stabilization [2], in which a sparse motion field with motions only located at mesh vertexes was estimated. In this work, we extend our unsupervised deep homography pipeline to support mesh-based registration, namely Deep Meshflow, with largely improved robustness against scenes that suffer from feature detection and matching problems.

Image Stitching. Image stitching methods [8], [35] are mainly traditional methods that focus on stitching images under large baselines [36] for the purpose of constructing the panorama [37]. The stitched images were often captured with dramatic viewpoint differences. In this work, we focus on images with small baselines for the purpose of multi-frame applications.

Weakly-Supervised Semantic Alignment. Our method is also related to weakly-supervised semantic alignment, which estimates dense correspondences between different objects belonging to the same category. SCNet [38] proposed to learn semantic correspondence using object-proposal level labels. WeakAlign [39] proposed to learn deep descriptor and alignment model by maximizing the soft-inlier count of a 4D dense correlation computed from high-level features extracted by ResNet-101 backbone. Following WeakAlign, RTNs [40] estimated geometric field in a recurrent manner based on a constraint 4D correlation volume. Different from these semantic alignment methods, our method is specially designed for low-level image alignment, e.g., homography estimation.

3 Algorithm

3.1 Network Structure

Our method is built upon convolutional neural networks. It takes two grayscale image patches I_a and I_b as input, and produces a homography matrix $\mathcal{H}_{ab}$ from I_a to I_b as output. The entire structure could be divided into three modules: a feature extractor $f(\cdot)$, a mask predictor $m(\cdot)$ and a homography estimator $h(\cdot)$. $f(\cdot)$ and $m(\cdot)$ are fully convolutional networks which accepts input of arbitrary sizes, and the $h(\cdot)$ utilizes a backbone of ResNet-34 [41] and produces 8 values. Fig. 2a illustrates the network structure.

3.1.1 Feature Extractor

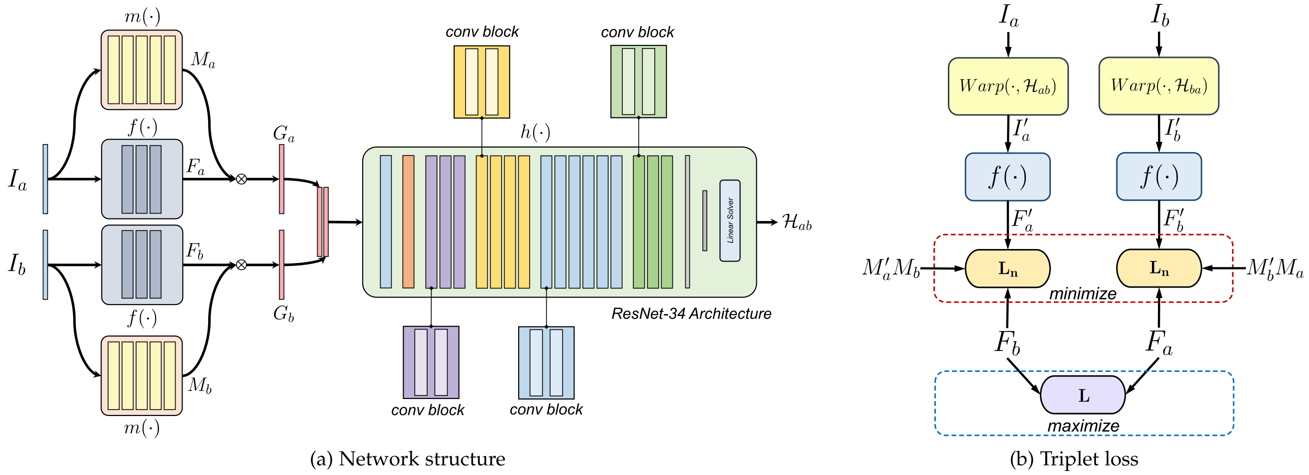
Unlike previous DNN based methods that directly utilize the pixel intensity values as the feature, here, our network automatically learns a deep feature from the input for robust feature alignment. To this end, we build a fully convolutional network (FCN) that takes an input of size $H \times W \times 1$, and produces a feature map of size $H \times W \times C$. For inputs I_a and I_b, the feature extractor shares weights and produces feature maps F_a and F_b, i.e.,

$$F_\beta = f(I_\beta), \quad \beta \in \{a, b\}. \tag{1}$$

The learned feature usually has more robust properties than pixel intensity when applied to loss calculation. Especially for the images with luminance change, the learned feature is pretty robust to overcome it compared with the pixel intensity values. See Section 4.3 and Fig. 3 for a detailed verification of the effectiveness of this module.

3.1.2 Mask Predictor

In non-planar scenes, especially those including moving objects, there exists no single homography that can align the two views. In traditional algorithms, RANSAC is widely applied to find the inliers for homography estimation, so as to solve the most approximate matrix for the scene alignment. Following the similar idea, we propose to build a sub-network to automatically learn the inliers' positions. Specifically, a sub-network $m(\cdot)$ learns to produce an inlier probability map or mask, highlighting the content in the feature maps that contribute much to the homography estimation. The size of the mask is the same as the size of the feature maps



(a) Network structure

(b) Triplet loss

Fig. 2. The overall structure of our deep homography estimation network (a) and the triplet loss we design to train the network (b). In (a), two input patches I_a and I_b are fed into two branches consisting of feature extractor $f(\cdot)$ and mask predictor $m(\cdot)$ respectively, generating features F_a, F_b and masks M_a, M_b. Then the features and masks are fed into a homography estimator to produce 8 values of the homography matrix $\mathcal{H}_{ab}$. In $h(\cdot)$, convolution blocks in various colors differ in the number of channels (detailed in Table 1). To train the network in (a), we design a triplet loss composed of $\mathbf{L_n}, \mathbf{L}$ as defined in Eqs. (4), (5) and (6).

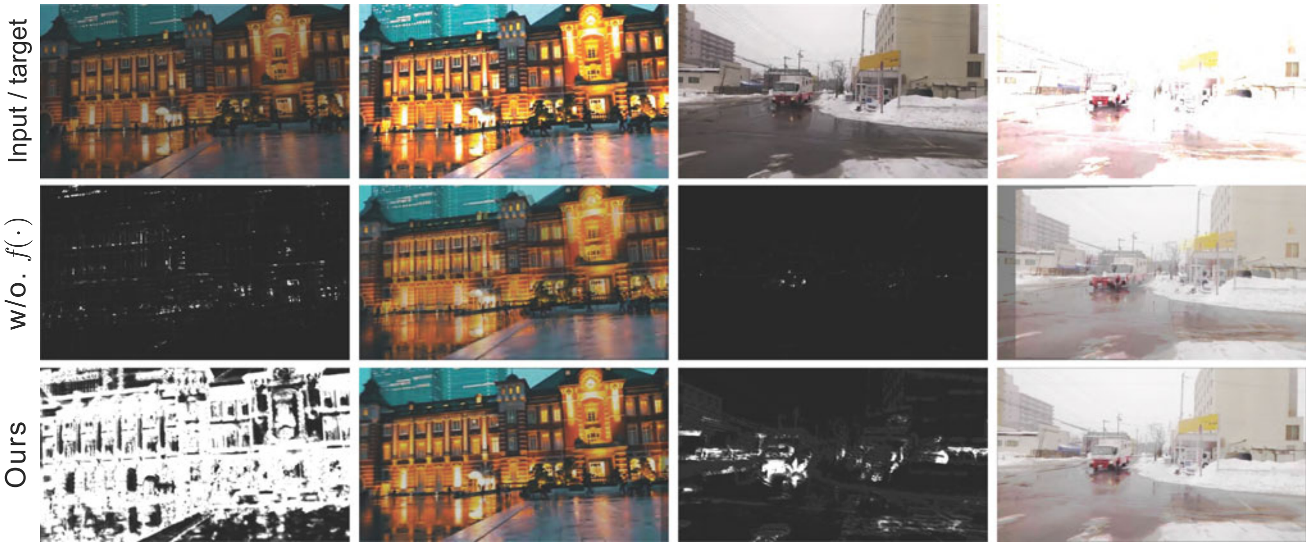


Fig. 3. Ablation study on the effectiveness of our feature extractor, demonstrated by examples with illuminance change, displayed separately in the left and right two columns. For each example, the input and target GT images are in Row 1, followed by the results by disabling the feature extractor $f(\cdot)$ (Row 2) and by ours (Row 3), including the learned masks and the aligned results in odd and even columns. As seen, our results are obviously stable for such a case.

F_a and F_b. With the masks, we further weight the features extracted by f before feeding them to the homography estimator, obtaining two weighted feature maps G_a and G_b as,

$$M_\beta = m(I_\beta), \quad G_\beta = F_\beta M_\beta, \quad \beta \in \{a, b\}. \tag{2}$$

As introduced later, the mask learned as above actually plays two roles in the network: one works as an attention map, and the other works as a RANSAC-like outlier rejection scheme. See the details in Sections 3.2, 4.3 and Fig. 4 for more discussion.

