Analyzing the Domain Shift Immunity of Deep Homography Estimation

Mingzhen Shao , Tolga Tasdizen , Sarang Joshi
Scientific Computing and Imaging Institute, University of Utah
{shao, tolga, sjoshi}@sci.utah.edu

Abstract

Homography estimation is a basic image-alignment method in many applications. Recently, with the development of convolutional neural networks (CNNs), some learning based approaches have shown great success in this task. However, the performance across different domains has never been researched. Unlike other common tasks (*e.g.*, classification, detection, segmentation), CNN based homography estimation models show a domain shift immunity, which means a model can be trained on one dataset and tested on another without any transfer learning. To explain this unusual performance, we need to determine how CNNs estimate homography. In this study, we first show the domain shift immunity of different deep homography estimation models. We then use a shallow network with a specially designed dataset to analyze the features used for estimation. The results show that networks use low-level texture information to estimate homography. We also design some experiments to compare the performance between different texture densities and image features distorted on some common datasets to demonstrate our findings. Based on these findings, we provide an explanation of the domain shift immunity of deep homography estimation. [1]

1 Introduction

Homography is one of the most fundamental concepts in computer vision. It provides a geometric relationship for any two images of the same planar surface in space. Therefore, estimating homographies correctly among images is the first step in understanding scene geometry, which can significantly improve the performance of many vision tasks such as multiframe HDR imaging [Gelfand *et al.*, 2010], multiframe image super resolution [Wronski *et al.*, 2019], burst image denoising [Liu *et al.*, 2014], video stabilization [Liu *et al.*, 2013], image/video stitching [Zaragoza *et al.*, 2013; Guo *et al.*, 2016], and SLAM [Mur-Artal *et al.*, 2015; Zou and Tan, 2013].

[1] https://github.com/MingzhenShao/Homography_estimation

Methods for homography estimation can be either geometric or deep learning based. Geometric based methods aim to find geometrically meaningful correspondences (*e.g.*, points, edges) across visual data and then match them to compute homographies. The performance of such methods is reliant on the accuracy of the feature correspondences. These methods perform quite well when correspondences can be clearly detected and matched. However, searching for correspondences across different viewpoints can be difficult and time-consuming under some conditions. Sometimes, developers may even need to design different methods for different scenes based on the viewpoint, illumination, and image properties to achieve better search results.

For these reasons, deep learning based methods have attracted interest in recent years. Some models have been proposed and have achieved impressive accuracy in their testing datasets [DeTone *et al.*, 2016; Nguyen *et al.*, 2018; Erlik Nowruzi *et al.*, 2017; Wang *et al.*, 2019; Le *et al.*, 2020; Zhang *et al.*, 2020]. However, none of these works analyze the performance across different domains. Based on a widely accepted concept, we need some domain transfer before applying a deep learning model to a different domain, but this is not the case in the homography estimation task. All deep learning based homography estimation models show a domain shift immunity, which means we do not need any domain transfer between different datasets.

We demonstrate the domain shift immunity by comparing the performance of the same model on different domains without any fine-tuning. In order to explain this unusual performance, the most direct approach is to find out which features are used for the estimation. The biggest challenge for this approach is the feature density. As we know from the geometric methods, a homography matrix can be calculated with very limited correspondences. If we directly visualize the feature map with real-world images, the rich information in such images may cause many disturbances when we analyze the features. Therefore we design a special dataset with simple shapes and visualize the feature maps with these images. By analyzing the features, we find the networks focus on the low-level texture information (features extracted by the initial convolutional layers such as edges, corner points, etc.).

Two experiments are designed to demonstrate that low-level texture information is the key feature for homography estimation. We first compare the performance among differ-

ent texture densities in the proposed dataset. Then we change different features of real-world images in some common datasets and compare the performance of the estimations. Based on our findings, we provide an explanation of the domain shift immunity of deep homography estimation. To the best of our knowledge, the present study is the first attempt to analyze how deep learning models estimate homography in different domains. Our contributions can be summarized as follows:

1. We find the domain shift immunity of deep learning based homography estimating models.
2. We propose a dataset and a visualizing method to show the focus of homography estimation models.
3. We provide an explanation of the domain shift immunity of homography estimation models.

