# SCPNet - Supplementary Materials

Runmin Zhang[1*], Jun Ma[2,1*], Si-Yuan Cao[2,1*†], Lun Luo[3], Beinan Yu[1], Shu-Jie Chen[4], Junwei Li[1], and Hui-Liang Shen[1]

[1] College of Information Science and Electronic Engineering, Zhejiang University
[2] Ningbo Innovation Center, Zhejiang University
[3] HAOMO.AI Technology Co., Ltd.
[4] Zhejiang GongShang University

## 1 Details of Network

We will illustrate the network details of SCPNet in this section, including the **detailed structure of the correlation-based homography estimation network** and the **homography parameterization using offsets of four corner points**.

### 1.1 Detailed Structure of the Correlation-based Homography Estimation Network

**Feature Extractor**. The architecture of the feature extractor is illustrated in Fig. 1. The image is initially processed by a convolutional layer of kernel size $7 \times 7$. Subsequently, the produced features undergo a max-pooling layer of stride 2, followed by two residual blocks with 64 channels. The features are then proceeded by another max-pooling layer of stride 2 and two residual blocks with 96 channels. Finally, the features are projected into 256 channels through a convolutional layer of kernel size $1 \times 1$.

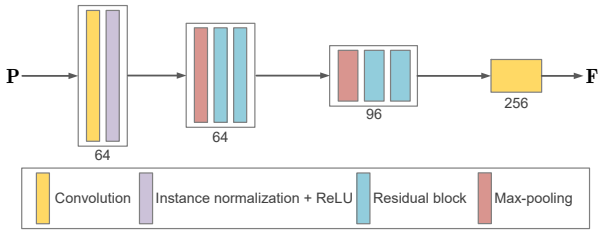



**Fig. 1:** The detailed architecture of the feature extractor.

**Homography Estimator**. We illustrate the architecture of the homography estimator in Fig. 2. Following the main text, the size of the correlation $\mathbf{C}$ is

* Equal Contributions. † Corresponding author. (cao_siyuan@zju.edu.cn)

$H \times W \times (2r+1) \times (2r+1)$. To enable the 2D convolution, $\mathbf{C}$ is reshaped into $(2r+1)(2r+1) \times H \times W$. As previous works [2,3,11], $\mathbf{C}$ is processed sequentially by a basic unit consisting of a convolutional layer of kernel size $3 \times 3$, a group normalization + ReLU layer, and a max-pooling layer of stride 2, until the spatial resolution of the feature is downsampled to $2 \times 2$. A convolutional layer is then used to project the feature to $2 \times 2 \times 2$, producing the residual offset prediction $\mathbf{O}$.

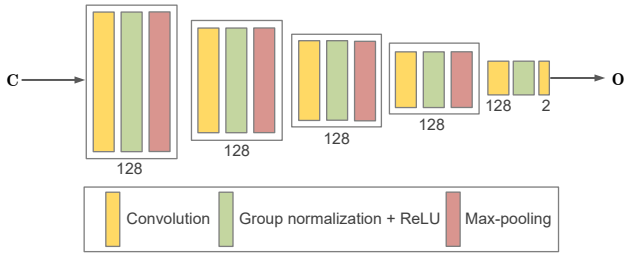



**Fig. 2:** The detailed architecture of the homography estimator.

### 1.2 Homography Parameterization using Offsets of Four Corner Points

Similar to the previous approaches [2, 3, 6, 9, 11], we use offsets of the four corner points of the image to parameterize the homography matrix, which can be expressed as

