Iterative Deep Homography Estimation
Supplementary Materials

Si-Yuan Cao Jianxin Hu Zehua Sheng Hui-Liang Shen
Zhejiang University
karlcao@hotmail.com, {hujianxin,shengzehua,shenhl}@zju.edu.cn

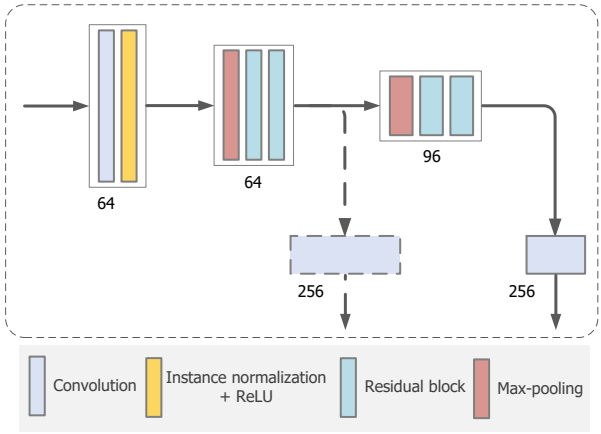

Figure 1. CNN feature extractor in iterative homography network (IHN). The solid lines denote the branch for the feature map of basic IHN, and the dashed lines denote the branch for the extra scale of 2-scale IHN. The numbers denote the number of filters of convolutional kernels.

1. Details of Network

We further illustrate the details of our IHN in this Section. The details include structure of feature extractor, details of correlation pooling, convolution of correlation slice and homography flow, and parameterization of homography matrix.

1.1. Structure of Feature Extractor

As illustrated in Fig. 1, we use 2 basic units containing 1 max-pooing and 2 residual blocks to extract the feature map for computing correlation. The image are first processed by 1 convolutional block with kernel size 7×7. The $1/2 \times 1/2$ and $1/4 \times 1/4$ resolution feature maps are successively produced by the 2 basic units. The feature maps are finally reprojected by 1 linear convolutional layer with kernel 1×1. The solid lines denote the branch for the feature map of basic IHN, and the dashed lines denote the branch for the extra scale of 2-scale IHN. The convolutional kernel size is set to 3×3 if not specifically mentioned.

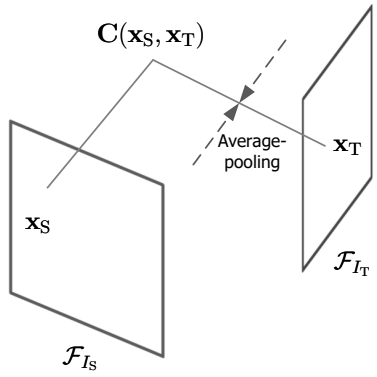

Figure 2. Average-pooling of the correlation volume.

1.2. Details of Correlation Pooling

As illustrated in Fig. 2, the correlation volume $\mathbf{C}$ is computed using the feature maps of the source image $\mathcal{F}_{I_\mathrm{S}}$ and the target image $\mathcal{F}_{I_\mathrm{T}}$. The average-pooling is conducted on the coordinate dimension of the feature map of the target image, namely the last 2 dimensions of $\mathbf{C}$. $\mathbf{C}$ of size $H \times W \times H \times W$ is made into $\mathbf{C}^{\frac{1}{2}}$ of size $H \times W \times H/2 \times W/2$ after the average-pooling.

1.3. Convolution of Correlation Slice and Homography Flow

At the beginning, the sampled correlation slice $\mathbf{S}$ is of size $H \times W \times (2r+1) \times (2r+1)$. To enable the 2D convolution, $\mathbf{S}$ is reshaped into $(2r+1)(2r+1) \times H \times W$. The homography flow $\mathbf{F}$ is of size $2 \times H \times W$, and thus can be concatenated with the reshaped $\mathbf{S}$ in the channel dimension. The concatenated feature map is then processed by the global motion aggregator (GMA) to estimate the residual homography.