2 Related Work

The estimation of homography by traditional approaches generally requires matched image feature points, such as SIFT [Lowe, 2004], SURF [Bay *et al.*, 2006], ORB [Rublee *et al.*, 2011], LPM [Ma *et al.*, 2017], GMS [Bian *et al.*, 2020], SOSNet [Yurun *et al.*, 2019], LIFT [Yi *et al.*, 2016], and OAN [Zhang *et al.*, 2019]. After a set of feature correspondences is obtained, a homography matrix is estimated by Direct Linear Transformation (DLT) [Hartley and Zisserman, 2004] with some outlier rejection methods, such as RANSAC [Fischler and Bolles, 1981], IRLS [Holland and Welsch, 1977], and MAGSAC [Barath *et al.*, 2019]. These traditional approaches heavily rely on the quality of image features, if the feature correspondences are well captured, they commonly achieve good performance. Estimations, however, may be inaccurate due to an 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.*, sky, ocean, grassland), repetitive patterns (*e.g.*, forest, bookshelf, symmetrical building), or illumination variations. Moreover, if the image contains dynamic objects (*e.g.*, a moving bus), it will be more challenging for the outlier rejection methods. There is another type of traditional approach can estimate a homography without feature correspondences, the direct method. These approaches, such as the Lucas-Kanade algorithm [1981], calculate the sum of squared differences (SSD) between two images. The differences guide the shift of the images, yielding updates about the homography. A follow-up method also proposed an enhanced correlation coefficient (ECC) [Evangelidis and Psarakis, 2008] to replace the SSD for robustness. Compared to the feature points based methods, the direct methods are more sensitive to the interference (*e.g.*, dynamic objects, illumination variations).

In recent years, inspired by the success of various deep learning based methods in many challenging tasks, a deep learning based homography estimating model is first proposed by DeTone *et al.* [2016]. This model has only eight convolutional layers and provides an end-to-end homography estimation. It takes source and target images as input and uses manually generated groundtruth to supervise the training. Some later works [Erlik Nowruzi *et al.*, 2017; Le *et al.*, 2020] also follow this pipeline and replace the backbone with some more complex network structures for better performance. There is an obvious challenge for supervised training: getting the homography matrix from real image pairs is difficult. A commonly adopted method is using a synthetic target image to avoid this problem, but this method may cause a depth disparity. In order to solve this problem, some unsupervised training methods are proposed. Nguyen *et al.* [2018] propose an unsupervised approach that computes photometric loss between two images and adopts a spatial transform network (STN) [Jaderberg *et al.*, 2015] for image warping. After that, Zhang *et al.* [2020] propose a method for learning a content-aware mask instead of calculated loss directly on the intensity and uniformly on the image plane in order to increase the prediction accuracy of the unsupervised training.

These learning based methods can provide pixel-level performance that is much better than traditional methods. However, unlike traditional ones, none of these methods explain how estimation works and if it can handle the input images from different domains. Our work shows that the deep learning based homography estimation task has domain shift immunity, and such immunity is backbone structure unrelated. By analyzing the feature used for the estimation, we also provide an explanation to the domain immunity.

3 Methods

3.1 Reforming the Homography Matrix

The most widely used representation of a homography is a 3×3 transformation matrix and fixed scale. Using $[u, v]$ for pixels in an image and $[u', v']$ for its projection to the other image in a homogenous coordinate plane, we get the representation of a homography matrix as follows:

$$\begin{pmatrix} u' \\ v' \\ 1' \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (1)$$

However, these nine parameters $(H_{11}, H_{12}, ..., H_{33})$ mix the rotational and translational terms performed on different scales in a single vector. Directly using these parameters to train deep learning models will lead to unbalance problems, increasing the difficulty of training. To solve this problem, we use four 2D offset vectors (eight values) to represent the homography matrix.

The method to reform a homography to our four 2D offset vectors is as follows:

1. choose four points with position $(u_i, v_i), i \in [1, 4]$ that can make up a rectangle.
2. find the same four points at the homogenous coordinate plane with position $(u'_i, v'_i), i \in [1, 4]$.
3. calculate as $\Delta u_i = u'_i - u_i, \Delta v_i = v'_i - v_i, i \in [1, 4]$ as the 2D offset vectors.