3.1.3 Homography Estimator

Given the weighted feature maps G_a and G_b, we concatenate them to build a feature map $[G_a, G_b]$ of size $H \times W \times 2\,C$. Then it is fed to the homography estimator network and four 2D offset vectors (8 values) are produced. With the 4 offset vectors, it is straightforward to obtain the homography matrix $\mathcal{H}_{ab}$ with 8 DOF by solving a linear system. We use $h(\cdot)$ to represent the whole process, i.e.,

$$\mathcal{H}_{ab} = h([G_a, G_b]). \tag{3}$$

The backbone of $h(\cdot)$ follows a ResNet-34 structure. It contains 34 layers of strided convolutions followed by a global average pooling layer, which generates fixed size (8 in our case) of feature vectors regardless of the input feature dimensions.

We list the layer details of the three modules above in Table 1.

3.2 Triplet Loss for Robust Homography Estimation

With the homography matrix $\mathcal{H}_{ab}$ estimated, we warp image I_a to I'_a and then further extracts its feature map as F'_a.

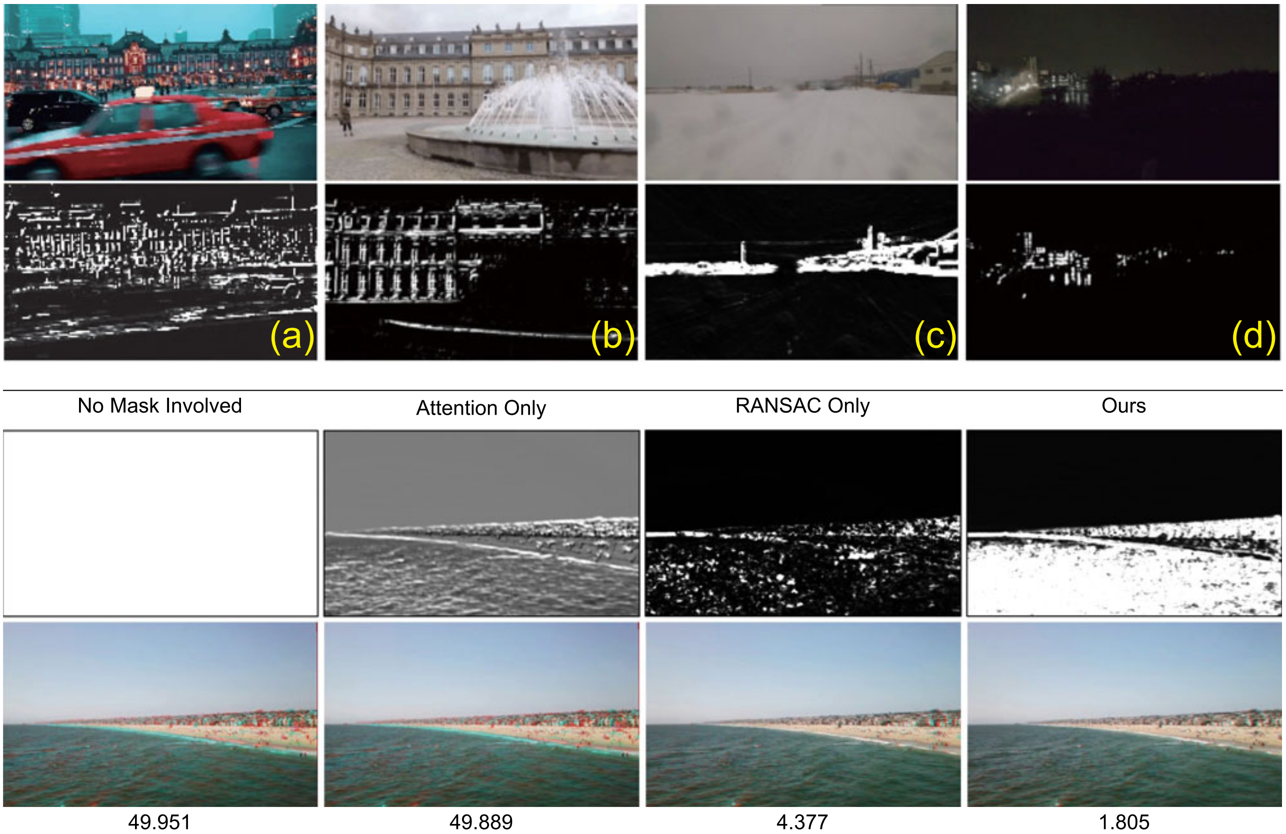


Fig. 4. Row 1 and 2: Our predicted masks for various of scenes. (a) and (b) contains large dynamic foreground. (c) contains few textures and (d) is an night example. Row 3 and 4: Ablation study on the content-aware mask. We disable both or either role of the mask to compare with ours. Errors are shown at the bottom for all cases.

TABLE 1
Layer Configurations of Feature Extractor (a), Mask Predictor (b) and Homography Estimator (c)

(a) Feature extractor $f(\cdot)$

Layer No. Type	1 *conv*	2 *conv*	3 *conv*
Kernel	3	3	3
Stride	1	1	1
Channel	4	8	1

(b) Mask predictor $m(\cdot)$

Layer No. Type	1 *conv*	2 *conv*	3 *conv*	4 *conv*	5 *conv*
Kernel	3	3	3	3	3
Stride	1	1	1	1	1
Channel	4	8	16	32	1

(c) Homography estimator $h(\cdot)$

Layer No. Type	1 *conv*	2 *pool*	$3\sim8$ *conv*	9 *conv*	$10\sim16$ *conv*	17 *conv*	$18\sim28$ *conv*	29 *conv*	$30\sim34$ *conv*	35 *pool*	36 *fc*
Kernel	7	3	3	3	3	3	3	3	3	-	-
Stride	2	2	1	2	1	2	1	2	1	1	-
Channel	64	-	64	128	128	256	256	512	512	-	8

In (c), Layer 2 and 35 are max pool and global average pool separately.

Intuitively, if the homography matrix $\mathcal{H}_{ab}$ is accurate enough, F'_a should be well aligned with F_b, causing a low l_1 loss between them. Considering the real scenes where a single homography matrix normally cannot satisfy the transformation between the two views, we also normalize the l_1 loss by M'_a and M_b. Here M'_a is the warped version of M_a. So the loss between the warped I_a and I_b is as follows,

$$\mathbf{L_n}(I'_a, I_b) = \frac{\sum_i M'_a M_b \cdot ||F'_a - F_b||_1}{\sum_i M'_a M_b}, \tag{4}$$

where $F'_a = f(I'_a), \quad I'_a = Warp(I_a, \mathcal{H}_{ab})$ and i indicates a pixel location in the masks and feature maps. Here we utilize STN [31] to achieve the warping operation. In order to avoid the situation when the masks are learned to be all zeros, we place the sum of the masks in the denominator as an implicit normalization term. In this way, the loss value may be larger when there are more zeros in the masks. Therefore, our network tends to predict an inlier mask for the dominant plane with the largest area so that this normalization term can be minimized. As shown in Figs. 4a and 4b, the background objects are marked as the dominant plane by our inlier mask. On the contrary, in Fig. 5, the foreground is chosen as the dominant plane by our network to be well aligned while the background is treated as outliers to be ignored, because the foreground area is larger than the background.

Note that in Eq. (5), we assume that there is a dominant plane in the image that is misaligned and needs to be aligned

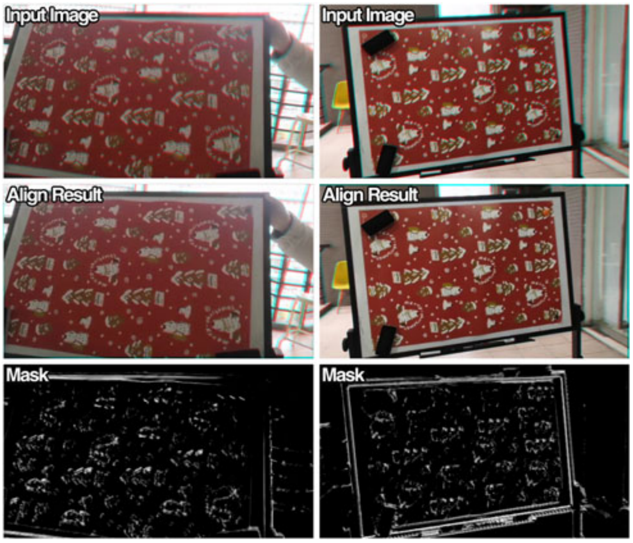


Fig. 5. Examples where the foreground object is the dominant plane in an image. The network treats the foreground dominant plane as the inlier while backgrounds as the outliers. We show the overlapped input images in Row 1, the aligned images in Row 2, and the inlier masks in Row 3.

by our network. We do not consider the situation where the dominant plane in the input image is already aligned because in most of the cases, the input of a homography estimator is a pair of misaligned images. Under this condition, we define Eq. (4) as the main objective term to encourage our network to align two images in the feature space. However, directly minimizing Eq. (4) may easily cause trivial solutions, where the feature extractor only produces all zero maps, i.e., $F'_a = F_b = 0$. In this case, the features learned indeed describe the fact that I'_a and I_b are "well aligned," but it fails to reflect the fact that the original images I_a and I_b are mis-aligned. To this end, we involve another loss between F_a and F_b, i.e.,

$$\mathbf{L}(I_a, I_b) = ||F_a - F_b||_1, \tag{5}$$

and further maximize it when minimizing Eq. (4). This strategy avoids the trivial all-zero solutions, and enables the network to learn a discriminative feature map for image alignment. Although the regularization in Eq. (5) may fail in some sub regions which are originally aligned in the input image pair, it performs well to help our network to be optimized globally for predicting a homography matrix to align the dominant plane in the image. As discussed later in our ablation study in Section 4.3, our triple loss can improve the performance by at least 50% lower error on average.