$$\mathbf{A}\mathbf{h} = \mathbf{b}, \tag{1}$$

where $\mathbf{b}$ is the coordinate of the projected 4 corner points, $\mathbf{A}$ is composed of the projected 4 corner points and the original 4 corner points, $\mathbf{h}$ is the vectorized $\mathbf{H}$. We define the 4 corner points in $\mathbf{I}_\mathrm{A}$ as $(u_1, v_1)$, $(u_2, v_2)$, $(u_3, v_3)$, $(u_4, v_4)$, and the corresponding projected ones in $\mathbf{I}_\mathrm{B}$ as $(u'_1, v'_1)$, $(u'_2, v'_2)$, $(u'_3, v'_3)$, $(u'_4, v'_4)$. Through the above 4 pairs of matched points, the predicted corner points $\mathbf{O}$ can be formulated as

$$\begin{aligned}
u'_1 &= u_1 + \mathbf{O}(0,0,0) \\
v'_1 &= v_1 + \mathbf{O}(1,0,0) \\
u'_2 &= u_2 + \mathbf{O}(0,0,1) \\
v'_2 &= v_2 + \mathbf{O}(1,0,1) \\
u'_3 &= u_3 + \mathbf{O}(0,1,0) \\
v'_3 &= v_3 + \mathbf{O}(1,1,0) \\
u'_4 &= u_4 + \mathbf{O}(0,1,1) \\
v'_4 &= v_4 + \mathbf{O}(1,1,1).
\end{aligned} \tag{2}$$

**A** can be expressed as

$$\mathbf{A} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1u'_1 & -v_1u'_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1v'_1 & -v_1v'_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2u'_2 & -v_2u'_2 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2v'_2 & -v_2v'_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3u'_3 & -v_3u'_3 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3v'_3 & -v_3v'_3 \\ u_4 & v_4 & 1 & 0 & 0 & 0 & -u_4u'_4 & -v_4u'_4 \\ 0 & 0 & 0 & u_4 & v_4 & 1 & -u_4v'_4 & -v_4v'_4 \end{bmatrix}, \tag{3}$$

and **b** as

$$\mathbf{b} = \begin{bmatrix} u'_1 & v'_1 & u'_2 & v'_2 & u'_3 & v'_3 & u'_4 & v'_4 \end{bmatrix}^{\mathsf{T}}. \tag{4}$$

Finally, the vectorized homography can be expressed as

$$\mathbf{h} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} & \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} & \mathbf{H}_{31} & \mathbf{H}_{32} \end{bmatrix}^{\mathsf{T}}, \tag{5}$$

## 2 More Details of Datasets

Fig. 3 shows the example images of the cross-modal datasets including GoogleMap [13] and Flash/no-flash [7], cross-spetral datasets including Harvard [4] and RGB/NIR [1], together with the manually-made inconsistent dataset PDS-COCO [8], under [-32,+32] offset. For fair comparison, all compared approaches are trained and tested on the same training and test splitting of each dataset.

**GoogleMap** is a cross-modal dataset including corresponding satellite and map images. We choose the map image as the source image and the satellite image as the target one. We then use the training and testing data shared in [13] with the size of $192 \times 192$. The $128 \times 128$ image pairs with homography deformation are produced by center cropping. The simulation of homography is implemented by randomly perturbing the four corner points of the $128 \times 128$ images.

**Flash/no-flash** contains 120 indoor and outdoor image pairs. We first resize the image to $320 \times 213$, and then generate a $128 \times 128$ image pair with simulated homography in the same way as GoogleMap.

**Harvard** contains multispectral images of 77 real-world scenes, each of which has 31 spectral bands in the spectral range of 420nm∼720nm. We first resize the image to $348 \times 260$, and then use the 16th band (*i.e.* 570nm) image of each scene as the source image, and generate the target image by applying simulated homography to other bands.

**RGB/NIR** dataset has images of the RGB and NIR spectral bands. We resize the images to $256 \times 256$, select RGB as the target image, and NIR as the source one to generate homography deformation. As can be seen from Fig. 3 (d), there are some low-texture images in the RGB/NIR dataset, which makes the homography estimation task difficult.

