

# Deep Homography Estimation for Dynamic Scenes

Hoang Le

Portland State University

{hoanl, fliu}@cs.pdx.edu

Feng Liu\*

Shu Zhang

Google

shzhang@google.com

Aseem Agarwala\*

Adobe Research

aseem@agarwala.org

## Abstract

Homography estimation is an important step in many computer vision problems. Recently, deep neural network methods have shown to be favorable for this problem when compared to traditional methods. However, these new methods do not consider dynamic content in input images. They train neural networks with only image pairs that can be perfectly aligned using homographies. This paper investigates and discusses how to design and train a deep neural network that handles dynamic scenes. We first collect a large video dataset with dynamic content<sup>1</sup>. We then develop a multi-scale neural network and show that when properly trained using our new dataset, this neural network can already handle dynamic scenes to some extent. To estimate a homography of a dynamic scene in a more principled way, we need to identify the dynamic content. Since dynamic content detection and homography estimation are two tightly coupled tasks, we follow the multi-task learning principles and augment our multi-scale network such that it jointly estimates the dynamics masks and homographies. Our experiments show that our method can robustly estimate homography for challenging scenarios with dynamic scenes, blur artifacts, or lack of textures.

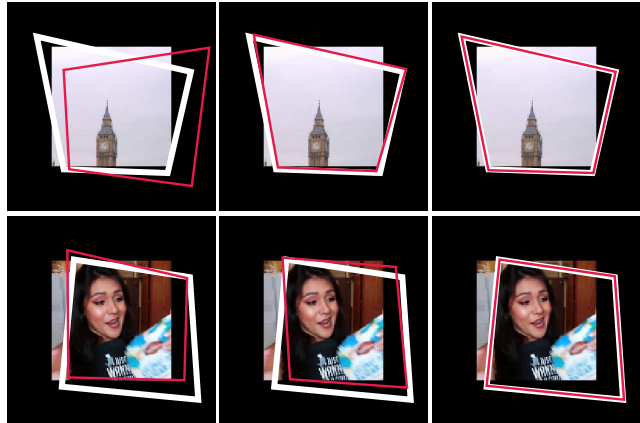
## 1. Introduction

A homography models the global geometric transformation between two images. It not only directly serves as a motion model for many applications like video stabilization [14, 24] and image stitching [31, 33], but also is used to estimate 3D motion and scene structure in algorithms, such as SLAM [10] and visual odometry [13].

There are two categories of methods for homography estimation [31]. Direct photometric-based approaches search for an optimal homography that minimizes the alignment error between two input images. Sparse feature-based approaches first estimate sparse feature correspondences between the input images and then compute the homography

\*Work mostly done while Feng and Aseem were at Google Research.

<sup>1</sup>Dataset: <https://github.com/lcmhoang/hmg-dynamics>



SIFT+RANSAC [21]

DHN [9]

Our results

Figure 1. Challenging examples for homography estimation. Each figure shows one of the two input images. Each white box indicates where the other image in the pair will be warped to in the first image according to the ground truth homography and each red box indicates the result using the estimated homography. As shown in the top row, the lack of texture fails the traditional SIFT + RANSAC method while the deep learning methods work well. The example at the bottom is difficult as the moving person dominates the image. Compared to both the SIFT-based method and the deep homography estimation method [9], our dynamics-aware method can better estimate homography for a dynamic scene.

from them. The direct approaches are robust to images with large textureless regions but are challenged by large motion. In contrast, while sparse feature-based approaches are more robust to large motion, they heavily depend on the quality of feature correspondences, which are often difficult to obtain in textureless regions or blurry images.

Recent deep homography estimation methods train a convolutional neural network to compute a homography between two images [8, 9, 11, 26]. These deep neural network methods leverage both local and global features and can often perform better than the traditional methods on challenging scenarios, such as images lacking texture, as shown in the first example of Fig. 1. However, how they handle challenging scenarios such as dynamic scenes has not been studied, as existing methods focus on image pairs that can be fully aligned using homographies. In prac-

tice, dynamic objects are common. Traditional non-deep learning approaches use a robust estimation algorithm like RANSAC [12] to exclude them as outliers. We need to empower deep learning methods with the capability of being resistant to dynamic content to make them more practical.

This paper investigates the problem of developing a deep learning method for homography estimation that is able to handle dynamic scenes without an iterative process like RANSAC. We first introduce a multi-scale deep convolutional neural network to handle image pairs with a large global motion. This network first estimates the homography from the low-resolution version of the input image pair and then progressively refines it at the increasingly higher resolutions. The architecture of the base neural network at each stage follows a VGG-style network similar to existing deep homography estimation methods [9, 26]. Our study shows that this multi-scale neural network can handle not only a large global motion, but also dynamic scenes to some extent when properly trained.

To address the problem of homography estimation of a dynamic scene in a more principled way, we need to detect dynamic content and eliminate their effect on homography estimation. Actually, homography estimation and dynamic content detection are two tightly coupled tasks. According to the research on multi-task learning, training a neural network to simultaneously perform these two tasks can greatly improve its performance for both tasks. Therefore, we enhance our multi-scale network such that it jointly estimates the dynamics mask and homography for an image pair. To train this network, in addition to the homography loss, we use an auxiliary loss function that compares the dynamics mask from the ground-truth dynamics map that is estimated from the training data. This multi-task learning training strategy empowers our multi-scale homography estimation network to robustly handle dynamic scenes.