In practise, we swap the features of I_a and I_b and produce another homography matrix $\mathcal{H}_{ba}$. Following Eq. (4) we involve a loss $\mathbf{L_n}(I'_b, I_a)$ between the warped I_b and I_a. We also add a constraint that enforces $\mathcal{H}_{ab}$ and $\mathcal{H}_{ba}$ to be inverse. So, the optimization procedure of the network could be written as follows,

$$\min_{m,f,h} \mathbf{L_n}(I'_a, I_b) + \mathbf{L_n}(I'_b, I_a) - \lambda \mathbf{L}(I_a, I_b) + \mu ||\mathcal{H}_{ab}\mathcal{H}_{ba} - \mathcal{I}||_2^2, \tag{6}$$

where λ and μ are balancing hyper-parameters, and $\mathcal{I}$ is a 3-order identity matrix. We set $\lambda = 2.0$ and $\mu = 0.01$ in our experiments. We illustrates the loss formulations in Fig. 2b.

3.3 Unsupervised Content-Awareness Learning

As mentioned above, our network contains a sub-network $m(\cdot)$ to predict an inlier probability map or mask. It is designed that our network can be of content-awareness by the two-fold roles. First, we use the masks M_a, M_b to explicitly weight the features F_a, F_b, so that only highlighted features could be fully fed into homography estimator $h(\cdot)$. The masks actually serve as attention maps for the feature maps. Second, they are also implicitly involved in the normalized loss Eq. (4), working as a weighting item. By doing this, only those regions that are really fit for alignment would be taken into account, just like RANSAC. For those areas containing low texture or dynamic foreground, because they are non-distinguishable or misleading for alignment, they are naturally removed for homography estimation during optimizing the triplet loss as proposed. Such a content-awareness is achieved fully by an unsupervised learning scheme, without any GT mask data as supervision. To demonstrate the effectiveness of the mask as the two roles, we conduct an ablation study by disabling the effect of the mask working as an attention map or as a loss weighting item. As seen in Table 5, the accuracy has a significant decrease when the mask is removed in either case.

We also illustrate several examples in Fig. 4 to show the mask effectiveness. For example, in Figs. 4a and 4b where the scenes contain dynamic objects, our network successfully rejects moving objects, even if the movements are inapparent as the fountain in (b), or the objects occupy a large space as in (a). These cases are very difficult for RANSAC to find robust inliers. Fig. 4c is a low-textured example, in which the blue sky occupies half-space of the image. It is challenging for traditional methods because the sky provides no enough features. Our predicted mask concentrates on the horizon and takes advantage of the texture in the sea waves. Last, Fig. 4d is a low light example, where only visible areas contain weights as seen. We also illustrate an example to show the two effects by the mask as separate roles in the bottom 2 rows of Fig. 4. Details about this ablation study are introduced later in Section 4.3.

We adopt a two-stage strategy to train our network. Specifically, we first train the network by disabling the attention map role of the mask, i.e., $G_\beta = F_\beta,\ \beta \in \{a, b\}$. After about 60 k iterations, we finetune the network by involving the attention map role of the mask as Eq. (2). We validate this training strategy by another ablation study detailed in Section 4.3, where we train the network totally from scratch. This two-stage training strategy reduces the error by 4.40% on average, as shown in Row 11 of Table 5.

3.4 Generalizing to MeshFlow

In practice, a single homograph is usually unable to align the entire scene as there commonly exists more than one plane to fit. A workaround is to use a mesh to warp the scene so that each of the local mesh grids is subject to one homography, such as MeshFlow [2]. For our network, we can also make appropriate adjustments to support this kind of output.

Fig. 6 shows the adjusted network structure for the multiple homography output as stated above. We remove the linear solver after the last layer of the homography estimator $h(\cdot)$, and divide it into K branches, each of which connects a

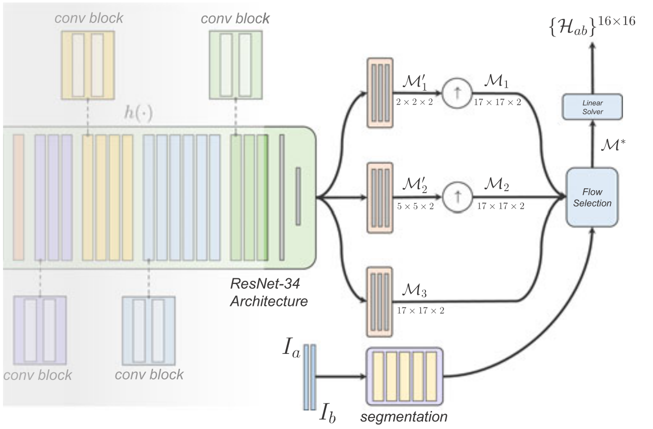


Fig. 6. The network structure being generalized to MeshFlow, namely Deep MeshFlow. The hidden part of the network is the same as the one in Fig. 2.

3-layer convolutional block to produce a tensor $\mathcal{M}'_k$ of size $(H_{g,k}+1)\times(W_{g,k}+1)\times 2, k=1,2,\ldots,K,$ where $H_{g,k}=H_{g,K}\cdot 2^{k-K}, W_{g,k}=W_{g,K}\cdot 2^{k-K}$. Here the tensor $\mathcal{M}'_k$ is actually a mesh with $(H_{g,k}+1)\times(W_{g,k}+1)$ vertices, and each vertex is associated with a 2D displacement vector. Then, the meshes except the finest one $\mathcal{M}'_K$, are upsampled to the same size of $\mathcal{M}'_K$, in order to be further fused together. As $\mathcal{M}'_K$ needs no upsampling, we also write it as $\mathcal{M}_K$ for convenience.

As for the fusion, we first train a simple segmentation network s to produce a K-class segmentation map S of size $(H_{g,K}+1)\times(W_{g,K}+1)\times K$ as follows,

$$S = s(I_a, I_b), \tag{7}$$

and then produce the final mesh output $\mathcal{M}^*$ in the manner as below,

$$\mathcal{M}^*(u,v) = \mathcal{M}_{\tilde{k}}(u,v), \tag{8}$$

where $\tilde{k} = \arg\max_k S(u,v,k)$ and (u,v) is the vertex coordinate on the mesh. By this strategy, the output mesh flow conveys homography alignment in various scales for each local grid. It has enough DoF to align the two views and is still easy for training.

We set $H_{g,K}=W_{g,K}=16$ and $K=3$, so the sizes of meshes $\{\mathcal{M}'_k\}$ are $2\times2\times2$, $5\times5\times2$ and $17\times17\times2$. The final output mesh $\mathcal{M}^*$ is also of size $17\times17\times2$. Using the displacement information in $\mathcal{M}^*$, we can obtain 16×16 homography $\{\mathcal{H}_{ab}\}^{16\times16}$ by a linear solver as Fig. 6 shows.

As we get the mesh flow $\mathcal{M}^*$ with a group of homography matrices $\{\mathcal{H}_{ab}\}^{16\times16}$, we achieve warping with finer control. It means the warped I'_a in Eq. (4) is calculated with $I'_a = Warp(I_a, \mathcal{M}^*)$ instead. Meanwhile, the triplet loss is also slightly modified from Eq. (6) to as follows,

$$\begin{aligned}\min_{m,f,h} \mathbf{L_n}(I'_a, I_b) + \mathbf{L_n}(I'_b, I_a) - \lambda \mathbf{L}(I_a, I_b)\\ +\mu \sum_{(u,v)} ||\mathcal{H}_{ab}(u,v)\mathcal{H}_{ba}(u,v) - \mathcal{I}||_2^2.\end{aligned} \tag{9}$$

We call this adjusted version of network "Deep MeshFlow," which is supposed to be of stronger capability for alignment compared with a single homography. Note that, as this is the extension of our Deep Homography baseline, we will first reveal the Deep Homography over the previous methods, then present some comparisons regarding Deep MeshFlow. However, the emphasis will be given to the Deep Homography.

4 Experimental Results

4.1 Dataset and Implementation Details

Dataset. Previously, there is no dedicated dataset designed to evaluate the performance of homography fitting. The supervised method [17] synthesized homographies from a single image, so it cannot reflect disparities and occlusions. The unsupervised method [15] adopted aerial images that lack the generalization ability. Therefore, we propose our dataset for comprehensive evaluations.