1.4. Parameterization of Homography Matrix

We parameterize the homography matrix using the displacement vectors of the 4 corner points of the image. Taking the least square method for example, given 2 sets of corner points, the homography matrix can be obtained by

solving the least squares problem,

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

where $\mathbf{A}$ and $\mathbf{b}$ denote the reformed coordinates of the 4 corner points of the source image I_{S} and the target image I_{T}, $\mathbf{h}$ the reformed homography matrix. Let us take 1 corner point $\mathbf{x}_1 = (u_1, v_1)$ in I_{S} for example, its corresponding corner point $\mathbf{x'}_1 = (u'_1, v'_1)$ in I_{T} can be projected by the homography matrix,

$$\begin{aligned} u'_1 &= \frac{\mathbf{H}_{11}u_1 + \mathbf{H}_{12}v_1 + \mathbf{H}_{13}}{\mathbf{H}_{31}u_1 + \mathbf{H}_{32}v_1 + 1} \\ v'_1 &= \frac{\mathbf{H}_{21}u_1 + \mathbf{H}_{22}v_1 + \mathbf{H}_{23}}{\mathbf{H}_{31}u_1 + \mathbf{H}_{32}v_1 + 1}. \end{aligned} \tag{2}$$

We multiply both sides of the equation by the right side numerator and have

$$\begin{aligned} u'_1 &= \mathbf{H}_{11}u_1 + \mathbf{H}_{12}v_1 + \mathbf{H}_{13} - \mathbf{H}_{31}u_1u'_1 - \mathbf{H}_{32}v_1u'_1 \\ v'_1 &= \mathbf{H}_{21}u_1 + \mathbf{H}_{22}v_1 + \mathbf{H}_{23} - \mathbf{H}_{31}u_1v'_1 - \mathbf{H}_{32}v_1v'_1. \end{aligned} \tag{3}$$

The above multivariate equation of the elements of $\mathbf{H}$ can be solved using least squares with more than 4 corner points known. We rewrite $\mathbf{H}$ into its vector form as

$$\mathbf{h} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} & \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} & \mathbf{H}_{31} & \mathbf{H}_{32} \end{bmatrix}^{\mathsf{T}}, \tag{4}$$

and accordingly $\mathbf{A}$ becomes

$$\mathbf{A} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1u'_1 & -v_1u'_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1v'_1 & -v_1v'_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2u'_2 & -v_2u'_2 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2v'_2 & -v_2v'_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3u'_3 & -v_3u'_3 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3v'_3 & -v_3v'_3 \\ u_4 & v_4 & 1 & 0 & 0 & 0 & -u_4u'_4 & -v_4u'_4 \\ 0 & 0 & 0 & u_4 & v_4 & 1 & -u_4v'_4 & -v_4v'_4 \end{bmatrix}, \tag{5}$$

where $(u_1, v_1) \sim (u_4, v_4)$ denote the 4 corner points of the image. We form $\mathbf{b}$ as

$$\mathbf{b} = \begin{bmatrix} u'_1 & v'_1 & u'_2 & v'_2 & u'_3 & v'_3 & u'_4 & v'_4 \end{bmatrix}^{\mathsf{T}}. \tag{6}$$

We note that the corner points of I_{S} and I_{T} are related by the displacement cube $\mathbf{D}$ produced by our IHN, which can be formulated as

$$\begin{aligned} u'_1 &= u_1 + \mathbf{D}(0,0,0) \\ v'_1 &= v_1 + \mathbf{D}(1,0,0) \\ u'_2 &= u_2 + \mathbf{D}(0,0,1) \\ v'_2 &= v_2 + \mathbf{D}(1,0,1) \\ u'_3 &= u_3 + \mathbf{D}(0,1,0) \\ v'_3 &= v_3 + \mathbf{D}(1,1,0) \\ u'_4 &= u_4 + \mathbf{D}(0,1,1) \\ v'_4 &= v_4 + \mathbf{D}(1,1,1). \end{aligned} \tag{7}$$

The homography matrix can also be solved by the direct linear transform (DLT) [1] or other methods.

2. Preparation of Datasets and More Experimental Results

2.1. Preparation of Datasets

We illustrate the example images of MSCOCO [7], Google Earth [11], Google Map & Satellite, and SPID [9] in Fig. 3. The size of input image of IHN for all datasets is set to 128×128. MSCOCO contains everyday RGB images, Google Earth contains cross-season images, Google Map & Satellite contains cross-modal images, and SPID contains images with moving objects. Detailed processing for the above datasets is as follow.