As there are no publicly available large number of image pairs or videos of scene dynamics that come with known ground-truth homographies, we collected 32,385 static video clips from YouTube with a Creative Commons license. We then applied random homography sequences to these static clips to produce the training examples. These examples contain a wide variety of dynamic scenes. As shown in our experiments, our neural networks trained on this synthetic dataset generalize well to real-world videos.

This paper contributes to the research on homography estimation by investigating ways to develop and train deep neural networks that are robust against dynamic scenes. First, we build a large video dataset with dynamic scenes for training deep homography estimation neural networks. Second, we develop a multi-scale neural network that can handle large motion and show that by carefully training, it can already reasonably accommodate dynamic scenes. Third, we develop a dynamics-aware homography estimation net-

work by integrating a dynamics mask network into our multi-scale network for simultaneous homography and dynamics estimation. Our experiments show that our method can handle various challenges, such as dynamic scenes, blurriness, lack of texture, and poor lighting conditions.

## 2. Related Work

Homography estimation is one of the basic computer vision problems [16]. According to multi-view geometry theory, two images of a planar scene or taken by a rotational camera can be related by a  $3 \times 3$  homography matrix  $\mathbf{H}$ :

$$\hat{\mathbf{x}} = \mathbf{H}\mathbf{x} \quad (1)$$

where  $\hat{\mathbf{x}}$  and  $\mathbf{x}$  are the homogeneous coordinates of two corresponding points in the two images. Note, the above equation is only valid for corresponding points on static objects. Below we first briefly describe two categories of off-the-shelf algorithms for homography estimation and then discuss the recent deep neural network based approaches.

Pixel-based approaches directly search for an optimal homography matrix that minimizes the alignment error between two input images. Various error metrics between two images and parameter searching algorithms, such as hierarchical estimation and Fourier alignment, have been developed to make direct approaches robust and efficient. These direct methods are robust to images lacking in texture, but often have difficulty in dealing with large motion. Feature-based approaches are now popular for homography estimation. They first estimate local feature points using algorithms, such as SIFT [21] and SURF [4], and then match feature points between two images. For a video, corner points are often detected and then tracked across two consecutive frames for efficiency [29]. Given the set of corresponding feature pairs, an optimal homography matrix can be obtained by solving a least squares problem based on Eq. 1. In practice, errors can occur during feature matching and feature points can come from moving objects. Therefore, a robust estimation algorithm like RANSAC [12] and Magsac [3] is often used to remove the outliers. The performance of feature-based approaches depends on local feature detection and matching. For images suffering from blurriness or lacking in texture, they often cannot work well.

Our work is more related to the recent deep learning approaches to homography estimation. In their seminal work, DeTone *et al.* developed VGG-style deep convolutional neural networks for homography estimation. They showed that a deep neural network can effectively compute the homography between two images [9]. Nguyen *et al.* extended this work by training the neural network using a photometric loss that measures the pixel error between a warped input image and the other one. This photometric loss allows for the unsupervised training of the neural network without ground-truth homographies [26]. To deal with the large motion between two images, Nowruzi *et al.*

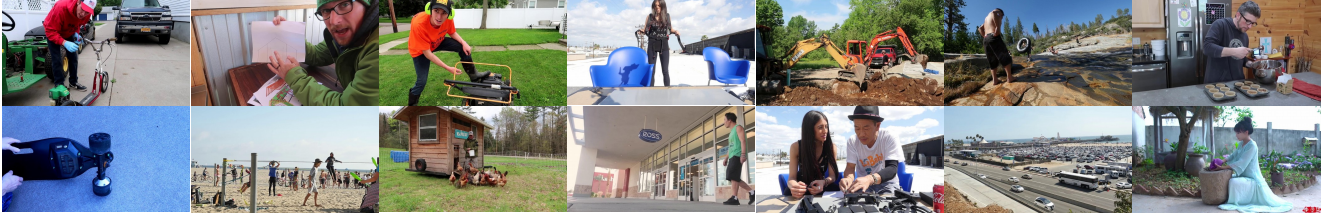


Figure 2. Static video samples in our dynamic-scene homography dataset.

developed a hierarchical neural network that iteratively refines homography estimation [11]. Chang *et al.* designed a Lucas-Kanade layer that is able to regress the homography between two images from their corresponding feature maps extracted by a shared convolutional neural network [8]. This Lucas-Kanade network can be cascaded to progressively refine the estimated homography. Zeng *et al.* developed a perspective field neural network to model the pixel-to-pixel bijection between two images and used that for homography estimation [32]. These deep homography estimation approaches are shown successful on images of a static scene; however, they do not consider dynamic scenes.

Our multi-scale neural network extends the multi-stage approaches discussed above. Compared to Nowruzi *et al.* [11], our method starts from low-resolution versions of the input images and gradually increases the input image sizes, instead of taking the original input images as input at each stage. This makes our method more robust to large motion. Compared to Chang *et al.* [8], our method pre-aligns the input images to the next stage using the homography estimated in the previous stage to minimize the global motion. This helps the late-stage network to account for the global motion. More importantly, we further enhance our multi-scale neural network with a dynamics mask network to handle dynamic scenes, which was not considered in the previous neural network-based methods.