Our dataset contains 5 categories, including regular (**RE**), low-texture (**LT**), low-light (**LL**), small-foregrounds (**SF**), and large-foregrounds (**LF**) image pairs. Each category contains around $16\ k$ image pairs, thus totally $80\ k$ image pairs in the dataset, as shown in Fig. 7. The category partition is based on the understanding of traditional homography registration. For regular examples (Fig. 7 RE), image features can be extracted easily due to rich textures and the scene is flat which is friendly for homography estimation. For low-texture and low-light examples (Fig. 7 LT, LL), only a few image features could be extracted, which causes troubles for traditional homography fitting. For scenes containing foreground or contain dynamic objects (Fig. 7 SF), the scene structure is no longer a plane. In such cases, a best fitting homography would align the most dominant planar structure of the scene, while other non-planar objects are excluded. This can be achieved by RANSAC outlier rejection for traditional methods, but may cause troubles for the previous two deep methods [15], [17] which treat the image content equally. The most challenging case is the scene with large foreground (Fig. 7 LF), for which even the RANSAC cannot handle it easily. Experimental results demonstrate our method is robust overall categories as seen in Figs. 1, 8, 9 and the supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3174130.

With respect to the testing data, we randomly choose $4.2\ k$ image pairs from all categories. For each pair, we manually marked $6\sim10$ equally distributed matching points for the purpose of quantitative comparisons, as illustrated in Fig. 7. Note that these labeled point pairs may locate in either the dominant plane or non-dominant plane in the image. For the homography estimation evaluation, we manually select 6 point pairs in the dominating plane, and for the mesh-based registration evaluation, we employ all of the labeled point pairs.

Implementation Details. Our network is trained with $120\ k$ iterations by an Adam optimizer [42], with parameters being set as $l_r = 1.0\times10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1.0\times10^{-8}$. The batch size is 64, and for every $12\ k$ iterations, the learning rate l_r is reduced by 20%. Each iteration costs about 1.2 s and it takes nearly 40 hours to complete the entire training. The implementation is based on PyTorch and the network training is performed on 4 NVIDIA RTX 2080 Ti. To augment the training data and avoid black boundaries appearing in the warped image, we randomly

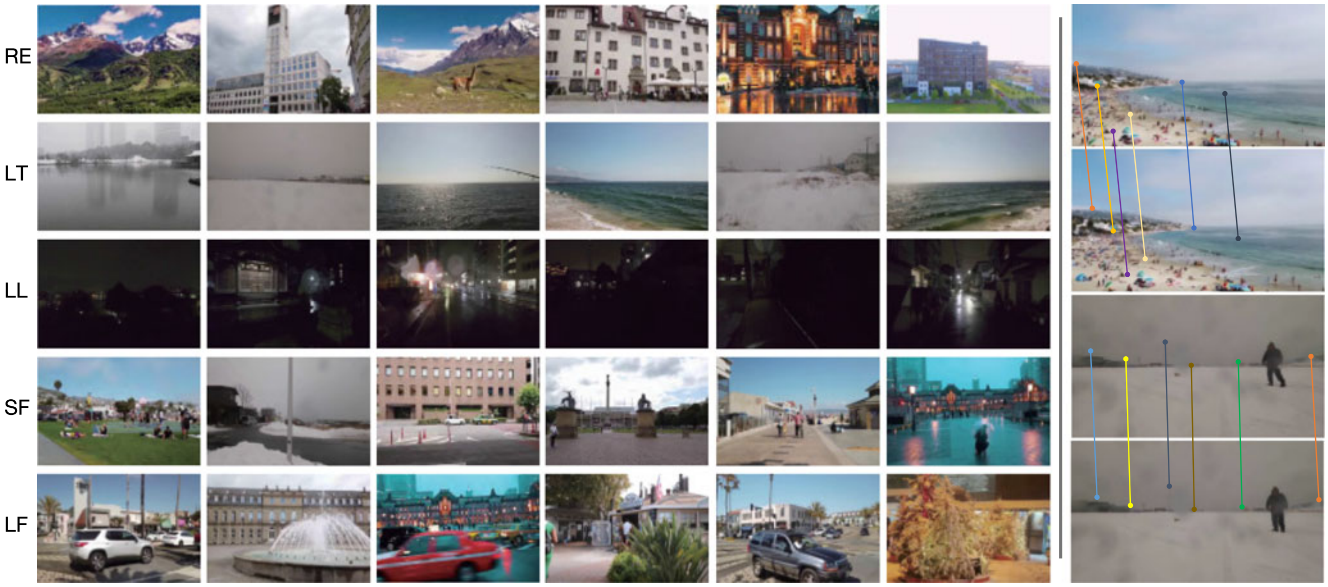


Fig. 7. A glance of our dataset. For left 6 columns, from top to bottom: regular (RE), low texture (LT), low light (LL) examples, examples of small foreground (SF) and large foreground (LF). The rightmost column shows two examples of human labeled point correspondences for quantitative evaluation.

crop patches of size 315×560 from the original image to form I_a and I_b. Code will be released upon the acceptance of this paper.

Evaluation Metric. For quantitative comparison on our test dataset, we use the average point distance error as the evaluation metric. For each test sample, we first use the estimated homography matrix to warp the labeled points in I_a and then compute an average distance error between the warped points and the labeled points in I_b

$$error = \frac{1}{N}\sum_{i}^{N} ||\boldsymbol{p}_b^i - Warp(\boldsymbol{p}_a^i, \mathcal{H}_{ab})||_2, \tag{10}$$

where N is the number of the labeled points for the test sample, i is the index of the labeled point pair. Note that we use the labeled point pairs on the dominant plane to evaluate the performance of homography estimation. For mesh-based registration, which can align multiple planes in the image, we use all the labeled point pairs for evaluations.

4.2 Comparisons With Existing Methods

4.2.1 Qualitative Comparison

We first compare our method with the existing two deep homography ones, i.e., the supervised [17] and the unsupervised [15] approaches, as illustrated in Fig. 8.

Fig. 8a shows an synthesized example with no disparities. In this case, the supervised solution [17] performs well enough as ours. However, it fails in the case that real consecutive frames of the same footage are applied (Fig. 8b), because it is unable to handle large disparities and moving objects of the scene. Fig. 8c shows an example that contains a dominant planar building surface, where all methods work well. However, if the image pair involves illumination variation caused by camera flash, the unsupervised method [15] fails due to its alignment metric being pixel intensity value difference instead of semantic feature difference, as seen in Fig. 8d. Figs. 8e and 8f contain near-range objects and two dominant planes with moving objects at corners respectively, and Figs. 8g and 8h are low texture and low light examples separately. Similarly, in all of these scenarios, our method produces warped images with more pixels aligned, so as to obviously outperform the other two DNN-based ones.

We also compare our method with some feature-based solutions. Specially, we choose SIFT [14], ORB [19], LIFT [20] and SOSNet [28] as the feature descriptors and choose RANSAC [16] and MAGSAC [22] as the outlier rejection algorithms, obtaining 8 combinations. We show 3 examples in Fig. 9, where (a) and (b) show the 8 combinations produce reasonable but low quality results, and (c) shows one that most of them fail thoroughly. Note that this kind of failure case caused by low texture or low light condition frequently exists for our dataset, and it may lead to unstable results in real applications such as multi-frame image fusion. In comparison, our method is robust enough for these challenging cases.

4.2.2 Quantitative Comparison

We further demonstrate the performance of our method by comparing it with all of the other methods quantitatively. The comparison is based on our dataset and the average l_2 distances between the warped points and the human-labeled GT points are evaluated as the error metric. We report the errors for each category and the overall averaged error in Table 2, where $\mathcal{I}_{3\times3}$ refers to a 3×3 identity matrix as a "no-warping" homography for reference. As seen, our method outperforms the others for all categories, except for regular (RE) scenes if compared with feature-based methods. This result is reasonable because in RE scenes, rich texture delivers sufficient high-quality features so that it is naturally friendly for the feature-based solutions. Even though, our error is only 5.85% higher than the best solution in this case, i.e., SIFT [14] + MAGSAC [22]. For the rest scenes, our method consistently beats the others, especially for the low texture (LT) and low light (LL) scenes, where our error is