MSCOCO. We process MSCOCO [7] images in the same way as the data processing in [3, 4, 6, 11]. The images are first resized to 320×240. An 128×128 image pair related by a simulated homography is cropped from the resized image. The homography is produced by randomly perturbing the corners of 128×128 image with the maximum range of [-32,32]. We note that our feature extractor of IHN can process RGB inputs, while [4, 6] that use image concatenation strategy can only process the grayscale image. Another difference is that previous iterative methods using the IC-LK iterator [3, 11] need to expand the boarders of the target image by 32 pixels, making it of size 192×192. On the contrary, IHN doesn't require this specific operation, which means IHN use less image information than [3, 11]. We train and test IHN and 2-scale IHN on MSCOCO 2017 using the provided training and test data.

Google Earth and Google Map & Satellite. We use the same image processing method as in [11], in which those 2 datasets are proposed. As the perturbations of the test data are provided in the datasets, we test IHN on exactly the same homographies and cropping positions as in [11]. We note that IHN only needs 128×128 target image, and hence the provided 192×192 target image is center cropped to 128×128.

SPID. After the 2 different ways of processing, images in SPID are uniformly resized into 220×220. The image pair of 128×128 related by the [-32,32] perturbation homography is then cropped. We test all the methods under exactly the same simulated homographies and cropping positions.

2.2. More Experimental Results of the Iterative Process

ACE at Each Iteration. We illustrate more experimental results of visualization of homography estimation with average corner error (ACE) at each iteration in Fig. 4. The results include our IHN and DLKFM [11] (which use the traditional IC-LK iterator) on Google Earth and Google Map & Satellite datasets. It is observed that our 1-scale IHN can provide stable ACE reduction and produce accurate homography estimation results during the iterative process.

