Towards Generalized Multimodal Homography Estimation

Supplementary Material

1. Preliminaries

This section presents the fundamentals of the homography matrix, homography transformation, and mean average corner error.

1.1. Homography Matrix and Transformation

A homography matrix is represented as a 3×3 matrix and can be formulated as

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}. \tag{19}$$

Given two matching points $p_{src} = (x_{src}, y_{src}) \in \mathbb{R}^2$ and $p_{tar} = (x_{tar}, y_{tar}) \in \mathbb{R}^2$ in the source and target images, we assume that H describes the perspective transformation from p_{src} to p_{tar}. The transformed point $p'_{tar} = (x'_{tar}, y'_{tar})$ can be mathematically expressed as

$$\begin{aligned} x'_{tar} &= \frac{h_{11} \cdot x_{src} + h_{12} \cdot y_{src} + h_{13}}{h_{31} \cdot x_{src} + h_{32} \cdot y_{src} + 1}, \\ y'_{tar} &= \frac{h_{21} \cdot x_{src} + h_{22} \cdot y_{src} + h_{23}}{h_{31} \cdot x_{src} + h_{32} \cdot y_{src} + 1}. \end{aligned} \tag{20}$$

The point p'_{tar} is equal to p_{tar} if the homography matrix H is perfectly estimated. From Eqs. (19) and (20), we see that the homography matrix can be derived using four point pairs. Typically, the four corner points of the source image are selected. We denote these points as p^1_{src} , p^2_{src}, p^3_{src}, and p^4_{src}. Their corresponding points in the target image are denoted as p^1_{tar} , p^2_{tar}, p^3_{tar}, and p^4_{tar}. Assuming the source image has a size of $S \times S$, its corner points are defined as $p^1_{src} = [0, 0]^T$, $p^2_{src} = [0, S]^T$, $p^3_{src} = [S, 0]^T$, and $p^4_{src} = [S, S]^T$. p^1_{tar} , p^2_{tar}, p^3_{tar}, and p^4_{tar} are predicted by

$$\begin{aligned} p^1_{tar} &= p^1_{src} + O_1, \\ p^2_{tar} &= p^2_{src} + O_2, \\ p^3_{tar} &= p^3_{src} + O_3, \\ p^4_{tar} &= p^4_{src} + O_4, \end{aligned} \tag{21}$$

where O_1, O_2, O_3, $O_4 \in \mathbb{R}^2$ are the position offsets between the matching points and are predicted by the deep-learning homography estimation models. Subsequently, the homography matrix H is calculated by solving the least squares problem as follows:

$$Ah = b, \tag{22}$$

where

$$A = \begin{bmatrix} x^1_{src} & v^1_{src} & 1 & 0 & 0 & 0 & -x^1_{src}x^1_{tar} & -y^1_{src}x^1_{tar} \\ 0 & 0 & 0 & x^1_{src} & v^1_{src} & 1 & -x^1_{src}y^1_{tar} & -y^1_{src}y^1_{tar} \\ x^2_{src} & v^2_{src} & 1 & 0 & 0 & 0 & -x^2_{src}x^2_{tar} & -y^1_{src}x^2_{tar} \\ 0 & 0 & 0 & x^2_{src} & v^2_{src} & 1 & -x^2_{src}y^2_{tar} & -y^1_{src}y^2_{tar} \\ x^3_{src} & v^3_{src} & 1 & 0 & 0 & 0 & -x^3_{src}x^3_{tar} & -y^3_{src}x^3_{tar} \\ 0 & 0 & 0 & x^3_{src} & v^3_{src} & 1 & -x^3_{src}y^3_{tar} & -y^3_{src}y^3_{tar} \\ x^4_{src} & v^4_{src} & 1 & 0 & 0 & 0 & -x^4_{src}x^4_{tar} & -y^4_{src}x^4_{tar} \\ 0 & 0 & 0 & x^4_{src} & v^4_{src} & 1 & -x^4_{src}y^4_{tar} & -y^4_{src}y^4_{tar} \end{bmatrix}, \tag{23}$$

$$h = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}]^T, \tag{24}$$

$$b = [x^1_{tar}, y^1_{tar}, x^2_{tar}, y^2_{tar}, x^3_{tar}, y^3_{tar}, x^4_{tar}, y^4_{tar}]^T. \tag{25}$$

In this manner, the homography matrix can be efficiently estimated, enabling the transformation between two images.