**PDS-COCO** artificially simulates random combined changes in brightness, contrast, saturation, and hue noise to MS-COCO dataset [10]. Similar to previous

**Table 1:** Ablation study of the degree of offset of the intra-modal self-supervised learning.

| Cross-modal \ Intra-modal | [-8, +8] | [-16, +16] | [-32, +32] |
|---|---|---|---|
| [-8, +8] | 1.60 | 1.59 | 2.21 |
| [-16, +16] | 2.33 | 2.35 | 2.96 |
| [-32, +32] | 6.29 | 5.73 | 4.35 |

approaches [5, 6, 9, 13], we resize the image to $320 \times 240$ and then construct the homography deformed data.

## 3 More Quantitative Results

### 3.1 Additional ablation study on GoogleMap Dataset

**Deeper Look at the Degree of Offset of Intra-modal Self-supervised Learning.** In real applications, the deformation degree between the cross-modal images is generally unknown. It would be interesting to alter the degree of the intra-modal self-supervised learning to evaluate their effects on the cross-modal homography estimation. To this end, we train our SCPNet under the cross-modal offset of [-8, +8], [-16, +16], and [-32, +32] with each of their corresponding intra-modal self-supervised learning having different ranges of offset under [-8, +8], [-16, +16], and [-32, +32]. The results are listed in Table 1. As expected, even if there is a significant deviation from the actual cross-modal offset degrees, the intra-modal self-supervised learning is continuously effective on the cross-modal homography estimation. The above results further demonstrate the powerful capability and generalization ability of our intra-modal self-supervised learning.

**Unsupervised loss functions.** We compare the cross-modal intensity based loss in SCPNet (Eq. 9 in the main text) and the triplet loss in CA-UDHN [12]. The quantitative results are listed in Table. 2. The performance using triplet loss is similar to our cross-modal intensity based loss. Both the two losses aim to minimize the distance between $\mathbf{P}_{\mathrm{A}}$ (anchor) and $\mathbf{P}_{\mathrm{B}}^{'}$ (positive), while maximizing the distance between $\mathbf{P}_{\mathrm{A}}$ and $\mathbf{P}_{\mathrm{B}}$ (negative). We note that our cross-modal intensity based loss can omit the margin in triplet loss, which avoids the hyperparameter tuning.

**Table 2:** Comparison of unsupervised loss functions.

| Loss function | Cross-modal intensity based loss | Triplet loss |
|---|---|---|
| MACE ↓ | 4.35 | 4.50 |

**Table 3:** Cross dataset evaluation MACEs of SCPNet.

| Train \ Test | GoogleMap | Flash/no-flash | Harvard | RGB/NIR | PDS-COCO |
|---|---|---|---|---|---|
| GoogleMap | 4.35 | 3.84 | 6.01 | 4.39 | 1.80 |
| Flash/no-flash | 16.95 | 2.67 | 5.04 | 5.27 | 1.36 |
| Harvard | 20.22 | 2.30 | 4.00 | 4.49 | 1.40 |
| RGB/NIR | 22.84 | 2.52 | 4.99 | 4.78 | 1.25 |
| PDS-COCO | 22.84 | 2.87 | 5.62 | 5.03 | 1.09 |

### 3.2 Cross Dataset Evaluation

We further conduct a cross dataset evaluation of our SCPNet, and the results are listed in Table 3. It is observed that the model trained on challenging datasets, such as GoogleMap, has superior generalization ability. This not only demonstrates the robustness of our SCPNet but also highlights its great potential for real-world applications.

## 4 More Visualization Results

We further show more visualization results, including the comparison of the consistent feature maps produced by concatenation and correlation, as well as qualitative homography estimation.