### 3. Homography Dataset of Dynamic Scenes

Existing deep neural networks for homography estimation are trained with image pairs that can be perfectly aligned. A common approach is to use a subset of the MSCOCO image set [19], and warp each image using a known homography to form an image pair [8, 9, 11]. Since we aim to train a neural network that can handle dynamic scenes, we cannot produce a dataset in this way.

Ideally, each image pair in our dataset should contain dynamic scenes with a known homography. To our best knowledge, there is not such a public homography dataset that is large enough for training a deep neural network. Our solution is to first collect a large set of videos capturing dynamic scenes while the cameras are held static, and then apply known homography sequences to them. Specifically, we downloaded 877 videos with a Creative Commons License from YouTube. From these videos, we extracted 32,385 static video clips and then applied a known homography se-

quence to each of them to generate image/video pairs. Fig. 2 shows some sample frames from this video set.

**Static video clip detection.** Since our goal is to build a reliable homography benchmark to train a deep neural network, it is critical that each image pair in our dataset can be perfectly aligned by a homography (except moving objects.) Thus, instead of using a structure-from-motion method to estimate and analyze camera motion, we use a more conservative approach *aiming for a very high precision rate at the cost of a low recall rate when identifying static clips.*

We observe that if a video clip has camera motion, its boundary area changes temporally. Also, the central part of a video clip may change significantly due to the scene motion if the video is captured by a static camera. Accordingly, for every two consecutive frames, we calculate the similarity between their boundary areas. Specifically, we first resize each video frame to  $256 \times 256$ . We consider the outer boundary with a 5-pixel thickness. We subdivide the boundary area into 32 boundary blocks of size  $32 \times 5$  pixels. If more than 25% of blocks stay the same between two consecutive frames, we consider that there is no camera motion between the two frames. We consider a boundary block unchanged over time if more than 90% of its pixels only slightly change the color ( $\delta_c \leq 6.67$ ). To further remove non-static video clips, we estimate optical flows between frames in each video clip using the PWC-Net algorithm [30]. Since the difference between two consecutive frames is small, we skip  $\delta t = 7$  frames to compute the optical flow. We remove the clips if the areas of moving pixels in the frames are greater than 65%. Finally, we consider a video clip static if every two consecutive frames are static. To avoid the drifting error, we also enforce that the following frames in a sequence must also be considered static with regard to the first frame of the sequence. We extract static video clips that have a minimum length of 10 frames from a video using a greedy search method.

We finally manually examine the resulting video clips to remove those misidentified as static ones. In total, we obtained 32,385 static video clips. The average length of these video clips is 22 frames and the scene motion ranges from 0-25 pixels. We split the dataset into three portions, 70% for training, 20% for testing and 10% for validation.

**Image pair generation.** Given a sequence of  $n$  frames  $\{I_i, 1 \leq i \leq n\}$  in a video clip, we randomly sample 2 frames  $I_j$  and  $I_k$ , with  $1 \leq j, k \leq n$ , and  $|j - k| \leq 5$ .

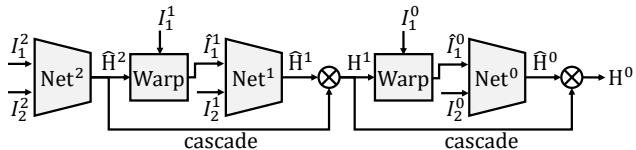


Figure 3. Multi-scale neural network for homography estimation. This network progressively estimates and refines homography from coarse to fine.

We then use a similar method to Detone *et al.* [9] to generate a pair of image patches, one from  $I_j$  and the other from  $I_k$ . Specifically, we first randomly crop an image patch  $I_j^p$  of size  $128 \times 128$  at location  $[x_p, y_p]$  in  $I_j$ . We then randomly perturb the coordinates of the four corner points of  $I_j^p$  by a value  $r$ ,  $-32 \leq r \leq 32$ . We use the four corner displacements to compute the corresponding homography  $\mathbf{H}$ . Then, we apply the inverse of this homography,  $\mathbf{H}^{-1}$ , on the other image  $I_k$ :  $\hat{I}_k = \mathbf{H}^{-1}(I_k)$ , and then extract the corresponding patch at the same location  $[x_p, y_p]$  to obtain image patch  $\hat{I}_k^p$ . Each training sample contains  $I_j^p$ ,  $\hat{I}_k^p$ , and their corresponding ground-truth homography  $\mathbf{H}$ .

## 4. Homography Estimation Neural Networks

We first introduce a multi-scale deep homography estimation network that handles large motion, and then describe how we improve it with a dynamics mask network to better handle dynamic scenes.

### 4.1. Multi-scale Neural Network

Our neural network takes two gray-scale images of size  $128 \times 128$  as input, and outputs the homography between them. Following the previous work [9, 11, 26], we use the displacements of the four image corners to represent the homography. As reported in previous work [1, 8, 11], progressively estimating and refining a homography with a multi-stage procedure is helpful to cope with large global motion between two images. We extend these methods and design a multi-scale multi-stage neural network to estimate the homography for a pair of images from coarse to fine.