With the four offset vectors, it is straightforward to obtain the homography matrix H with 8 degrees of freedom by solv-

ing a linear system. The four-point parameterization represents a homography as follows:

$$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix} \quad (2)$$

3.2 Homography Estimation Network

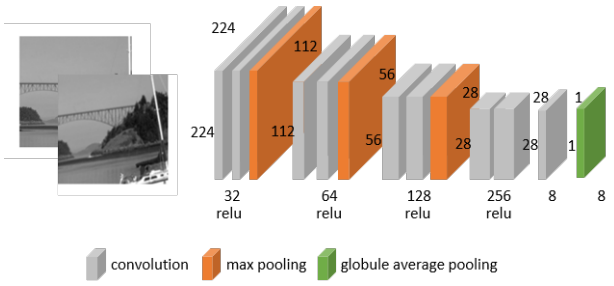


Figure 1: Homography estimation network structure

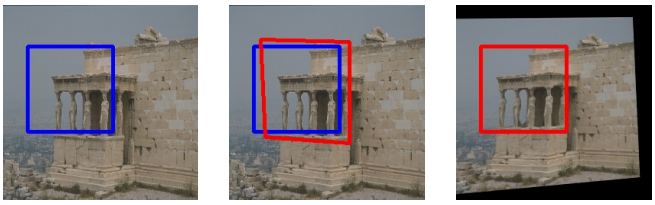
Compared to the very first approach proposed by DeTone *et al.*, some very deep models with complex backbones have been proposed for better performance in accuracy or robustness. However, these complex backbones increase the difficulty of analyzing how the models perform across different domains. Furthermore, the experiments show that the domain shift immunity does not rely on some specific backbone structures.

We propose a shallow network structure with only nine convolution layers to avoid the disturbance of too many parameters. Our homography estimation network (HEN) uses a global average pooling (GAP) layer to turn the eight channel feature maps into eight output values, which also makes analyzing easier than using the fully connected layer. Our HEN takes a grayscale source and target image pair as input and outputs the eight values (H_{4point}) to represent the homography matrix. Due to the depth of the model, our HEN could not provide comparable accuracy to the state-of-the-art approaches, but it can still provide pixel-level accuracy that is robust enough to analyze the domain shift immunity. The structure of our homography estimation network is shown in Figure 1.

3.3 Data Generation

We use the synthetic image pairs to train and test our networks. These generated image pairs may not be ample enough to show the homography transformation in the real world, but they are sufficient to compare the performance of different networks in common states. What's more, with these synthetic image pairs, we can easily compare the performance between different domains.

To generate an image pair, we first randomly crop a square patch I_s of size 128×128 at position p from the reference image I. (We avoid the borders to prevent bordering artifacts later in the data generation pipeline.) The four corners of the image patch I_s are randomly perturbed by values δ within the range $[-32, 32]$, and thus the four correspondences define a homography H. Then, we apply the inverse of the homography H^{-1} to I to produce the image I'. A second patch I_d is cropped from I' at the same position p. The two patches I_s and I_d are converted into grayscale and then stacked channel-wise to generate a 2-channel image that is used as the input for the model. The generating process is illustrated in Figure 2.



(a) Randomly crop a square at position p from the original image I as I_s.

(b) Perturb four corners of the square to get a tetragon and compute the homography H.

(c) Apply H^{-1} to I and crop a square at the same position p as image I_d.

Figure 2: Data generation

3.4 Geometric Simple Shape Dataset

Visualizing the focus of a deep learning based model is a widely used method for analyzing how it works. However, unlike other common deep learning tasks, homography can be calculated from very limited correspondences. Simply visualizing the networks' focus on common datasets (*e.g.*, BSD300, AFLW2000, etc.) will only result in high response areas with no logical consistency.

Therefore, we designed a simple dataset named geometric simple shape (GSS) to eliminate the disturbance of the rich information while analyzing the model's focus. The images in the GSS share a black background and contain only simple geometric shapes (squares, triangles, or circles). These shapes are located at a random position with arbitrary size and color. The number of shapes in each image is less than ten, and any elaborate curvilinear shapes are avoided.