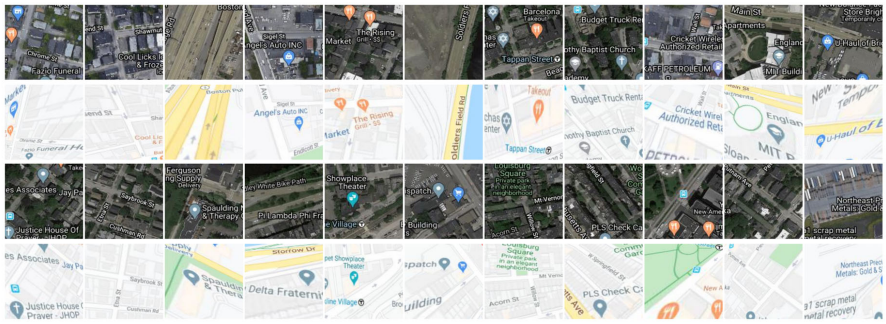
**Consistent feature map**. We illustrate more visualization results of consistent feature maps in Fig. 4. It can be observed that the feature maps generated by the correlation-based homography estimation network have salient structures and rich details, which further improves the homography estimation accuracy. However, the feature maps generated by the concatenation-based homography estimation network are blurry and ambiguous.

**Homography Estimation**. Fig. 5 visualizes more results of homography estimation. It can be seen that our SCPNet can produce accurate homography predictions in a variety of scenes across different datasets, while the comparison approaches cannot.

## References

1. Brown, M., Süsstrunk, S.: Multi-spectral SIFT for scene category recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 177–184 (2011)
2. Cao, S.Y., Hu, J., Sheng, Z., Shen, H.L.: Iterative deep homography estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1879–1888 (2022)

3. Cao, S.Y., Zhang, R., Luo, L., Yu, B., Sheng, Z., Li, J., Shen, H.L.: Recurrent homography estimation using homography-guided image warping and focus transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9833–9842 (2023)
4. Chakrabarti, A., Zickler, T.: Statistics of real-world hyperspectral images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 193–200 (2011)
5. Chang, C.H., Chou, C.N., Chang, E.Y.: CLKN: Cascaded lucas-kanade networks for image alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2213–2221 (2017)
6. DeTone, D., Malisiewicz, T., Rabinovich, A.: Deep image homography estimation. arXiv preprint arXiv:1606.03798 (2016)
7. He, S., Lau, R.W.: Saliency detection with flash and no-flash image pairs. In: Proceedings of the European Conference on Computer Vision. pp. 110–124. Springer (2014)
8. Koguciuk, D., Arani, E., Zonooz, B.: Perceptual loss for robust unsupervised homography estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4274–4283 (2021)
9. Le, H., Liu, F., Zhang, S., Agarwala, A.: Deep homography estimation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7652–7661 (2020)
10. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Proceedings of the European Conference on Computer Vision. pp. 740–755. Springer (2014)
11. Shao, R., Wu, G., Zhou, Y., Fu, Y., Fang, L., Liu, Y.: LocalTrans: A multiscale local transformer network for cross-resolution homography estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14890–14899 (2021)
12. Zhang, J., Wang, C., Liu, S., Jia, L., Ye, N., Wang, J., Zhou, J., Sun, J.: Content-aware unsupervised deep homography estimation. In: Proceedings of the European Conference on Computer Vision. pp. 653–669. Springer (2020)
13. Zhao, Y., Huang, X., Zhang, Z.: Deep Lucas-Kanade homography for multimodal image alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15950–15959 (2021)