Fig. 3 illustrates a three-scale version of our multi-scale homography neural network. Given a pair of images  $(I_1, I_2)$ , we build a pair of image pyramid  $(I_1^k, I_2^k)$ , where  $k$  indicates the pyramid level. The down-scaling factor at level  $k$  is  $2^k$ . We start from the highest scaling level  $(I_1^2, I_2^2)$  and use the base network  $\text{Net}^2$  to estimate a homography  $\hat{\mathbf{H}}^2$ . Then, we warp  $I_1^1$  using  $\hat{\mathbf{H}}^2$  to obtain a pre-aligned image pair  $(\hat{I}_1^1, I_2^1)$ . The homography of the unaligned image pair  $(I_1^1, I_2^1)$  can be computed by cascading  $\hat{\mathbf{H}}^2$  and  $\hat{\mathbf{H}}^1$ .

$$\mathbf{H}^1 = \hat{\mathbf{H}}^1 \mathbf{S}^{-1} \hat{\mathbf{H}}^2 \mathbf{S} \quad (2)$$

where  $\mathbf{H}^1$  is the homography of the unaligned image pair  $(I_1^1, I_2^1)$  and  $\mathbf{S}$  is a scaling matrix that down-samples an image by half to account for the different image sizes at the two scales. We continue by using  $\mathbf{H}^1$  to pre-align  $(I_1^0, I_2^0)$  as input for the subsequent base network and obtain

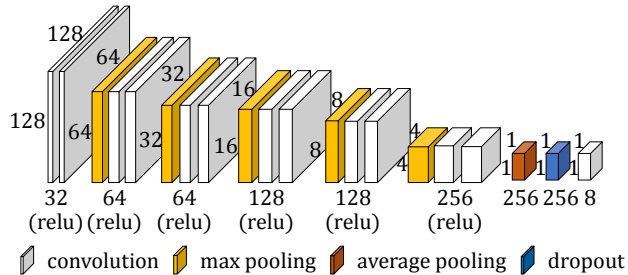


Figure 4. Architecture of the base network  $\text{Net}^0$ .

the homography  $\hat{\mathbf{H}}^0$  and compute the homography for the original image pair in a similar way to Eq. 2.

We use a variation of the neural network introduced by DeTone *et al.* [9] as the base network at each stage. Fig. 4 shows the architecture of the base network  $\text{Net}^0$ . It starts with twelve  $3 \times 3$  fully convolutional layers, each coupled with Batch Normalization [17] and a Rectified Linear Unit layer (ReLU). Every two consecutive convolutional layers are followed by a max pooling layer. These convolutional layers are followed by an average pooling layer, a dropout layer and a  $1 \times 1$  convolution layer that outputs a homography vector of length 8. The architecture of other base networks  $\text{Net}^k$  is similar except that each has  $2k$  less convolutional layers due to the smaller input image size.

To train this multi-scale neural network, we first calculate the  $l_2$  loss between the estimated and ground-truth homography, each parameterized using a corner displacement vector. We then compute the total loss at all the scales as the loss function.

When our multi-scale neural network is trained using the examples derived from the MS-COCO dataset in a similar way to previous work [9], it works well with image pairs with only static background, for which the transformation can be perfectly modeled by a homography. However, when it is used to estimate homographies for image pairs with scene motion, the results are less accurate as reported in Sec. 5. This is expected as the training examples derived from the MS-COCO dataset do not contain dynamic scenes.

To further examine the capability of this multi-scale neural network, we trained another version on our dynamic dataset described in Sec. 3. We found that training this multi-scale network on our dataset can improve its capability in handling dynamic scenes, as found in our experiments (Sec. 5.2). To better handle scene motion, we develop a mask-augmented network as described in the next section.

### 4.2. Mask-augmented Deep Neural Network

To better handle dynamic scenes, we need to identify dynamic content in the scene. Actually, dynamic content detection and homography estimation are two tightly coupled tasks. An accurate estimation of homography helps robustly detect dynamic content while correctly identifying dynamic content helps with accurate homography estimation. Based

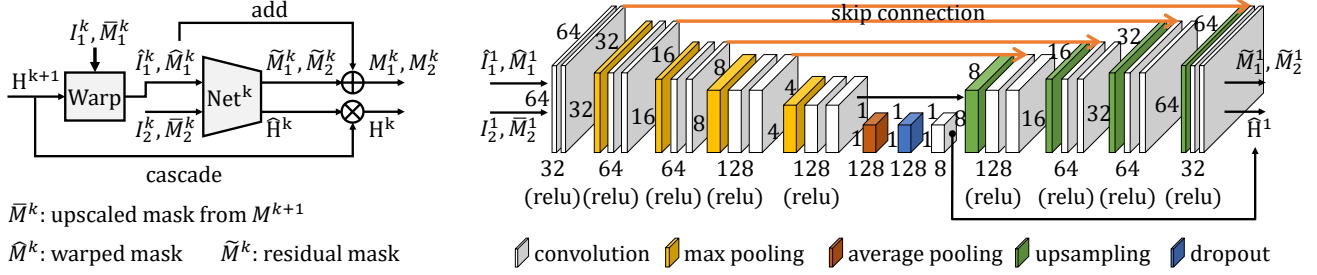


Figure 5. Mask-augmented Deep Neural Network at level  $k = 1$ . The network is augmented with a convolutional decoder that helps predict areas of dynamic objects to improve the homography prediction.

on the research on multi-task learning, jointly training a model to perform these two tasks simultaneously can enable the success of both tasks [7, 28]. Accordingly, we improve our base homography network so that it estimates both the dynamics maps and the homography for a pair of images. To this end, we incorporate a dynamics mask estimator into our base homography neural network, as shown in the right of Fig. 5. This new network uses the same architecture as the previous base network except that we add a sub-neural network to regress a pair of dynamics maps. We refer to this new base network as a mask-augmented base network.