Figure 3 is a demonstration of the focus visualizations on different datasets using the class activation mapping (CAM) method proposed by Zhou *et al.* [2016]. The top line is a visualization on common datasets, and the bottom line is the visualization on our proposed GSS dataset. With the proposed GSS dataset, we can significantly reduce the disturbance of too many features (with some decline in accuracy).

3.5 Relative Focus Visualization

Another challenge in visualizing the focus of a homography estimation model is the basic noise. The basic noise, which is the output of a homography estimation model when we input a pair of full black images. All the outputs should be zero because there is no homograph transform between these images. Still, due to the bias values in each convolutional layer, the network will provide nonzero outputs. The basic noise will not cause trouble during normal prediction, but when we use CAM to visualize the focus, especially on the GSS dataset (most areas are black), it may import some disturbance into our focus visualizations.

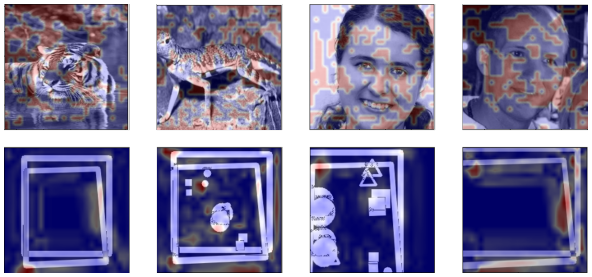


Figure 3: Visualizing the focus on different datasets. (top: common datasets (BSD300 and AFLW2000), bottom: GSS)

In order to remove the disturbance of the basic noise, we use a new relative focus visualization method. With the H_{4point} homography matrix, the output of the last convolutional layer contains eight channels, one for each value in the H_{4point}. Instead of directly visualizing the feature maps, we enhance their interpretability by subtracting them channel-wise from the output of a black image pair. Using Is for an input GSS image pair and Bs for a black image pair,

$$\textit{Focus}_i = abs(ch_i(Is) - ch_i(Bs)), \quad i \in \{1, ..., 8\} \qquad (3)$$

where $\textit{Focus}_i$ is the focus map of the ith channel, and $ch_i(Is)$, $ch_i(Bs)$ are the feature maps of ith channel from the last convolutional layer with the input image pair Is, Bs separately. The green circles in Figure 4 show the basic noise in the original feature map, which has been removed by the proposed visualization method.

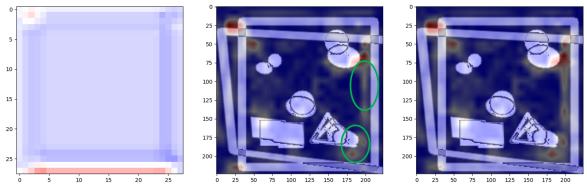


Figure 4: Comparison of focus maps on a single channel, with green circles highlighting regions affected by basic noise: (left) focus map of black image pair, (middle) focus map before removing basic noise, (right) focus map after removing basic noise.

4 Experimental Results and Analysis

In this section, we first show the immunity of homography estimation models to the domain shift. Then we use the proposed visualizing method on our GSS dataset to analyze the features used for homography estimation. The results show that the network is focusing on the low-level textures, such as edges.

By comparing the performance between different texture densities and distorted image features, we demonstrate that the low-level textures are the critical features for homography estimation. In addition, we provide an explanation of the domain shift immunity based on these findings. All models are trained on the BSD300 dataset with the proposed data generation method and tested on target datasets without any fine-tuning.

4.1 Immunity to Domain Shift

We use the BSD300, AFLW2000, and ISBI to test the immunity of homography estimation networks on different domains. These datasets cover a wide range of different domains such as scenery, faces, and cells. We also test the performance with varying network structures, from the proposed shallow HEN network to some very deep networks, such as VGG16 and ResNet50.

The results are shown in Table 1, and some demo images are shown in Figure 5. We first observe that although there is some volatility in shallow networks, the predictions for each network in different domains (results in each row) all achieve pixel-level accuracy, which is an order of magnitude less than the errors associated with a classical ORB descriptor with the RANSAC method (around 11.7 pixels) on comparable datasets [DeTone *et al.*, 2016]. A very deep network like ResNet50 even provides steady accuracy in all three domains. The pixel-level accuracy demonstrates the effectiveness of networks across various domains. From Figure 5, we can find the differences in the textures of the different datasets with $ISBI > BSD300 > AFLW2000$. A commonly acknowledged principle in the field of neural networks is that shallow networks possess limited capacity to extract essential features from inputs, which makes them more sensitive to the datasets. This principle explains the HEN becoming less accurate in each domain, and the shallow networks all performing worse in AFLW2000 compared to ISBI.