(a) GoogleMap

(b) Flash/no-flash

(c) Harvard

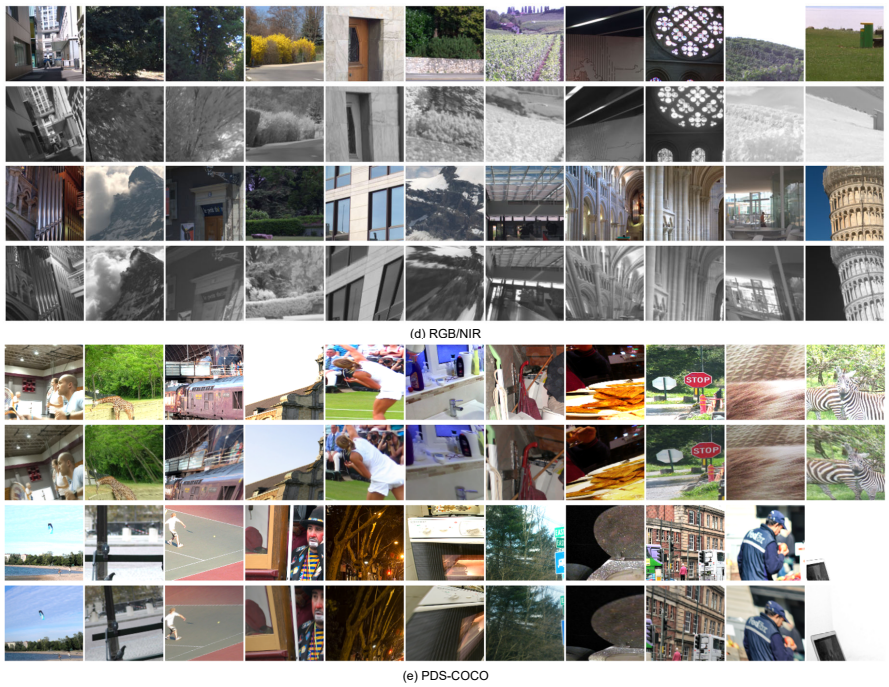
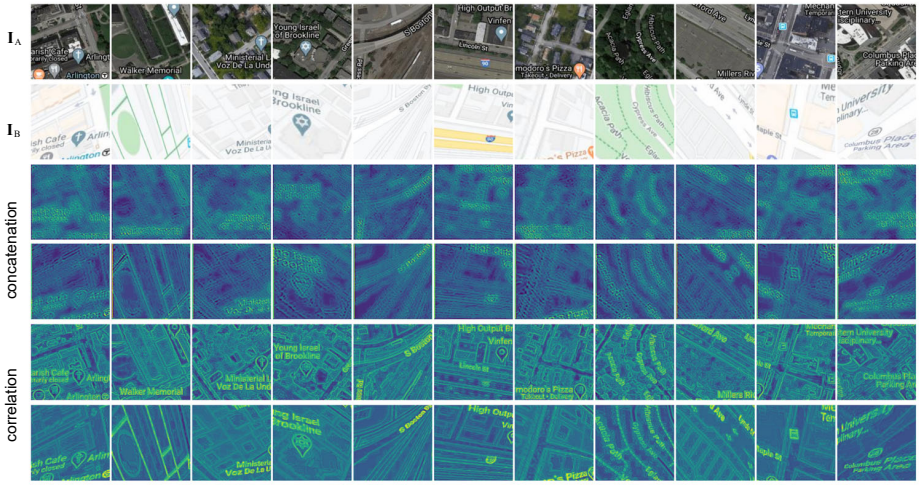
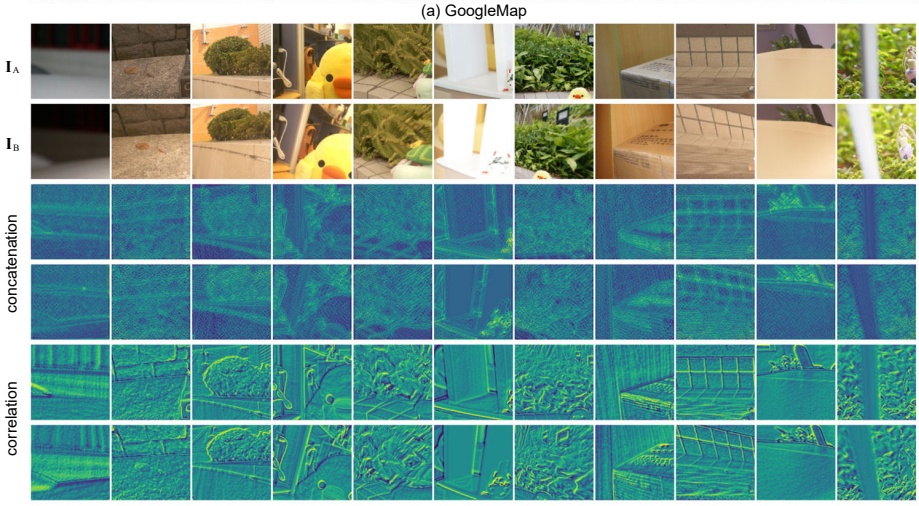