We embed this mask-augmented base network into our multi-scale network in a similar way as before. As shown in the left of Fig. 5, the mask-augmented base network  $\text{Net}^{k+1}$  outputs a pair of dynamics maps  $(M_1^{k+1}, M_2^{k+1})$  and a homography  $\hat{\mathbf{H}}^{k+1}$ , from which we can compute a cascaded homography  $\mathbf{H}^{k+1}$  according to Eq. 2. Then, we first up-sample the pairs of dynamics maps to match the image size in the next level  $k$  and obtain  $(\bar{M}_1^k, \bar{M}_2^k)$ . Then we warp  $\bar{M}_1^k$  using the homography  $\mathbf{H}^{k+1}$  and obtain  $\tilde{M}_1^k$  to match the warped input image  $\hat{I}_1^k$ , and finally concatenate  $(\tilde{M}_1^k, \bar{M}_2^k)$  and  $(\hat{I}_1^k, \hat{I}_2^k)$  as input for the new base network at the next level  $k$ . We further improve the performance of this mask-augmented network by first estimating the mask residuals  $(\tilde{M}_1^k, \tilde{M}_2^k)$  and then obtaining the final masks by adding the residuals to the input upsampled masks  $(\hat{M}_1^k, \hat{M}_2^k)$ .

This mask-augmented homography neural network is difficult to train as it cannot automatically figure out the role of the dynamics maps. According to multi-task learning [7, 28], having an extra supervising signal helps train a neural network. So we estimate the ground-truth dynamics maps and use them as the extra supervising signals. Specifically, since each image pair  $(I_i, I_j)$  is created from two frames of a static video, we first estimate the optical flow between the two video frames using PWC-Net [30] and then create a ground truth mask by labelling a pixel as 1 if the flow magnitude is greater than 1 pixel and 0 otherwise. We then generate a pair of masks  $(M_i, M_j)$ , one for each image in the pair, by cropping and warping the ground truth mask in the same way as we generate the image pair  $(I_i, I_j)$ .

Now we have two loss functions: a homography loss  $l_f$  described in Sec. 4.1 and a dynamics mask-based loss  $l_d$ .

We calculate  $l_d$  as a binary cross entropy loss as follows [5].

$$l_d = \frac{-M_g \cdot \log(M_p)^T - (1 - M_g) \cdot \log(1 - M_p)^T}{|M_g|} \quad (3)$$

where  $M_p$  and  $M_g$  are the predicted dynamics mask and the ground-truth mask, each arranged as a row vector.  $|M_g|$  is the total number of pixels in  $M_g$ . We combine the two losses to train our network:  $L = \sum_k \sigma_f l_f^k + \sigma_d l_d^k$  where  $l_f^k$  and  $l_d^k$  are the homography loss and mask loss at scale  $k$  and  $\sigma_f$  and  $\sigma_d$  are weights.

**Implementation details.** Our neural networks are implemented using TensorFlow. We initialize the weight values using the Xavier algorithm [15] and the biases with zero. We use the Adam optimization algorithm to train the neural networks [18] using an exponentially decaying learning rate with an initial value of  $10^{-4}$ , a decay step of  $10^5$ , and a decay rate of 0.96. The dropout keep probability is set to 0.8. The mini-batch size is 32.

We train the mask-augmented network in three steps. First, we train it on our dynamic-scene dataset with  $\sigma_f = 1$  and  $\sigma_d = 0$  for  $2 \times 10^6$  global iterations so that the network learns to predict homography only with ground-truth homography labels. We continue training it for another  $10^6$  iterations with  $\sigma_f = 1$  and  $\sigma_d = 10$  so that it learns to predict both the homography and dynamics masks by leveraging the ground-truth masks. Finally, we train it for another  $10^6$  iterations with  $\sigma_f = 1$  and  $\sigma_d = 0$  so that it learns to leverage the predicted dynamics masks to ultimately boost the performance of homography estimation for dynamic scenes.

## 5. Experiments

We evaluate two versions of our method: the multi-scale homography network (MHN) and the dynamics mask-augmented network (MHN<sub>m</sub>). We compare them to both feature matching-based and deep learning-based methods. The first category of methods use the SIFT feature or its recent variations followed by a robust estimation method, including SIFT+RANSAC [21], SIFT+MAGSAC [3], SIFT+GeoDesc+RANSAC [23], SIFT+ContextDesc+RANSAC [22], LF-Net [27], and AFFNET [25]. We also use PWC-Net [30], a state-of-the-art optical flow method, to estimate dense correspondences

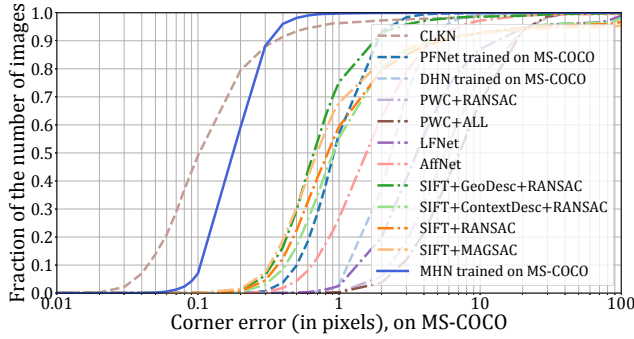


Figure 6. Evaluation on static scenes.