1.2. Mean Average Corner Error

The mean average corner error (MACE) is used to evaluate estimation accuracy. It is calculated based on the offsets of the four corner points as follows:

$$MACE = \frac{1}{4} \sum_{i=1}^{4} \|O^{gt}_i - O_i\|_2, \tag{26}$$

where O^{gt}_i and O_i denote the ground-truth and predicted offsets as defined in Eq. (21); $\|\cdot\|_2$ represents the Euclidean distance.

2. Ablation Studies

This section provides ablation studies for the proposed training data synthesis method and cross-scale color-invariant network (CCNet), along with more visualization results for both within-dataset and zero-shot evaluations.

2.1. Training Data Synthesis

Range of Content Weight. Our data synthesis method uses the coefficients $1-\alpha_i$ and $1-\alpha_j$, defined in Eqs. (4) and (5), to control the intensity of the rendered textures and colors. A larger value indicates greater intensity. In our method, $1 - \alpha_i$ and $1 - \alpha_j$ are drawn from a uniform distribution $\mathcal{U}(0, \alpha)$, where α represents the maximum possible value and is set to 1. We investigate the zero-shot performance across different values of α. The results are presented in Tab. 7. The average values of MACE are 12.382, 6.311, 5.197, 4.861, 4.112, and 3.414 as α increases. Overall, generalization performance improves with higher α, as greater variations in textures and colors enhance robustness against modality changes. When α is zero, the training data lacks

diverse textures and colors, leading to significant degradation in generalization performance. Notably, performance on GoogleEarth tends to decrease as α increases. This is because the paired images in GoogleEarth are captured with the same device across different seasons, resulting in smaller modality differences compared to GoogleMap and RGB-NIR. A smaller α yields minimal changes in textures and colors, which aligns well with the characteristics of GoogleEarth. Conversely, a larger α induces more substantial changes, thereby degrading performance on GoogleEarth but enhancing accuracy on GoogleMap and RGB-NIR.

Random Sampling. The coefficients $1-\alpha_i$ and $1-\alpha_j$ in Eqs. (4) and (5) are randomly sampled from the uniform distribution $\mathcal{U}(0,1)$. We investigate the performance difference between using these randomly sampled values and fixed values of 1. Tab. 8 demonstrates the results. The mean value of MACE increases from 3.414 to 3.539 when random sampling is not applied. This indicates a decrease in accuracy. The random sampling introduces greater diversity in the textures and colors, thus enhancing generalization performance.

Range of Smoothing Weight. Our method employs the smoothing weights β_i and β_j defined in Eq. (6) to control texture smoothness. Higher values indicate greater smoothness. The weights β_i and β_j are drawn from a uniform distribution $\mathcal{U}(0,\beta)$, where β represents the maximum possible value. We investigate the performance with varying values of β. Tab. 9 shows the results. The smoothing operation is disabled when β is set to 0. As β increases from 0 to 0.001, the mean value of MACE on the three datasets decreases from 3.503 to 3.414, indicating improved accuracy. This enhancement in texture smoothness positively contributes to generalization performance. However, if β becomes too large, MACE increases due to excessive smoothing, which can lead to a significant loss of structural information.

Number of Template Images. Our training data synthesis method leverages multiple template images to render the same content image with a variety of textures and colors. We examine the impact of the number of template images on generalization performance. Tab. 10 presents the results. The rendering strategy is not applied when M is set to 0. As observed, homography estimation performance generally improves with an increase in M. Specifically, the average value of MACE on the three datasets decreases from 12.382 to 3.414 as M increases from 0 to 1000. Notably, generalization performance is considerably poor when the rendering strategy is not utilized (i.e., $M=0$). This demonstrates that our synthesis method effectively enhances estimation robustness against various modalities.

Table 7. Zero-shot performance with various values of α for $1-\alpha_i \sim \mathcal{U}(0,\alpha)$ and $1-\alpha_j \sim \mathcal{U}(0,\alpha)$ in the proposed training data synthesis method. Evaluations are conducted using our CCNet.