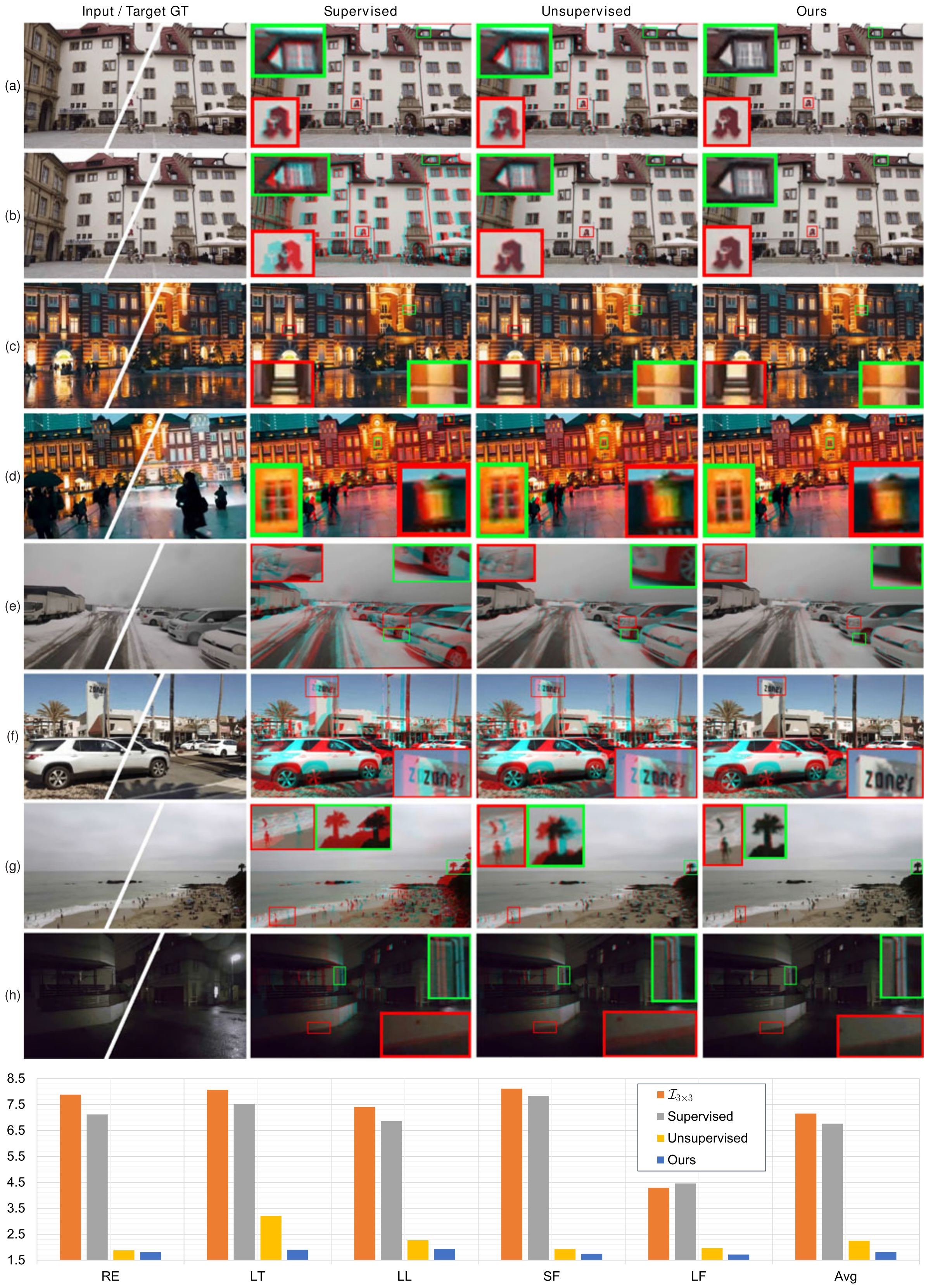


Fig. 8. Comparison with existing DNN-based approaches. Column 1 shows the input and GT target images, columns 2 to 4 are results by the supervised [17], the unsupervised [15] and our method. The errors by all the DNN-based methods are displayed by a bar chart at the bottom.

lower than the 2nd best by 25.78% and 7.62% respectively. For the scenes containing small (SF) and large (LF) foreground, although the 2nd best method SOSNet [28] + MAGSAC [22] only loses to ours very slightly (0.57% and 2.82%), it cannot well handle the LT and LL scenes, where its errors are higher than the 2nd best by 100.78% and 109.05% separately. It is worth noting that the two solutions involving LIFT [20] feature produce rather stable results for all scenes, but their

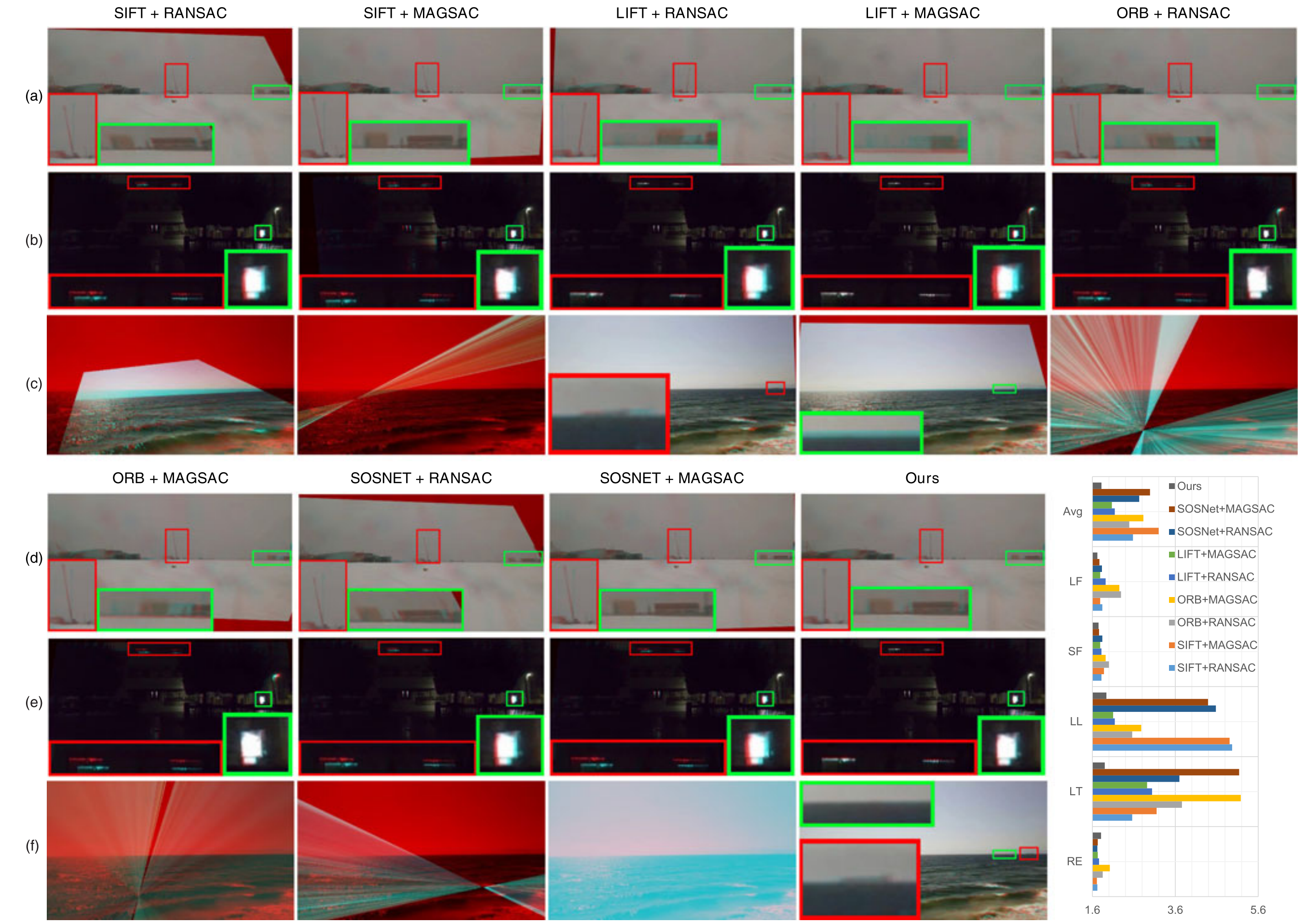


Fig. 9. Comparison with 8 feature-based solutions on 3 examples, shown in (a)(d), (b)(e) and (c)(f). For the first 2 examples, our method produces more accurate results, while for the last one but not the least, most of the feature-based solutions fail extremely, which is a frequent phenomenon for the low texture or low light scenes. We also display the errors by all the methods in the bar chart at the bottom right corner.

TABLE 2
Quantitative Comparison Between Ours and all Other Methods Including DNN-Based (Row 3, 4) and Feature-Based (Row $5 \sim 12$) ones

1)		RE	LT	LL	SF	LF	Avg
2)	$\mathcal{I}_{3\times 3}$	7.88 (+360.82%)	8.07 (+215.23%)	7.41 (+252.86%)	8.11 (+360.80%)	4.29 (+142.37%)	7.15 (+245.41%)
3)	Supervised [17]	7.12 (+316.37%)	7.53 (+194.14%)	6.86 (+226.67%)	7.83 (+344.89%)	4.46 (+151.98%)	6.76 (+226.57%)
4)	Unsupervised [15]	1.88 (+9.94%)	3.21 (+25.39%)	2.27 (+8.10%)	1.93 (+9.66%)	1.97 (+11.30%)	2.25 (+8.70%)
5)	SIFT [14] + RANSAC [16]	1.72 (+0.58%)	2.56 (+0.00%)	4.97 (+136.67%)	1.82 (+3.41%)	1.84 (+3.95%)	2.58 (+24.64%)
6)	SIFT [14] + MAGSAC [22]	**1.71 (+0.00%)**	3.15 (+23.05%)	4.91 (+133.81%)	1.88 (+6.82%)	1.79 (+1.13%)	3.20 (+54.59%)
7)	ORB [19] + RANSAC [16]	1.85 (+8.19%)	3.76 (+46.88%)	2.56 (+21.90%)	2.00 (+13.64%)	2.29 (+29.38%)	2.49 (+20.29%)
8)	ORB [19] + MAGSAC [22]	2.02 (+18.13%)	5.18 (+102.34%)	2.78 (+32.38%)	1.92 (+9.09%)	2.25 (+27.12%)	2.83 (+36.71%)
9)	LIFT [20] + RANSAC [16]	1.76 (+2.92%)	3.04 (+18.75%)	2.14 (+1.90%)	1.82 (+3.41%)	1.92 (+8.47%)	2.14 (+3.38%)
10)	LIFT [20] + MAGSAC [22]	1.73 (+1.17%)	2.92 (+14.06%)	2.10 (+0.00%)	1.79 (+1.70%)	1.79 (+1.13%)	2.07 (+0.00%)
11)	SOSNet [28] + RANSAC [16]	1.72 (+0.58%)	3.70 (+44.53%)	4.58 (+118.09%)	1.84 (+4.54%)	1.83 (+3.39%)	2.73 (+31.88%)
12)	SOSNet [28] + MAGSAC [22]	1.73 (+1.17%)	5.14 (+100.78%)	4.39 (+109.05%)	1.76 (+0.00%)	1.77 (+0.00%)	2.99 (+44.44%)
13)	Ours	**1.81 (+5.85%)**	**1.90 (-25.78%)**	**1.94 (-7.62%)**	**1.75 (-0.57%)**	**1.72 (-2.82%)**	**1.82 (-12.08%)**