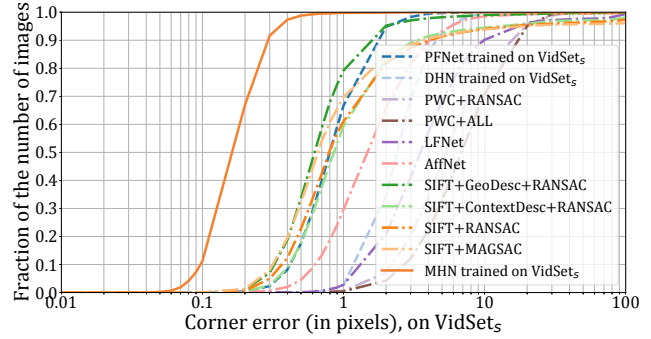


Figure 7. Evaluation on dynamic scenes.

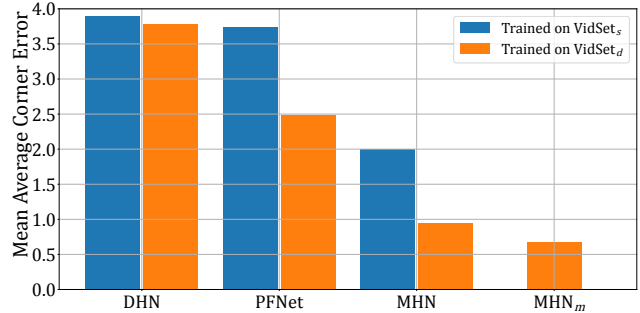
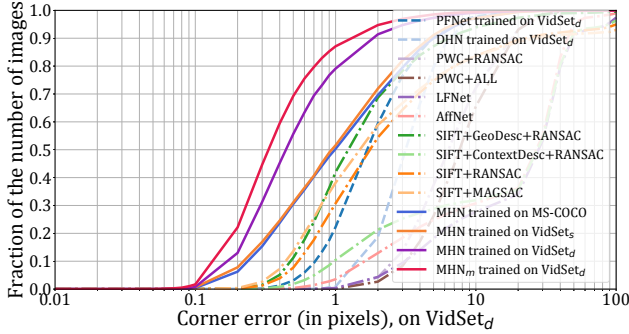


Figure 8. Effect of training datasets. Our dynamic-scene dataset  $\text{VidSet}_d$  enables deep homography estimation networks better handle dynamic scenes than its static version  $\text{VidSet}_s$ .

between input images and then use RANSAC to estimate the homography. The deep learning homography estimation methods include CLKN [8], DHN [9], and PFNet [32].

Our experiments use three datasets. The first is derived from the MS-COCO [19], following the procedure in the recent work [8, 9]. The image size is  $128 \times 128$  and the image corners are randomly shifted in the range of  $[-32, 32]$  pixels. Please refer to [8, 9] for more details. The second, named as  $\text{VidSet}_s$ , is the static version of our dynamic-scene dataset by creating image pairs from a single video frame so that there is no dynamic content in each image pair. The third is our dynamic-scene dataset described in Sec. 3, named as  $\text{VidSet}_d$ . Following [8], we use the mean corner error as the evaluation metric:  $e_c = \frac{1}{4} \sum_{j=1}^4 \|c_j - \hat{c}_j\|_2$  where  $c_j$  and  $\hat{c}_j$  are corner  $j$  transformed by the estimated homography and the ground-truth homography, respectively.

### 5.1. Evaluation on Static Scenes

Existing deep homography estimation methods focus on pairs of images that can be fully aligned by homography. We compare our multi-scale network (MHN) with these methods on both MS-COCO and  $\text{VidSet}_s$ . As shown in Fig. 6, our multi-scale network described in Sec. 4.1 outperforms the competitive learning-based methods [9] and [32] but performs slightly worse than CLKN [8] in the high-precision region. Moreover, our method outperforms all the competitive feature matching or flow-based methods. This is consistent with the reports from the previous work [8, 9] that deep neural network approaches can often perform better than the conventional matching-based methods.

### 5.2. Evaluation on Dynamic Scenes

In this test, we train and test our networks MHN and  $\text{MHN}_m$  and the other competitive networks, including DHN [9] and PFNet [32], on our dynamic-scene dataset  $\text{VidSet}_d$ . Fig. 7 shows that our networks significantly outperform the existing learning-based approaches and matching-based approaches in handling dynamic scenes, even though we train the existing deep learning methods on our dynamic-scene dataset. In addition, when trained on  $\text{VidSet}_d$ , our multi-scale network MHN can already handle dynamic scenes to some extent. On around 80% of examples, it achieves a good accuracy of  $\leq 1.0$  pixel. Our dynamics mask augmented network  $\text{MHN}_m$  further improves its performance, and achieves the same accuracy for more than 85% of the testing images. We were not able to evaluate the performance of CLKN [8] on  $\text{VidSet}_d$  since their code is not available. However, we also expect that our mask network and our  $\text{VidSet}_d$  can also be married to CLKN to better handle dynamic scenes.