**Fig. 3:** Example images of the cross-modal datasets including GoogleMap and Flash/no-flash, cross-spetral datasets including Harvard and RGB/NIR, together with the manually-made inconsistent dataset PDS-COCO respectively, under [-32,+32] offset.




(a) GoogleMap




(b) Flash/no-flash

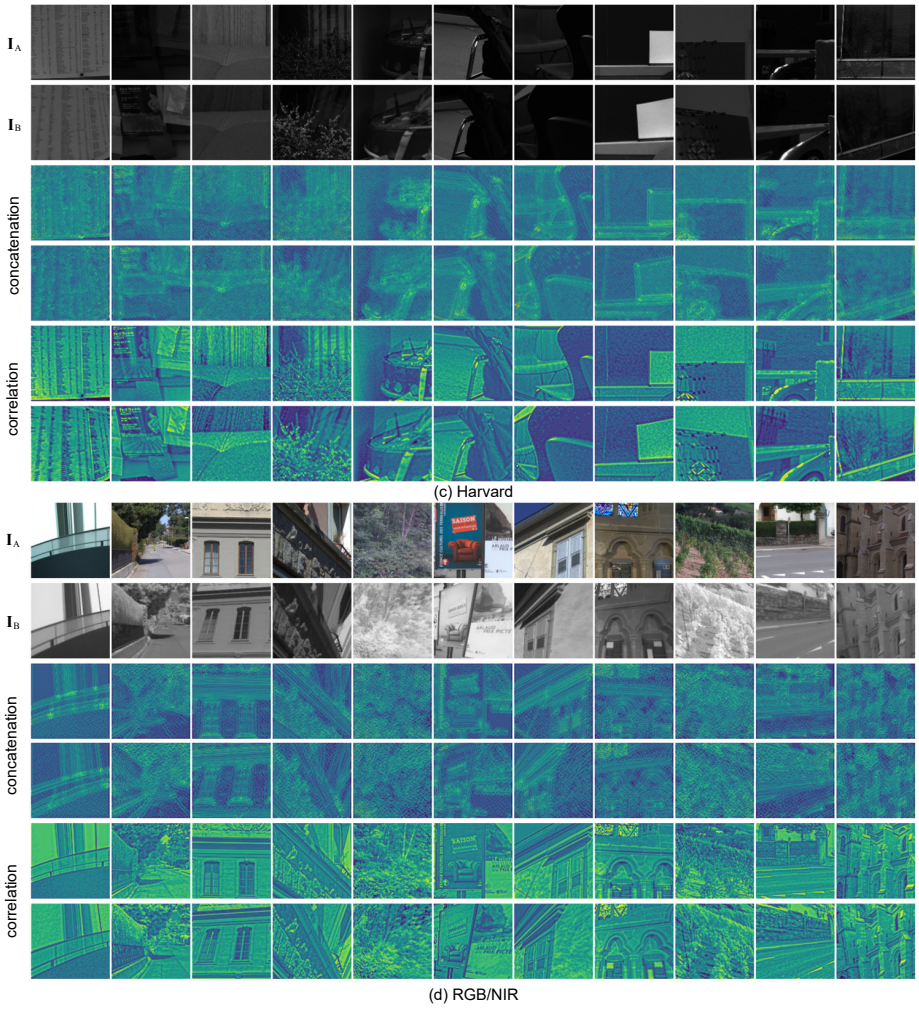
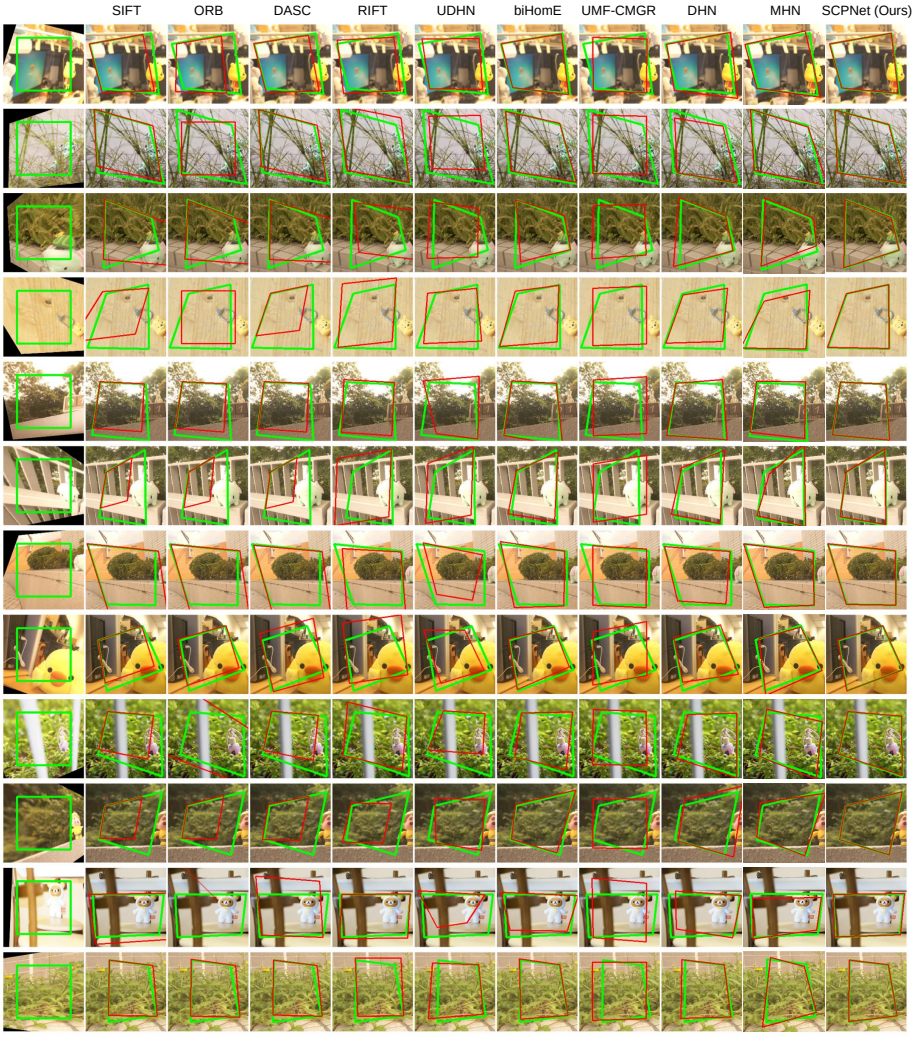



**Fig. 4:** Comparison of the consistent feature maps produced by concatenation and correlation on GoogleMap, Flash/no-flash, Harvard, and RGB/NIR datasets respectively, under [-32,+32] offset.


(a) GoogleMap




(b) Flash/no-flash


(c) Harvard


**Fig. 5:** Qualitative homography estimation results on GoogleMap, Flash/no-flash, Harvard, and RGB/NIR datasets respectively, under [-32,+32] offset. Green polygons denote the ground-truth homography deformation from $\mathbf{I}_{\mathrm{B}}$ (source, the deformed image) to $\mathbf{I}_{\mathrm{A}}$ (target). Red polygons denote the estimated homography deformation using different algorithms on $\mathbf{I}_{\mathrm{A}}$ (target).