α	GoogleMap	GoogleEarth	RGB-NIR
0.0	22.647	2.811	11.688
0.2	11.041	1.342	6.549
0.4	8.650	1.343	5.597
0.6	7.808	1.377	5.398
0.8	5.571	1.361	5.403
1.0	4.383	1.399	4.461

Table 8. Zero-shot performance with and without sampling content weights from the uniform distribution $\mathcal{U}(0,1)$ in the proposed training data synthesis method. In the no-sampling scenario, all content weights are set to 1. Evaluations are conducted using our CCNet. The best results are written in bold.

Random	GoogleMap	GoogleEarth	RGB-NIR
✗	**3.569**	1.888	5.159
✓	4.383	**1.399**	**4.461**

Table 9. Zero-shot performance with various values of β sampled from the uniform distribution $\mathcal{U}(0,\beta)$ for the smoothing weight in the proposed training data synthesis method. Evaluations are conducted using our CCNet. The best results are written in bold.

β	GoogleMap	GoogleEarth	RGB-NIR
0.000	4.455	1.447	4.607
0.001	**4.383**	**1.399**	**4.461**
0.005	4.739	1.445	4.572
0.010	6.158	1.491	5.310

Table 10. Zero-shot performance with varying numbers of template images (M) used in the proposed training data synthesis method. Evaluations are conducted using our CCNet.

M	GoogleMap	GoogleEarth	RGB-NIR
0	22.647	2.811	11.688
1	13.406	1.392	6.755
10	4.779	1.471	4.300
100	4.541	1.397	5.148
1000	4.383	1.399	4.461

2.2. Cross-Scale and Color-Invariant Network

Modules. We conduct experiments to validate the effectiveness of the proposed cross-scale feature fusion and color decoupling modules. Tab. 11 presents the results. As observed, estimation performance declines in the absence of either the cross-scale feature fusion or the color decoupling modules. Integrating both modules allows our network to achieve better generalization performance. Therefore, the cross-scale feature fusion and color decoupling modules indeed enhance estimation accuracy.

Table 11. Ablation studies for CCNet in zero-shot evaluation. The model is trained on the data synthesized from MSCOCO. CI and CD represent the cross-scale feature fusion and color decoupling modules, respectively.

CI	CD	GoogleMap	GoogleEarth	RGB-NIR
✓		5.064	1.475	4.983
	✓	4.566	1.405	4.867
✓	✓	**4.383**	**1.399**	**4.461**

Table 12. Ablation studies for CCNet with varying numbers of scales and iterations in zero-shot evaluation. The model is trained on the data synthesized from MSCOCO. Here, S and K denote the number of scales and iterations, respectively. The default values of S and K are set to 3 and 2.

	GoogleMap	GoogleEarth	RGB-NIR
S=1	8.596	5.099	9.317
S=2	5.097	1.714	4.868
S=3	4.383	1.399	4.461
K=1	6.201	1.838	5.410

Table 13. Zero-shot performance with varying values of λ for color decoupling.

λ	GoogleMap	GoogleEarth	RGB-NIR
0.1	4.430	1.445	5.184
0.3	4.741	1.431	5.244
0.5	4.383	1.399	4.461
0.7	4.296	1.425	4.761
0.9	4.660	1.378	4.906

Number of Scales and Iterations. Our CCNet employs the iterative estimation strategy to progressively refine the predicted offsets. Tab. 12 presents the accuracy under various numbers of scales and iterations. As shown, increasing the number of scales and iterations generally results in higher accuracy.

Hyperparameter for Color Decoupling. The color decoupling is integrated into our CCNet. It relies on the loss function defined in Eq. (17) with a hyperparameter λ. We conduct experiments to evaluate accuracy across various values of λ. The results are shown in Tab. 13. The average MACE values on the three datasets are 3.686, 3.805, 3.414, 3.494, and 3.648 as λ increases. Both high and low values of λ result in performance degradation.

2.3. Visualization Results

Within-Dataset Results. Figs. 6 to 8 present additional within-dataset visualization results for GoogleMap, GoogleEarth, and RGB-NIR. As observed, our CCNet consistently outperforms the baselines in estimation accuracy.

Zero-Shot Results. Figs. 9 to 11 present additional zero-shot visualization results for GoogleMap, GoogleEarth, and RGB-NIR. As demonstrated, our CCNet consistently outperforms the baselines in estimation performance.