These results show: the deep homography estimation models have a domain shift immunity, and such immunity is network structure unrelated.

Dataset	AFLW2000	BSD300	ISBI
ResNet50	1.11	1.05	0.77
VGG16	3.42	2.60	2.25
HEN	5.84	5.47	4.95

Table 1: Prediction accuracy of networks in different domains (MAE in pixels)

4.2 Focus of Models

The immunity shows that the homography estimation models use some general features commonly existing in different domains. To find these features, we use the proposed HEN, GSS dataset, and visualization method to show the model's focus. We refrain from using deep networks like ResNet50 in this analysis as they have been shown to be robust in extracting information from inputs as seen in Table 1. This robustness may introduce significant interference when analyzing the accuracy changes after altering specific textures.

Before analyzing the focus map of the model, we first conduct a simple experiment to show the proposed visualization method can locate the high-contribution parts. The experiment compares the performance between two feature densities, normal2gap and selected2gap. Normal2gap uses all of the output of the final convolutional layer as input to the GAP layer, whereas selected2gap uses only the top 80% high-response features based on the $Focus_i$. Since we use the

Figure 5: Network prediction results on different domains (from top to bottom, BSD300, AFLW2000, and ISBI). The groundtruth is shown in blue and the prediction is shown in red.

GAP layer to turn the feature map into the final prediction, the relationship between the prediction accuracy and the features is easy to tell. Features that contain more relevant information for the prediction will result in greater accuracy. The prediction accuracy of these two type features is shown in Table 3.

Feature type	normal2gap	selected2gap
MAE (pixel)	13.73	12.07

Table 2: Prediction accuracy on GSS with different feature densities.

We first notice that the accuracy of the prediction for the GSS dataset is slightly worse than for other common datasets (last row in Table 1). We observe that the accuracy of the prediction on the GSS dataset is slightly less than on other commonly used datasets, as seen in the last row of Table 1. This decline is caused by the limited information contained in the GSS dataset. As discussed above, although registration networks can estimate homography with very limited correspondences, more information in the inputs can still provide better performance.

By comparing the MAE between normal2gap and selected2gap, selected2gap provides approximately 1.7 $pixels$ improvement in accuracy, which shows the proposed visualizing method can locate the high-contribution parts.

We use the proposed visualization method to show the focus of different images on the GSS dataset. Some illustrations are shown in Figure 6. The first two columns are the input image pairs, and the following eight columns are the visualization of focus on each channel. We can easily find the high-response areas located near low-level texture parts (edges in these images). Moreover, because homography can be estimated with very limited correspondences, we notice that not all edges are highly responded to.

4.3 Performance with Different Texture Densities

Based on the focus visualization results, we hypothesize that homography estimation models use low-level textures for the prediction. The straightforward approach is to compare the performance between different texture densities; the images with more low-level textures should perform better. The proposed GSS dataset can be easily applied to this task. The dataset can be separated into two types: only one shape and multiple shapes. These two types are tested with the HEN, and the prediction accuracy is listed in Table 3.

The input images with multiple shapes perform better, and from the focus map in Figure 6, we can see that the images with multiple shapes provide more high-response areas. Furthermore, it is evident that the distinction between these two types is based solely on the density of low-level textures. This makes it clear that low-level textures play a crucial role in homography estimation.

Image type	single shape	multiple shapes
MAE (pixel)	13.25	11.48

Table 3: Prediction accuracy between different texture densities.