For each scene category, we mark the best solution in bold and red. For the scenes ours beats all the others, we mark the 2nd best solution in blue.

average errors are higher than ours by at least 12.08%. As for the DNN-based solutions, the supervised method [17] suffers severely from the generalization problem as demonstrated by its errors being higher than us by at least 142.37% for all scenes, and the unsupervised method [15] also apparently fails in the LT scene, causing over 50% higher error than ours in this case. Please see Table 2 and the bar charts in Figs. 8 and 9 for the detailed quantitative comparisons.

We also compare our method with other methods on the HPatches benchmark [48]. Since our method is limited to be applied to large baseline scenes (discussed later in Section 4.6), We only evaluate our method on the illumination change scenes in HPatches benchmark. We train the proposed network on our training data set and images from HPatches benchmark are kept unseen by our network. The experiment results are shown in Table 3. The evaluation metric is the percentage of the estimated homographies whose average corner error distance is less than $\epsilon = 1, 3, 5$ pixels. The supervision type and run time of each method are also reported in Table 3. As can be seen, our method is

TABLE 3
Quantitative Comparison Between Our Method and Other Methods on the Illumination Change Scenes in HPatches Benchmark [48]

	Method	Accuracy			Sup	Time (s)
		$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$		
1)	SuperPoint [43] + NN [44]	0.57	0.92	0.97	Full	0.12
2)	D2Net [45] + NN	**0.65**	**0.95**	**0.98**	Full	1.61
3)	R2D2 [46] + NN	0.63	0.93	**0.98**	Full	2.34
4)	SuperPoint + CAPS [47] + NN	0.62	0.93	**0.98**	Full	0.36
5)	SIFT+CAPS [47]+NN	0.48	0.89	0.95	Weak	0.73
6)	SIFT + RANSAC	0.45	0.64	0.68	–	0.43
7)	Ours	0.56	0.92	0.95	Unsup	**0.03**

The evaluation metric is the percentage of correctly estimated homographies with average corner error distance that belows to $\epsilon = 1, 3, 5$ pixels. The last two column are the supervision type and the running time of each method.

TABLE 4
Quantitative Comparison Between Our Method and Semantic Alignment Methods Including RTNs [40] (Row 2) and WeakAlign [39] (Row 3)

1)		RE	LT	LL	SF	LF	Avg
2)	RTNs [40]	2.99	3.73	3.19	2.98	2.97	3.17
3)	WeakAlign [39]	6.84	10.15	6.64	7.53	8.49	7.93
4)	WeakAlign [39]-finetune	9.33	11.85	7.95	9.00	11.61	9.95
5)	Ours - Mask as WeakAlign	2.52	3.61	3.01	3.57	2.08	2.96
6)	Ours	**1.81**	**1.90**	**1.94**	**1.75**	**1.72**	**1.82**

In Row 4, we finetuned WeakAlign under our dataset for the fairness. In Row 5, we train our network using the inlier masks generation method as in WeakAlign.

comparable with the full-supervised methods and even outperforms the weakly supervised method. Meanwhile, the inference speed of our method is faster than other methods because our method is an end-to-end deep network without explicit feature detection and matching.

In Table 4, we compare our method with semantic alignment methods such as AWeakAlign [39] and RTNs [40]. As a result, our method outperforms these semantic alignment methods by a relatively large margin. This is because these semantic alignment methods are mainly designed to align two images at the semantic level by learning high-level descriptors, which is not suitable for low-level image alignment. In Table 4 Row 4, we also finetune WeakAlign under our training data set. The result is even worse after finetuning, which also demonstrates that the high-level alignment method is not suitable for low-level image alignment task. In Table 4 Row 5, We train our network using the inlier masks computed by WeakAlign, in which the inlier masks are calculated by comparing the distance between the warped target coordinate and the source coordinate. Comparing Row 5 and 6 in Table 4, we can find that our learned inlier masks perform better than the inlier masks produced by WeakAlign. A possible reason is that adjacent pixels in an image are generally similar, especially for flat areas, the inlier mask of WeakAlign may lead to local optimum in our learning process.

4.3 Ablation Studies

Context-Aware Mask. As mentioned in Section 3.3, the content-aware mask takes effects in two folds, working as an attention for the feature map, or as a weighting map to reject the outliers like RANSAC. We verify its effectiveness by evaluating the performance in the case of disabling both or either effect and report the errors in Row 2, 3, 4 of Table 5. Specifically, for Row 3 "Mask as attention only" Eq. (4) is modified as $\mathbf{L_n}(I'_a, I_b) = \mathbf{L}(I'_a, I_b) = ||F'_a - F_b||_1$. On the contrary, for Row 4 "Mask as RANSAC only" Eq. (2) is modified as $G_\beta = F_\beta,\ \beta \in \{a, b\}$. As the errors indicate, for most scenes the mask takes effect increasingly by the two roles, except for the scenes LT and LF where disabling one role only may cause the worst result. We also illustrate one example in Row 3, 4 of Fig. 4, where in the case of "Mask as attention only" the mask learns to highlight the most attractive edges or texture regions without rejecting the other regions (Column 2). On the contrary, in the case of "Mask as RANSAC only," the mask learns to highlight only sparse texture regions (Column 3) as inliers for alignment. In contrast, our method balances the two effects and learns a comprehensive and informative weighting map, as shown in Column 4. In Table 5 Row 5, we also train our network with an explicit term to regularize the mask not to be all zeros. Compared with the implicit normalization term in Eq. (4), this explicit term is modified as $\mathbf{L_n}(I'_a, I_b) = \sum_i M'_a M_b \cdot ||F'_a - F_b||_1 + e^{-\sum_i M'_a M_b}$. As a result, our implicit regularization term performs better than the explicit term.

Feature Extractor. We also disable the feature extractor to verify its effectiveness. In this experiment, we set $F_\beta = I_\beta,\ \beta \in \{a, b\}$ so that the loss is evaluated on pixel intensity values instead. In this case, the network loses some robustness, especially if applied to images with luminance change, as Fig. 3 shows. As seen, if the feature extractor is disabled, the masks would be abnormally sparse because

TABLE 5
Ablation Studies on Mask (Rows 2 $\sim$ 5), Triplet Loss (Row 6), Feature Extractor (Row 7), Backbones (Rows $8 \sim 10$) and Training Strategy (Row 11)

1)		RE	LT	LL	SF	LF	Avg
2)	No mask involved	2.10 (+16.02%)	2.51 (+32.11%)	2.48 (+27.84%)	3.02 (+72.57%)	1.78 (+3.49%)	2.38 (+30.77%)
3)	Mask as attention only	1.85 (+2.21%)	3.37 (+77.37%)	2.16 (+11.34%)	2.29 (+30.86%)	1.75 (+1.74%)	2.27 (+24.73%)
4)	Mask as RANSAC only	1.85 (+2.21%)	2.16 (+13.68%)	2.17 (+11.86%)	2.04 (+16.57%)	2.16 (+25.58%)	2.07 (+13.74%)
5)	Mask explicit term	1.82 (+0.55%)	2.00 (+5.26%)	1.98 (+2.06%)	1.77 (+1.72%)	1.74 (+1.16%)	1.86 (+2.20%)
6)	w/o. Triple loss	2.16 (+19.34%)	4.15 (+118.42%)	3.30 (+70.10%)	2.49 (+42.29%)	2.09 (+21.51%)	2.84 (+56.04%)
7)	w/o. Feature extractor	1.89 (+4.42%)	2.54 (+33.68%)	2.13 (+9.79%)	1.80 (+2.86%)	1.79 (+4.07%)	2.03 (+11.54%)
8)	VGG [49]	1.91 (+5.52%)	2.89 (+52.11%)	2.05 (+5.67%)	2.14 (+22.29%)	1.88 (+9.30%)	2.17 (+19.23%)
8)	ResNet-18 [41]	1.84 (+1.66%)	2.30 (+21.05%)	2.05 (+5.67%)	2.28 (+30.29%)	1.85 (+7.56%)	2.06 (+13.19%)
10)	ShuffleNet-v2 [50]	2.05 (+13.26%)	2.85 (+50.00%)	2.61 (+34.54%)	2.72 (+55.43%)	1.99 (+15.70%)	2.44 (+34.07%)
11)	Train from scratch	1.87 (+3.31%)	2.00 (+5.26%)	1.98 (+2.06%)	1.90 (+8.57%)	1.77 (+2.91%)	1.90 (+4.40%)
12)	Ours	**1.81**	**1.90**	**1.94**	**1.75**	**1.72**	**1.82**