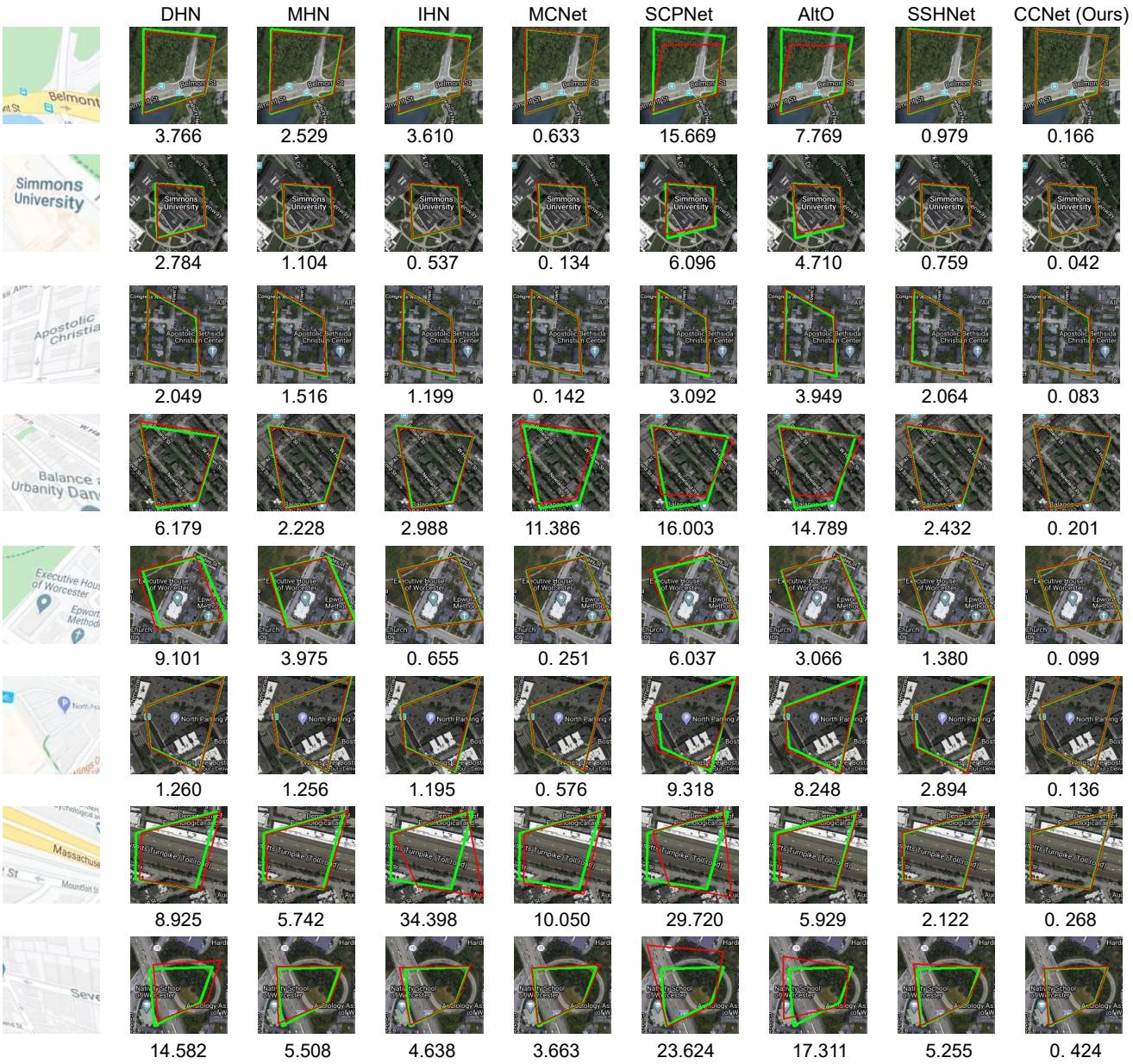

Figure 6. Visualization results of within-dataset evaluation on GoogleMap. The first column displays the source image to be warped, while the subsequent columns present the corresponding target images. Below each target image, the mean average corner error (MACE) is indicated. Greater similarity between the red and green quadrilaterals signifies higher estimation accuracy.