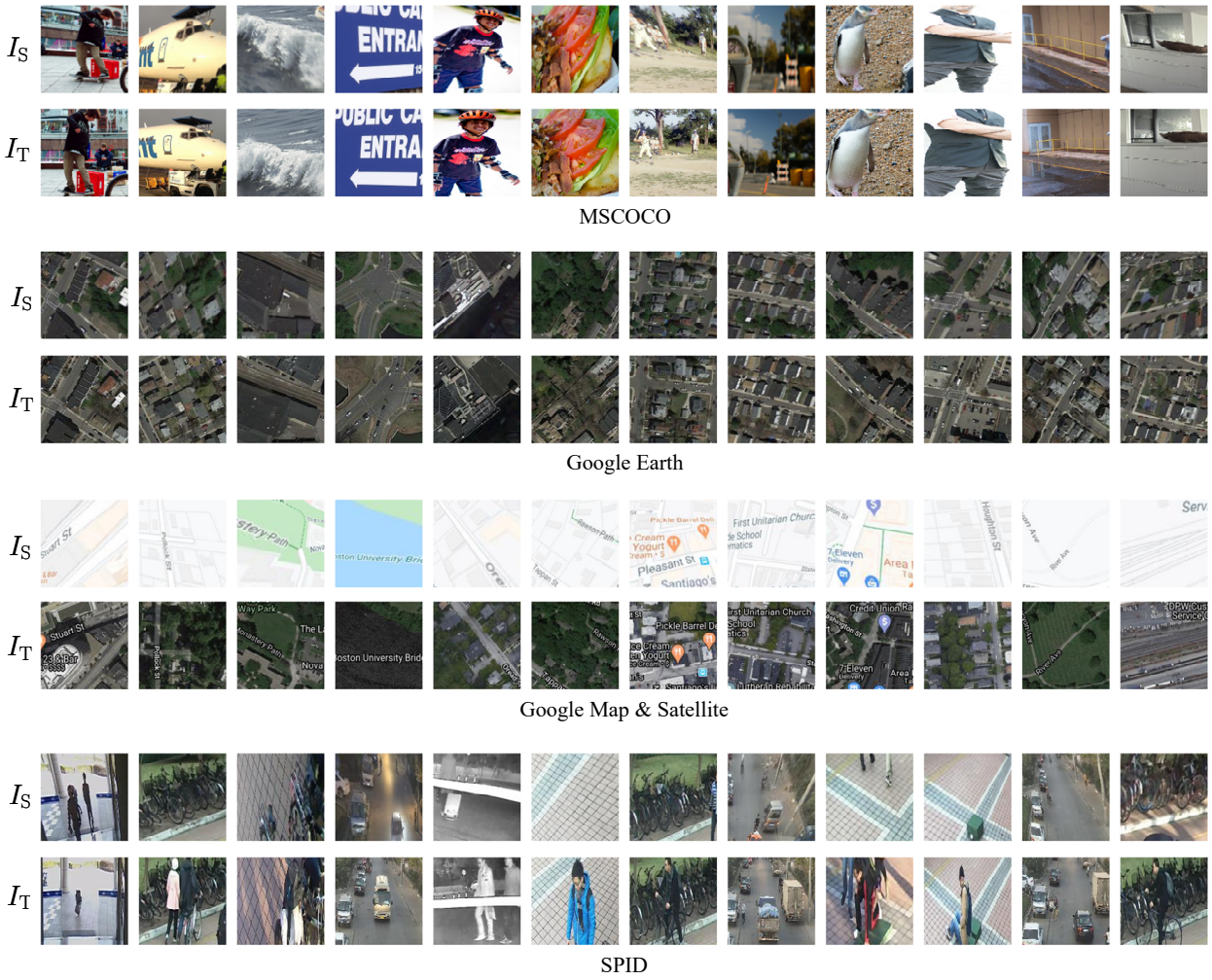

Figure 3. Example images of MSCOCO, Google Earth, Google Map & Satellite, and SPID datasets. MSCOCO contains everyday RGB images, Google Earth contains cross-season images, Google Map & Satellite contains cross-modal images, and SPID contains images with moving objects. I_S and I_T denote the source image and the target image, and I_S is perturbed by the simulated homography.

Weight Mask at Each Iteration. We note that the inlier masks produced by 1-scale IHN-mov are different during the iterative process. Fig. 5 illustrates the inlier mask at each iteration. The difference image calculates the difference of the source image I_S and the warped target image $I_{T,W}$, and it is reversed for roughly displaying the matching inliers. It is observed that the mask can produce a more accurate weighting for the matching inliers as the number of iteration grows.

2.3. More Experimental Results on SPID

We illustrate more experimental results of visualization of homography estimation on SPID [9]. The results of SIFT+RANSAC [5, 8], SIFT+MAGSAC [2], DHN [4], MHN [6], UDHN [10], 1-scale IHN, and 1-scale IHN-mov are displayed. It is observed that SIFT+RANSAC and SIFT+MAGSAC hardly produce satisfactory results. For deep homography methods, DHN, MHN, UDHN, and our 1-scale IHN are affected by the foreground moving-objects, while 1-scale IHN-mov is more robust and produces lower ACEs.

References

[1] Yousset I Abdel-Aziz, HM Karara, and Michael Hauck. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogrammetric Engineering & Remote Sensing*, 81(2):103–107, 2015. 2

[2] Daniel Barath, Jiri Matas, and Jana Noskova. MAGSAC: marginalizing sample consensus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10197–10205, 2019. 3