**Effect of training sets.** We trained two versions of each homography network, one on the static version of our dynamic-scene dataset ( $\text{VidSet}_s$ ) and the other on our dynamic-scene dataset ( $\text{VidSet}_d$ ). We tested them on the testing set of  $\text{VidSet}_d$ . As shown in Fig. 8, our dynamic-scene dataset can greatly improve most of these deep homography estimation networks in handling dynamic scenes.

**Effect of the dynamic area size.** To examine how our methods work on images with a various amount of dynamic

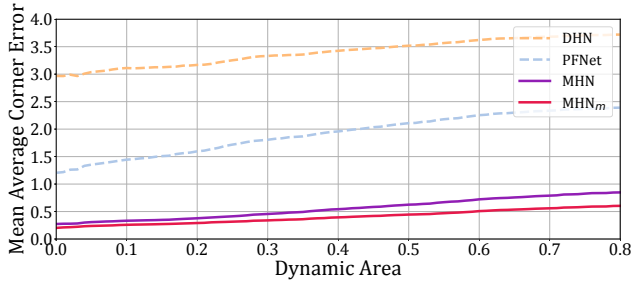


Figure 9. Effect of the size of dynamic area. Compared to existing deep homography estimation methods, such as DHN [9] and PFNet [32], our methods are robust against a large dynamic area. In our testing content, we calculate the dynamic area ratio of each example. Then, we create multiple versions of the testing set. For each set, we select a dynamic area ratio threshold and only include examples with the dynamic area ratio smaller than that threshold. We then test our methods on these testing sets. As reported in Fig. 9, both of our methods, MHN and  $MHN_m$  are more stable than the other deep homography methods when the dynamic area increases.

### 5.3. Discussions

**Scale selection.** An important hyper-parameter of our multi-scale neural network is the number of scales. We found that when increasing the number of scales from one to three, our network can be trained significantly faster. As shown in Fig. 10, it can handle larger global motion. But when the number of scale goes beyond three, the training becomes unstable. We attribute this to the very small image size that is processed by the first base neural network. This is similar to what Chang *et al.* reported [8].

**Real-world videos.** We experimented on videos from the NUS stabilization benchmark [20]. We estimate the homography between two frames that are ten frames apart. Fig. 11 shows several challenging examples with significant scene motion and lack of texture in the static background. We visualize each homography estimation result by first using the homography to warp  $Frame_0$  to align with  $Frame_{10}$  and then creating a red-cyan anaglyph from the warped  $Frame_0$  and  $Frame_{10}$ . Specifically, we take the red channel from  $Frame_{10}$  and the cyan channel from the warped  $Frame_0$  and merge these channels into an anaglyph image. Any non-boundary colorful pixels not on a moving object indicate misalignment. While our network is trained using synthetic examples (using ground truth homography estimated from images having dynamic objects and static background), it works well with the real videos. Fig. 11 also shows that our network can accurately identify the dynamic content by examining the dynamics masks.

**Parallax.** Although there is no parallax in the static background in our training set, our network can handle parallax well for the above real-world examples. Since no homography in theory can perfectly account for the parallax in an image pair, we examine how our network handles it. To

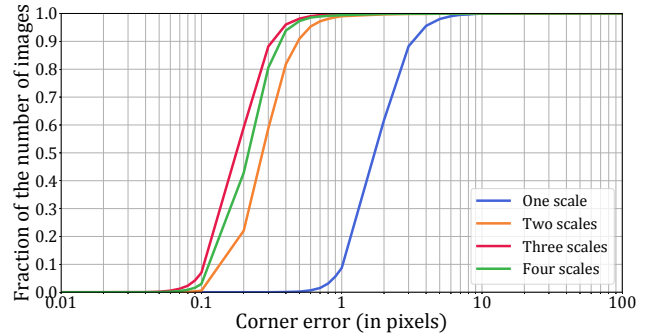


Figure 10. Effect of the number of scales in our multi-scale homography estimation network on MS-COCO.

this end, we test our method on examples from optical flow benchmarks, namely Middlebury [2] and Sintel [6]. In this test, we use our method to estimate the homography between two frames, align them using the estimated homography, and finally compute optical flow between the two aligned frames. As shown in Fig. 12 (c), there is little motion in the background, while the objects that are close to the camera are not aligned. This suggests that our method finds a homography that accounts for the motion in an as-large-as possible area while treating the foreground objects as outliers. As also shown in Fig. 12 (d), our method identifies the foreground object in each example that is far away from the background plane in the dynamics map. Note, while the foreground objects do not move, they are outliers for homography estimation similar to a moving object.

## 6. Conclusion

This paper studied the problem of estimating homography for dynamic scenes. We first collected a large video dataset of dynamic scenes. We developed a multi-scale, multi-stage deep neural network that can handle large global motion and achieve the state-of-the-art homography estimation results when trained and tested on examples derived from the MS-COCO dataset. We further showed that this multi-scale network, when trained on our dynamic-scene dataset, can already handle dynamic scenes to some extent. We then extended this multi-scale network by jointly estimating the dynamics masks and homographies. The dynamics masks enable our method to deal with dynamic scenes better. Our experiments showed that our deep homography neural networks can handle challenging scenarios, such as dynamic scenes, blurriness, and lack of texture.