4.4 Performance with Different Feature Distortions

Thus far, all experimental evaluations have been carried out utilizing our simplistic GSS dataset. We also need to test our findings on some real-world images. However, locating which low-level texture is used or comparing the low-level texture density between different datasets is difficult for

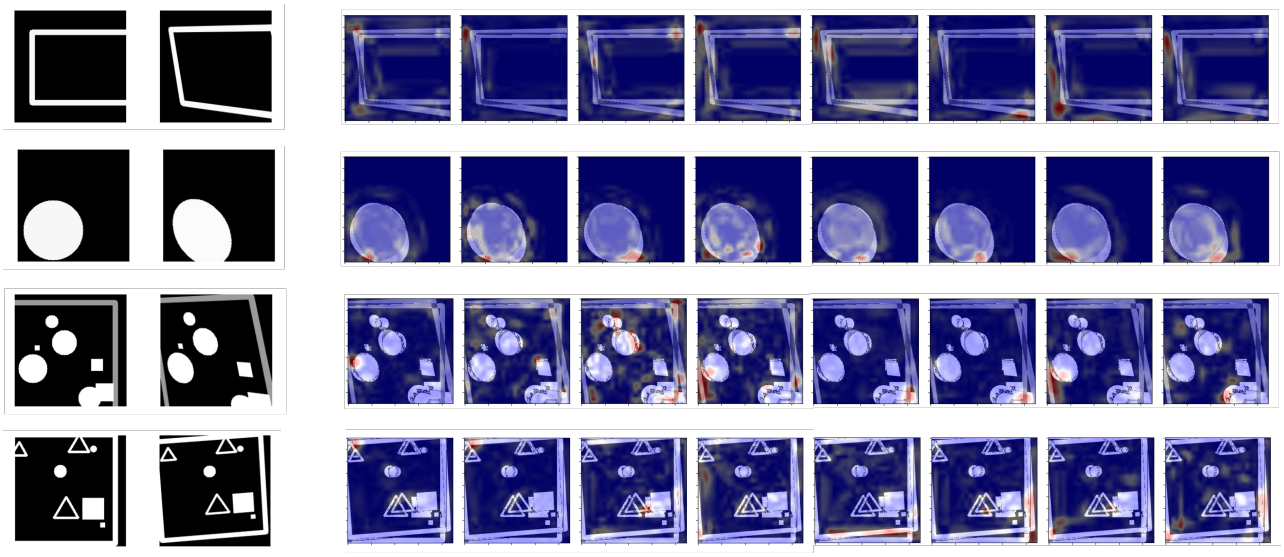


Figure 6: Focus map on the GSS dataset (shown in heatmap format)

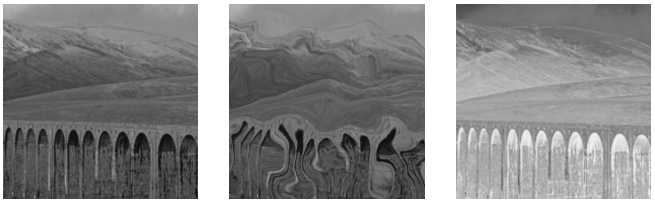
real-world images. Instead of directly locating these features, we evaluate the performance changes when applying different processing approaches to the features. We choose two classic image processing approaches: distortion and color inversion. The distortion will change low-level textures, and the color inversion changes only the color information and leaves the low-level textures almost the same. If low-level textures are the key features used for prediction in real-world images, registering the original image to an image with altered textures will result in a significant decrease in accuracy. Similarly, if other features are used for prediction (like the Lucas-Kanade method), there will be a decrease in accuracy upon registering to an image with inverted colors.

The distorted textures are generated using the following steps: We first initialize a random vector field ($12 \times 12 \times 2$). The range of the vector is $[-20, 20]$. Then we interpolate the field to the size of the original images and use this field for warping. The color inversion is simply using 255 minus the current value as the inverted color. We apply these two methods to the I_d while keeping I_s unchanged. Figure 7 is an illustration of these images.

We compare the performance of the proposed processing methods on the BSD300 and ISBI datasets. As shown in Table 4, the inverted color images share the same accuracy as the original images, whereas the texture distortion images are much less accurate. These results show that the deep homography estimation models also use low-level textures in real-world images.