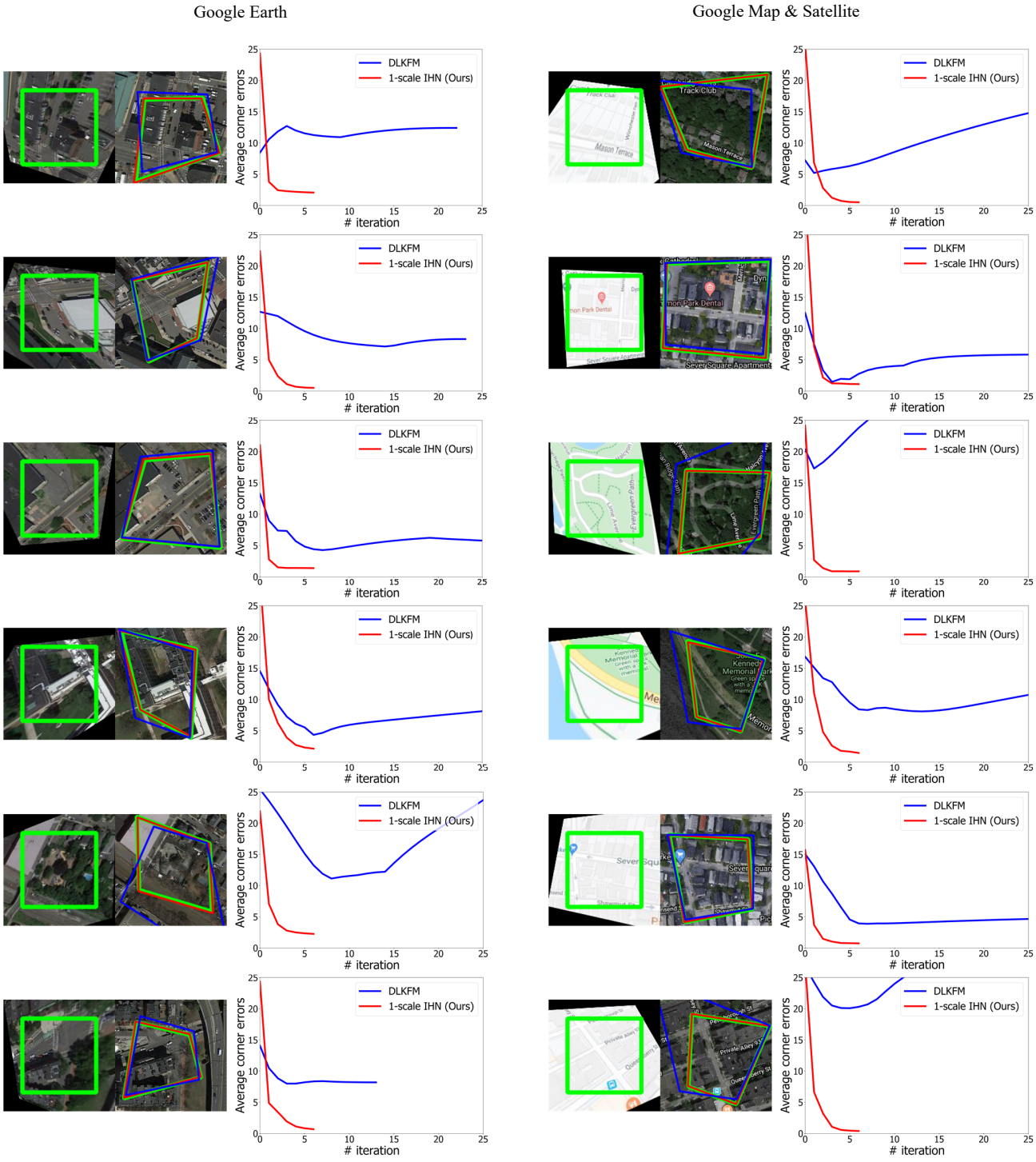

Figure 4. More experimental results of visualization of homography estimation and average corner error (ACE) at each iteration. The results include our IHN and DLKFM [11] (which use the traditional IC-LK iterator) on Google Earth and Google Map & Satellite datasets. Left 2 images for each dataset: image pair for homography estimation with the source image I_{S} on the left and the target image I_{T} on the right. Green polygons denote the ground-truth position of I_{S} on I_{T}. Blue polygons denote the estimated position using MHN+DLKFM. Red polygons denote the estimated position using our IHN. Right plot for each dataset: ACEs during first 12 iterations. IHN stops at iteration 6 while DLKFM has a dynamic stop criterion which iterates 21 times averagely.