**Acknowledgements.** Fig. 1 (top), Fig. 11, 12 (top), and Fig. 12 (bottom) originate from MS-COCO [19], NUS [20], Middlebury [2], and Sintel [6] datasets respectively. Fig. 1 (bottom) and Fig. 2 are used under a Creative Commons license from Youtube users Nikki Limo, chad schollmeyer, Lumnah Acres, Liziqi, Dielectric Videos, and 3DMachines. We thank Luke Ding for helping develop our dataset.



Figure 11. Real-world video examples from the NUS video stabilization benchmark [20]. We estimate homography for pairs of frames that are 10 frames apart. Each homography result is visualized using the red-cyan anaglyph that takes its red channel from  $\text{Frame}_{10}$  and the cyan channel from the warped  $\text{Frame}_0$ . Any non-boundary colorful pixels **not on a moving object** indicate misalignment.

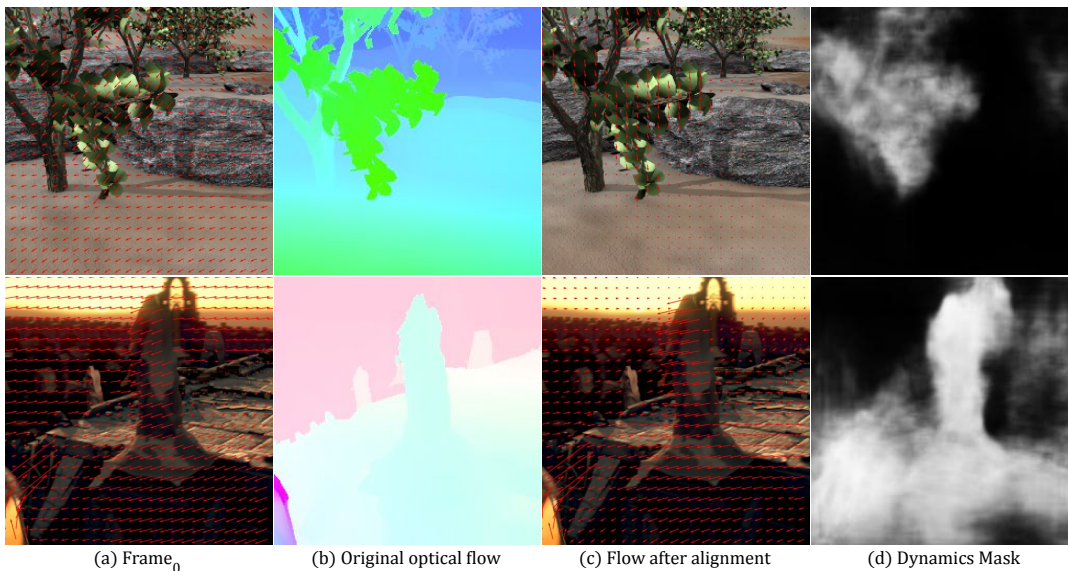


Figure 12. Evaluation on parallax. These examples show that after a pair of images are aligned, there is little motion in the background as shown in (c). This suggests that our method finds a homography that accounts for the motion in an as-large-as-possible area. As shown in (d), our method treats foreground objects as outliers in the same way as they treat dynamic content.



## References

- [1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 4
- [2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 7
- [3] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: marginalizing sample consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10197–10205, 2019. 2, 5
- [4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 2
- [5] Christopher M Bishop. *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006. 5
- [6] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision*, pages 611–625, 2012. 7
- [7] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, pages 41–48, 1993. 5
- [8] Che-Han Chang, Chun-Nan Chou, and Edward Y Chang. Clkn: Cascaded lucas-kanade networks for image alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2213–2221, 2017. 1, 3, 4, 6, 7
- [9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 1, 2, 3, 4, 6, 7
- [10] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 1
- [11] Farzan Erlik Nowruzi, Robert Laganiere, and Nathalie Japkowicz. Homography estimation from image pairs with hierarchical convolutional networks. In *Proceedings of The IEEE International Conference on Computer Vision Workshop*, pages 913–920, 2017. 1, 3, 4
- [12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2
- [13] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014. 1
- [14] Michael L Gleicher and Feng Liu. Re-cinematography: improving the camera dynamics of casual video. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 27–36, 2007. 1
- [15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 5
- [16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 2
- [17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 4
- [18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015. 5
- [19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing, 2014. 3, 6, 7
- [20] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics*, 32(4):78, 2013. 7, 8
- [21] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2, 5
- [22] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2527–2536, 2019. 5
- [23] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–183, 2018. 5
- [24] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):1150–1163, 2006. 1
- [25] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–300, 2018. 5
- [26] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018. 1, 2, 4
- [27] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6234–6244. Curran Associates, Inc., 2018. 5
- [28] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 5
- [29] Jianbo Shi et al. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600, 1994. 2

- [30] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of CNNs for optical flow estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, Jan. 2019. [3](#), [5](#)
- [31] Richard Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2007. [1](#)
- [32] Rui Zeng, Simon Denman, Sridha Sridharan, and Clinton Fookes. Rethinking planar homography estimation using perspective fields. In *Asian Conference on Computer Vision*, pages 571–586. Springer, 2018. [3](#), [6](#), [7](#)
- [33] Fan Zhang and Feng Liu. Parallax-tolerant image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3269, 2014. [1](#)