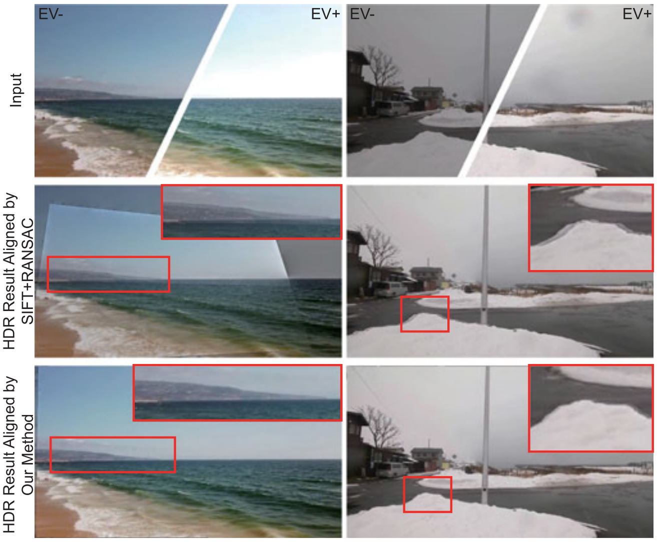

EV-
EV+
EV-
EV+
Input
HDR Result Aligned by SIFT+RANSAC
HDR Result Aligned by Our Method

Fig. 10. Results of HDR imaging from Exposure Fusion. Different exposures should be well aligned before the HDR fusion. The second row shows the fusion results of SIFT+RANSAC while the third row shows the results aligned by our Deep Homography.

the information from the loss reflects only a small falsely "aligned" region. Resultantly, the homography fails to be estimated by misleading information. In comparison, our results are stable enough thanks to the luminance invariant property of the learned feature. The errors are listed in Row 7 of Table 5.

Triplet Loss. We further exam the effectiveness of our triplet loss by removing the term of Eq. (5) from Eq. (6). As shown in Table 5 "w/o. triplet loss," the triplet loss brings us over 50% lower error, especially is so beneficial in LT (118.42% lower error) and LL (70.10% lower error) scenes, demonstrating that it not only avoids the problem of obtaining trivial solutions, but also facilitates a better optimization.

Backbone. We also exam several popular backbones, including VGG [49], ResNet-18 [41], ResNet-34 [41], and ShuffleNet [50] for the homography estimator $h(\cdot)$. As seen in Rows $8 \sim 10$ of Table 5, the ResNet-18 achieves similar performance as ours obtained by ResNet-34. The VGG backbone is slightly worse than ResNet-18 and ResNet-34. Interestingly, the light-weight backbone ShuffleNet-v2 achieves the performance on par with other large backbones, indicating the potential wide application to portable systems of our method.

Training Strategy. As aforementioned, we use a two-stage strategy to train the network. To validate this strategy, we conduct an ablation study here to train the network from scratch. As Row 11 and 12 of Table 5 reveal, our training strategy brings a 4.40% lower error on average, demonstrating its usefulness.

4.4 Applications to HDR Imaging

The HDR imaging technique usually requires the camera to capture several LDR images under various exposures, and then fuses them together to generate an image with higher dynamic range. A critical part of it is how to well align the multiple LDR images with a solid homography to avoid ghosting and reduce noise.

To this end, we embed our Deep Homography calculation step into the pipeline of Exposure Fusion [51], which requires the input images to be pre-aligned before running the core HDR fusion steps. If the input images involve an obvious movement, the alignment quality will directly determine whether severe ghosting exists in the final output. We compare our method with the classical SIFT + RANSAC on two low-textured examples shown in Fig. 10. The first row shows two different exposures, EV+ and EV-, of the two examples. The results of SIFT + RANSAC are shown in the second row of Fig. 10, which suffer from obvious ghosting effects due to the failure of alignments. Our results are free from such problems.

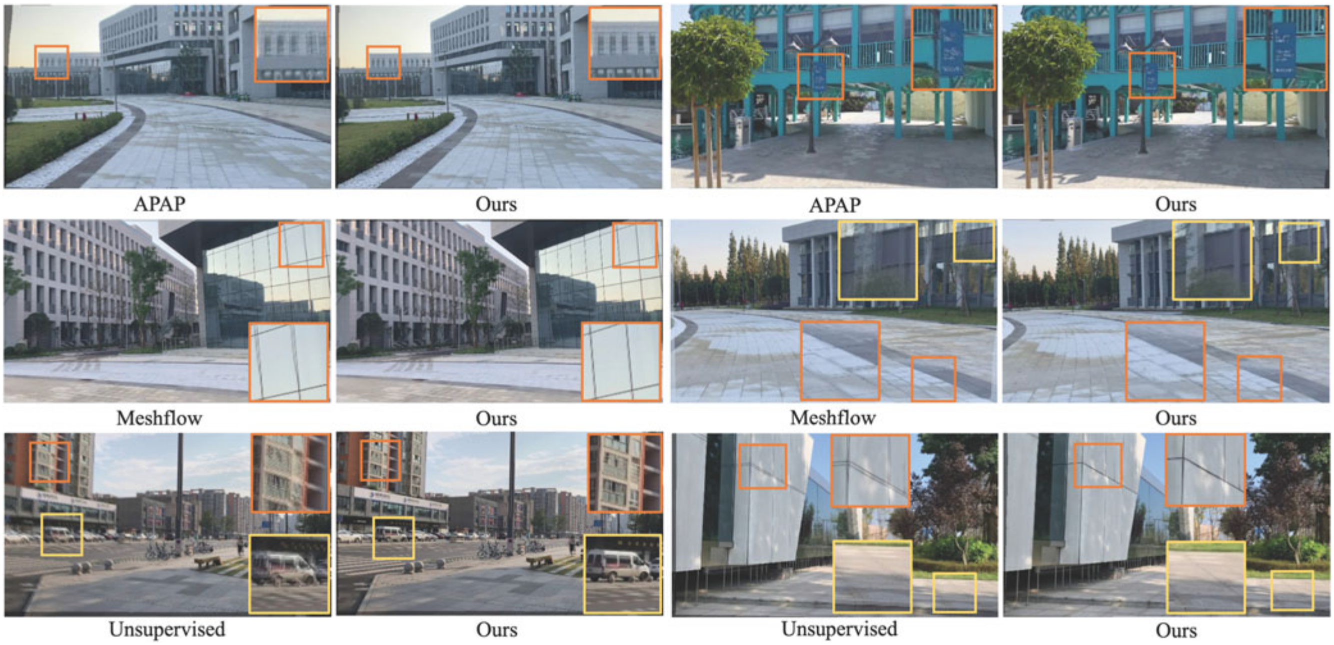


Fig. 11. Visual comparison with mesh-based approaches. We select three methods, APAP [8], Meshflow [2] and unsupervised deep homography [15], which are mostly related to our method for comparisons.

TABLE 6
Quantitative Comparison Between Our Deep Meshflow Method and all Other Methods Including Traditional Mesh-Based (Row 3, 4) and Homography-Based (Row $5 \sim 7$)

1)		RE	LT	LL	SF	LF	Avg
2)	$\mathcal{I}_{3\times3}$	7.98	8.39	7.61	8.70	4.86	7.51
3)	APAP [8]	2.53	3.45	2.67	2.69	2.55	2.78
4)	Meshflow [2]	1.73	2.48	1.97	2.06	2.28	2.10
5)	SIFT + RASAC	1.73	2.93	5.25	2.00	2.17	2.81
6)	Unsupervised [15]	1.94	3.50	2.45	2.27	2.36	2.51
7)	Ours-Homo	1.88	**2.36**	2.23	2.13	2.13	2.15
8)	Ours-Mesh	**1.71**	2.37	**1.93**	**1.88**	**1.96**	**1.97**

Note that during the evaluation, we use all the labeled point pairs to quantify the alignment accuracy of multiple planes in the image. As can be seen, our Deep Meshflow method can produce better results than other methods.

4.5 Mesh-Based Registration

We have extended Deep Homography to Deep Meshflow in Section 3.4. Here, we compare our mesh-based registration with several representative methods, including the classical traditional methods Meshflow [2], As-Projective-As-Possible mesh Warping [8] and unsupervised deep homography [15]. The source image is warped to the target image, where two images are blended for illustration. Methods that produces clearer blended images indicate good alignment. For each method, we show two examples as shown in Fig. 11. The first, second, and third row shows the comparison with As-Projective-As-Possible(APAP), Meshflow, and unsupervised deep homography approaches, respectively, in which our results are shown in the second and fourth columns. We highlight some regions for clearer illustration. These scenes contain depth variations of multiple planes, some of which are very close to the camera. As a result, a single homography is barely qualified. Moreover, some of the scenes contain repetitive structures that trouble the image feature matching, limiting the quality of traditional feature based mesh methods. In comparison, our Deep Meshflow is robust against these scenarios. In Table 6, we report the quantitative comparison result of our Deep Meshflow with other methods on our test data set. During the evaluation, all the labeled point pairs are used to compute the warping distance error as illustrated in Section 4.1. The experiment result shows that our Deep Meshflow method can produce better performance than traditional mesh-based registration methods and homography-based registration methods.