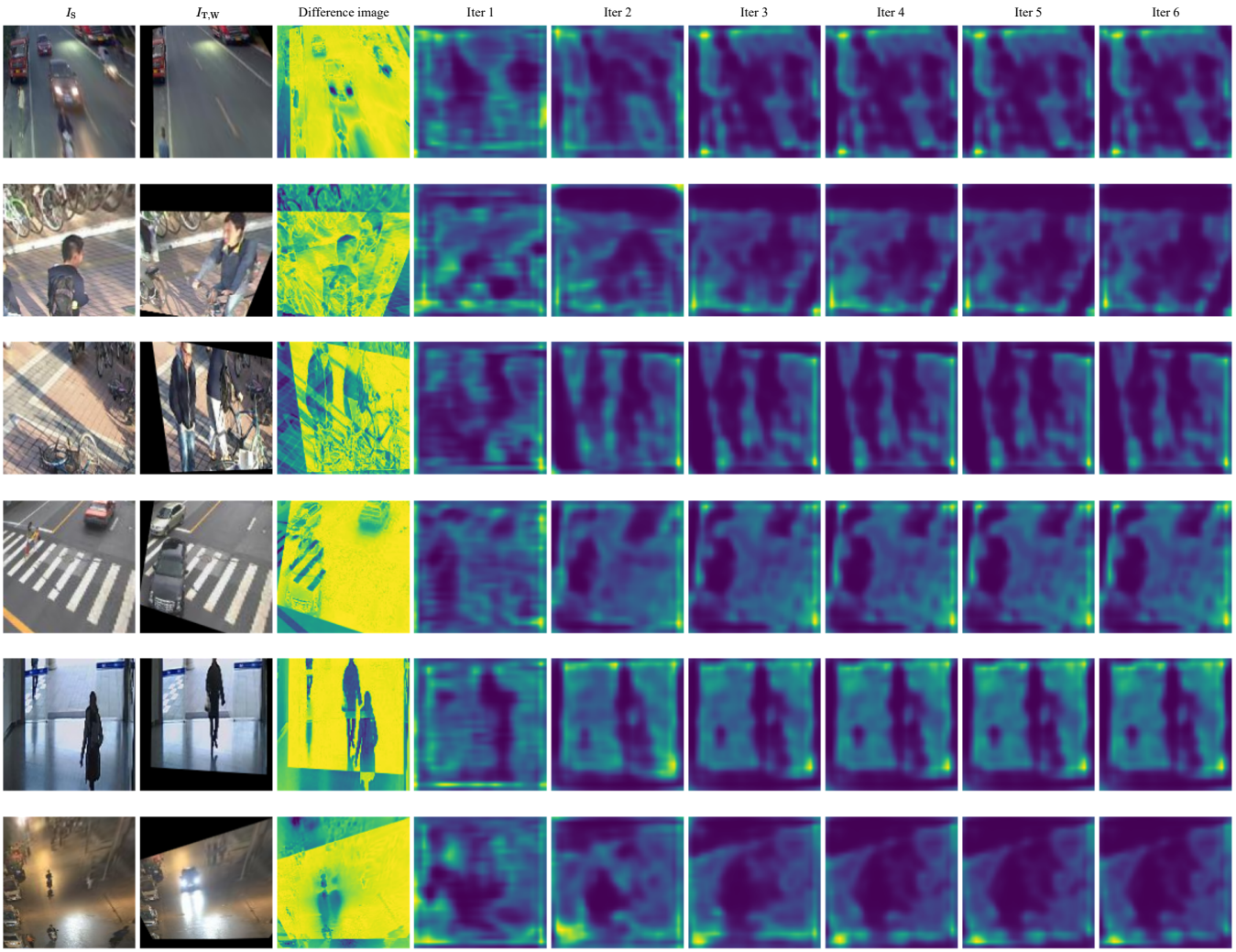

Figure 5. Inlier mask produced by 1-scale IHN-mov at each iteration. I_S denotes the source image and $I_{T,W}$ the warped target image. Iter 1 $\sim$ Iter 6 denote the number of iterations.

[3] 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. 2

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 2, 3

[5] 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. 3

[6] 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*, pages 7652–7661, 2020. 2, 3

[7] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 2

[8] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3

[9] Dan Wang, Chongyang Zhang, Hao Cheng, Yanfeng Shang, and Lin Mei. SPID: surveillance pedestrian image dataset and performance evaluation for pedestrian detection. In *Asian Conference on Computer Vision*, pages 463–477. Springer, 2016. 2, 3

[10] 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. Springer, 2020. 3

[11] Yiming Zhao, Xinming Huang, and Ziming Zhang. Deep Lucas-Kanade homography for multimodal image alignment. In *Proceedings of the IEEE/CVF Conference on Com-*