4.5 Immunity to Domain Shift

By showing the homography estimation models use low-level textures for prediction, we can provide an explanation of the domain shift immunity. As the experiments show, the domains for which the estimation models provide good immunity all contain a variety of low-level textures. While applying the models to some texture-less datasets such as the GSS, the immunity is not as robust as on other texture density



(a) Original gray image I_d (b) Image with distorted textures (c) Image with inverted color

Figure 7: Images with different processing approaches

Dataset	BSD300	ISBI
Original images	5.47	4.95
Texture distortion images	11.71	9.12
Inverted color images	6.09	4.86

Table 4: Prediction accuracy for images with different processing approaches (MAE in pixels).

datasets. Therefore, the immunity of the deep homography estimation models is derived from two factors: the homography estimation task does not require a large amount of data (as a small number of well-matched correspondences can lead to an accurate estimation); and the deep models use low-level textures that are commonly present across various domains for estimation.

5 Conclusions

This paper studied the immunity of homography estimation networks to domain shift. We first demonstrate the domain shift immunity of different networks and show that the immunity is unrelated to the network structure. We further designed a special dataset and visualizing method to show the focus of the estimating networks is located around the low-

level textures. We then developed two experiments to evaluate the performance under different texture densities and feature distortions. Our experiments show that neural networks use low-level textures for homography estimation. Based on these findings, we explain the domain shift immunity of homography estimation networks.

References

[Barath *et al.*, 2019] D. Barath, J. Matas, and J. Noskova. Magsac: Marginalizing sample consensus. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10189–10197. IEEE Computer Society, 2019.

[Bay *et al.*, 2006] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision – ECCV 2006*, pages 404–417, 2006.

[Bian *et al.*, 2020] JiaWang Bian, Wen-Yan Lin, Yun Liu, Le Zhang, Sai-Kit Yeung, Ming-Ming Cheng, and Ian Reid. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. *International Journal of Computer Vision (IJCV)*, 2020.

[DeTone *et al.*, 2016] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv*, preprint arXiv: 1606.03798, 2016.

[Erlik Nowruzi *et al.*, 2017] 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 (ICCV) Workshops*, Oct 2017.

[Evangelidis and Psarakis, 2008] Georgios D. Evangelidis and Emmanouil Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

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

[Gelfand *et al.*, 2010] Natasha Gelfand, Andrew Adams, Sung Hee Park, and Kari Pulli. Multi-exposure imaging on mobile devices. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 823–826, 2010.

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

[Hartley and Zisserman, 2004] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2004.

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

[Jaderberg *et al.*, 2015] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[Le *et al.*, 2020] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[Liu *et al.*, 2013] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Trans. Graph.*, 32(4), 2013.

[Liu *et al.*, 2014] Ziwei Liu, Lu Yuan, Xiaoou Tang, Matt Uyttendaele, and Jian Sun. Fast burst images denoising. *ACM Trans. Graph.*, 33(6), 2014.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints, 2004.

[Lucas and Kanade, 1981] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, page 674–679, 1981.

[Ma *et al.*, 2017] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4492–4498, 2017.

[Mur-Artal *et al.*, 2015] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[Nguyen *et al.*, 2018] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo J Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. In *Robotics and Automation Letters 3(3)*, pages 2346–2353, 2018.

[Rublee *et al.*, 2011] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[Wang *et al.*, 2019] Chen Wang, Xiang Wang, Xiao Bai, Yun Liu, and Jun Zhou. Self-supervised deep homography estimation with invertibility constraints. *Pattern Recognition Letters*, 128:355–360, 2019.

[Wronski *et al.*, 2019] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Trans. Graph.*, 38(4), 2019.

[Yi *et al.*, 2016] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Computer Vision – ECCV 2016*, pages 467–483, 2016.

[Yurun *et al.*, 2019] Tian Yurun, Yu Xin, Fan Bin, Wu Fuchao, Heijnen Huub, and Balntas Vassileios.

Sosnet: Second order similarity regularization for local descriptor learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Zaragoza *et al.*, 2013] Julio Zaragoza, Tat-Jun Chin, Michael S. Brown, and David Suter. As-projective-as-possible image stitching with moving dlt. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[Zhang *et al.*, 2019] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. *International Conference on Computer Vision (ICCV)*, 2019.

[Zhang *et al.*, 2020] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *European Conference on Computer Vision*, pages 653–669, 2020.

[Zhou *et al.*, 2016] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[Zou and Tan, 2013] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):354–366, 2013.