4.6 Failure Cases

Although our method achieves state-of-the-art performance in small baseline scenes compared with the existing methods, it still has its limitation of being applied to large baseline scenes. The reason behind may lie in the limited perception field of the network which is unable to perceive the alignment information between the two images. With this limitation, our method is unable to be applied to applications relying on large baseline alignment such as image stitching. We show two failure results in Fig. 12 for large baseline scenes by our method, in comparison with those by SIFT+RANSAC. As seen, SIFT+RANSAC produces stable results for the scenes. We will leave the solution for the large baseline alignment as a future work.

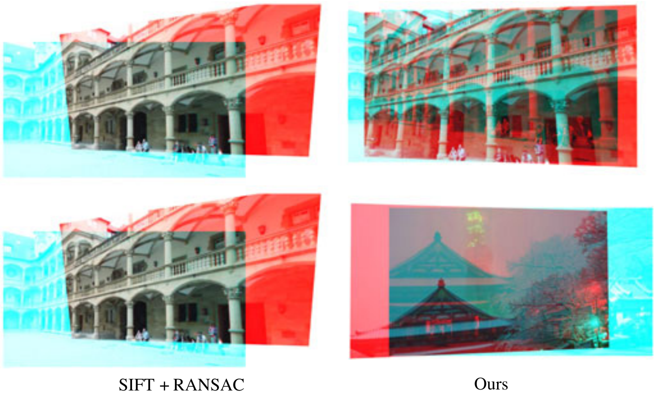


Fig. 12. Our method may fail when applied to images with large baseline. Left: images aligned by SIFT + RANSAC induced homography. Right: aligned by our Deep Homography.

5 Conclusion

We have presented a new architecture for unsupervised deep homography estimation with content-aware capability. Traditional feature based methods heavily rely on the quality of image features which are vulnerable to low-texture and low-light scenes. Large foreground also causes troubles for RANSAC outlier removal. Previous DNN-based solutions pay less attention to the depth disparity issue, and they treat the image content equally which can be influenced by non-planar structures or dynamic objects. To tackle these issues, our network learns a mask during the estimation to reject outlier regions for robust homography estimation, acting as a neural RANSAC. We also calculate loss with respect to learned deep features instead of directly comparing the image intensities. We further formulate a novel triplet loss to achieve the unsupervised training of our network. In addition, based on the proposed homography pipeline, we further show that it is possible to be extended for mesh based registration, by regressing multiple motion vectors at the mesh vertexes, resulting in a Deep Meshflow motion model.

We have conducted extensive experiments to demonstrate the effectiveness of the modules in our network and the triplet loss we designed. Results also reveal the superior capabilities of our method against the state-of-the-art, including DNN-based and feature-based solutions, on a newly presented comprehensive dataset for image alignment. The dataset is divided into 5 categories of scenes, which can be used for future research of image alignment models.

References

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge, U.K.: Cambridge Univ. Press, 2003.

[2] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, "MeshFlow: Minimum latency online video stabilization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 800–815.

[3] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2462–2470.

[4] N. Gelfand, A. Adams, S. H. Park, and K. Pulli, "Multi-exposure imaging on mobile devices," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 823–826.

[5] B. Wronski *et al.*, "Handheld multi-frame super-resolution," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–18, 2019.

[6] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun, "Fast burst images denoising," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–9, 2014.

[7] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 78.

[8] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving DLT," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2339–2346.

[9] H. Guo, S. Liu, T. He, S. Zhu, B. Zeng, and M. Gabbouj, "Joint video stitching and stabilization from moving cameras," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5491–5503, Nov. 2016.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[11] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[12] G. Simon, A. W. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," in *Proc. Int. Symp. Augmented Reality*, 2000, pp. 120–128.

[13] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[15] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar, "Unsupervised deep homography: A fast and robust homography estimation model," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2346–2353, Jul. 2018.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," 2016, *arXiv:1606.03798*.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[20] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 467–483.

[21] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Commun. Statist.-Theory Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[22] D. Barath, J. Matas, and J. Noskova, "MAGSAC: Marginalizing sample consensus," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10197–10205.

[23] B. D. Lucas *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[24] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[25] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1858–1865, Oct. 2008.

[26] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1385–1392.

[27] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "DeepMatching: Hierarchical deformable dense matching," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 300–323, 2016.

[28] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "SOSNet: Second order similarity regularization for local descriptor learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11016–11025.

[29] H. Altwaijry, A. Veit, S. J. Belongie, and C. Tech, "Learning to detect and match keypoints with deep architectures," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 49.1–49.12.

[30] H. Le, F. Liu, S. Zhang, and A. Agarwala, "Deep homography estimation for dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7649–7658.

[31] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[32] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 44.

[33] S. Li, L. Yuan, J. Sun, and L. Quan, "Dual-feature warping-based motion model estimation," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 4283–4291.

[34] K. Lin, S. Liu, L.-F. Cheong, and B. Zeng, "Seamless video stitching from hand-held camera inputs," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 479–487, 2016.

[35] K. Lin, N. Jiang, S. Liu, L.-F. Cheong, M. Do, and J. Lu, "Direct photometric alignment by mesh deformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2405–2413.

[36] F. Zhang and F. Liu, "Parallax-tolerant image stitching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3262–3269.

[37] M. Brown and D. Lowe, "Recognising panoramas," in *Proc. Int. Conf. Comput. Vis.*, 2003, pp. 1218–1225.

[38] K. Han *et al.*, "SCNet: Learning semantic correspondence," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1831–1840.

[39] I. Rocco, R. Arandjelović, and J. Sivic, "End-to-end weakly-supervised semantic alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6917–6925.

[40] S. Kim, S. Lin, S. Jeon, D. Min, and K. Sohn, "Recurrent transformer networks for semantic correspondence," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 6129–6139.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[43] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 224–236.

[44] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

[45] M. Dusmanu *et al.*, "D2-Net: A trainable CNN for joint description and detection of local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8092–8101.

[46] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2D2: Reliable and repeatable detector and descriptor," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 12405–12415.

[47] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 757–774.

[48] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5173–5182.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[50] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[51] T. Mertens, J. Kautz, and F. Van Reeth, "Exposure fusion," *Comput. Graph. Forum*, vol. 28, no. 1, pp. 382–390, 2007.

Shuaicheng Liu (Member, IEEE) received the BE degree from Sichuan University, Chengdu, China, in 2008, and the MSc and PhD degrees from the National University of Singapore, Singapore, in 2010 and 2014, respectively. He is currently an associate professor with the Institute of Image Processing, School of Information and Communication Engineering, University of Electronic Science and Technology of China. His research interests include computer vision and computer graphics.

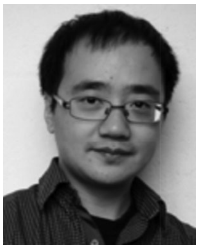
Nianjin Ye received the BEng degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2017, and the MS degree from the University of Electronic Science and Technology of China, in 2020. He is a researcher in Megvii Technology, Chengdu. His research interests include computer vision and deep learning.

Chuan Wang received the BEng degree from the University of Science and Technology of China, in 2010, and the PhD degree from The University of Hong Kong, in 2015. He was a computer vision staff researcher in Lenovo Group Limited, Hong Kong. He started his training program in Megvii, in 2018. His research interests include video analysis and computer vision.

Jirong Zhang received the BEng degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2017, and the MS degree from the University of Electronic Science and Technology of China, in 2020. He is currently an engineer with the China Academy of Space Technology. He is engaged in the intelligent design of spacecraft. Before that he has been an assistant researcher in Megvii Technology, Chengdu. His research interests include computer vision and deep learning.

Lanpeng Jia received the BEng degree from the Chengdu University of Technology, Chengdu, China, in 2012, and the MS degree from the University of Electronic Science and Technology of China, Chengdu, in 2015. He is now a researcher in Megvii Technology, Chengdu. His research interests include image processing and deep learning.

Kunming Luo received the BEng degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2016, and the MS degree from the University of Electronic Science and Technology of China, in 2019. He is now a researcher in Megvii Technology, Chengdu. His research interests include computer vision and deep learning.



Jue Wang (Senior Member, IEEE) received the BE and MSc degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2000 and 2003, respectively, and the PhD degree in electrical engineering from the University of Washington, Seattle, Washington, in 2007. He is the research manager of the Visual Computing Center, Tencent AI Lab. Before that, he was a senior director of Megvii from 2017 to 2020, and was a principle research scientist with Adobe Research. He received the Microsoft Research Fellowship and the Yang Research Award from the University of Washington, in 2006. He is a senior member of ACM. His research interests include image and video processing and computational photography.

Jian Sun (Senior Member, IEEE) received the BS, MS, and PhD degrees from Xian Jiaotong University, in 1997, 2000 and 2003, respectively. He is currently the chief scientist with the Megvii Technology. Immediately following, he joined Microsoft Research Asia, and has been working in the fields of computer vision and computer graphics, with particular interests in solving fundamental research problems and building real-world working systems. His primary research interests are computational photography and deep learning based image understanding. He has won the best paper awards twice with the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2009 and 2016.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.