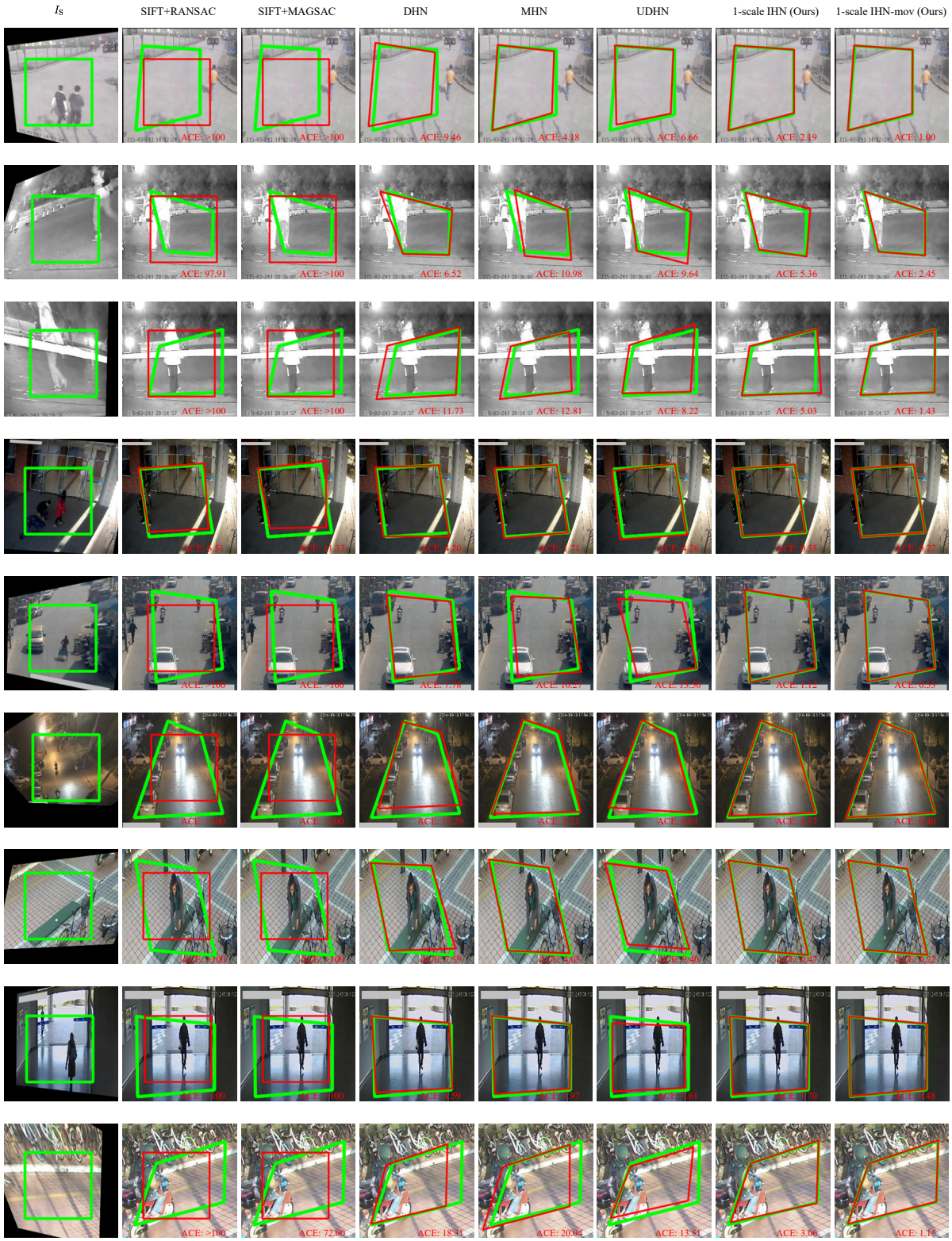

Figure 6. More experimental results of visualization of homography estimation on the SPID dataset with moving objects. Green polygons denote the ground-truth position of I_S on the target images. Red polygons denote the estimated position using different algorithms on the target images. The closer the 2 colors of polygons are, the better estimation accuracy (also indicated by a lower ACE).

puter Vision and Pattern Recognition, pages 15950–15959, 2021. 2, 4