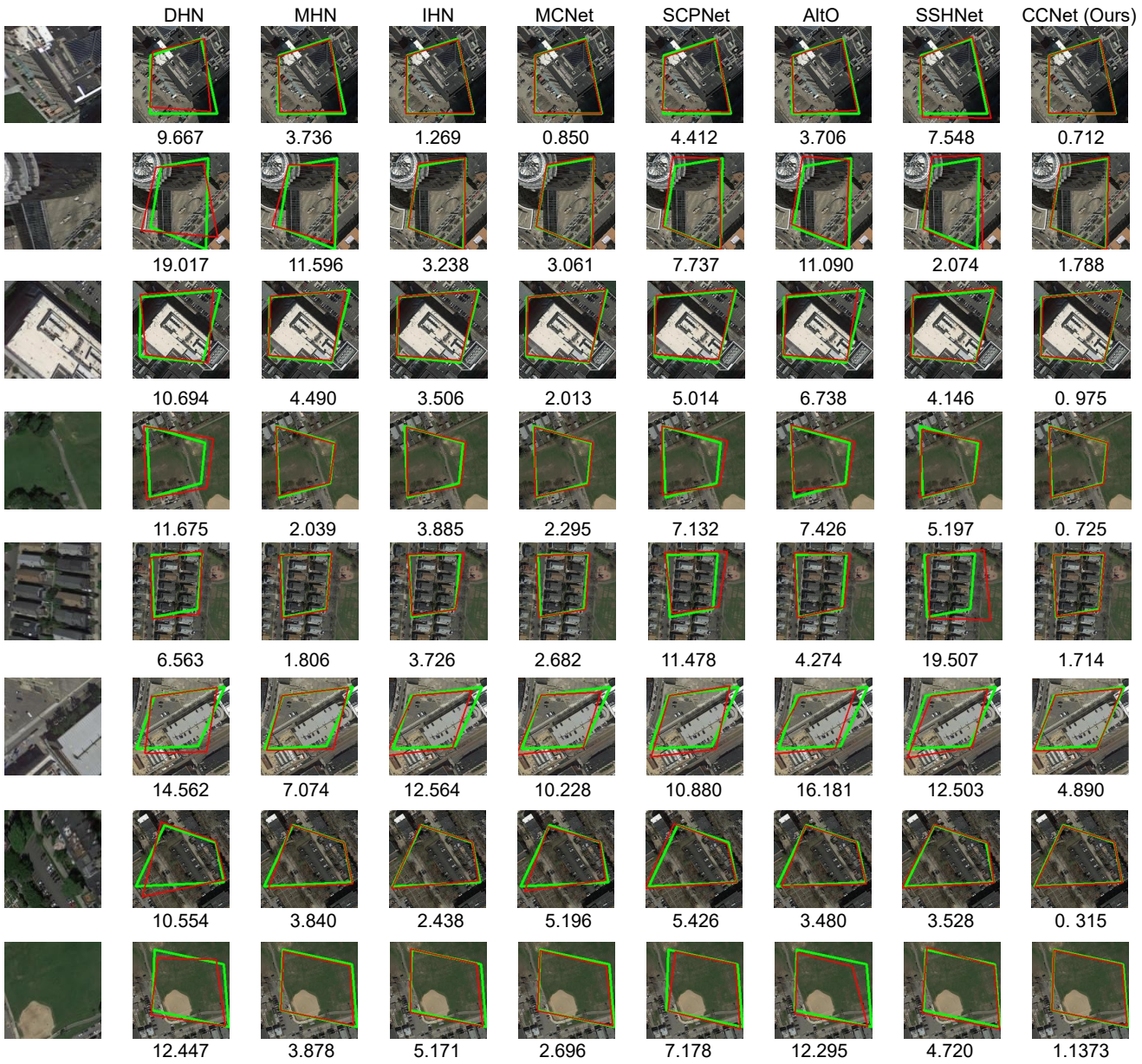

Figure 7. Visualization results of within-dataset evaluation on GoogleEarth. The first column displays the source image to be warped, while the subsequent columns present the corresponding target images. Below each target image, the mean average corner error (MACE) is indicated. Greater similarity between the red and green quadrilaterals signifies higher estimation accuracy.

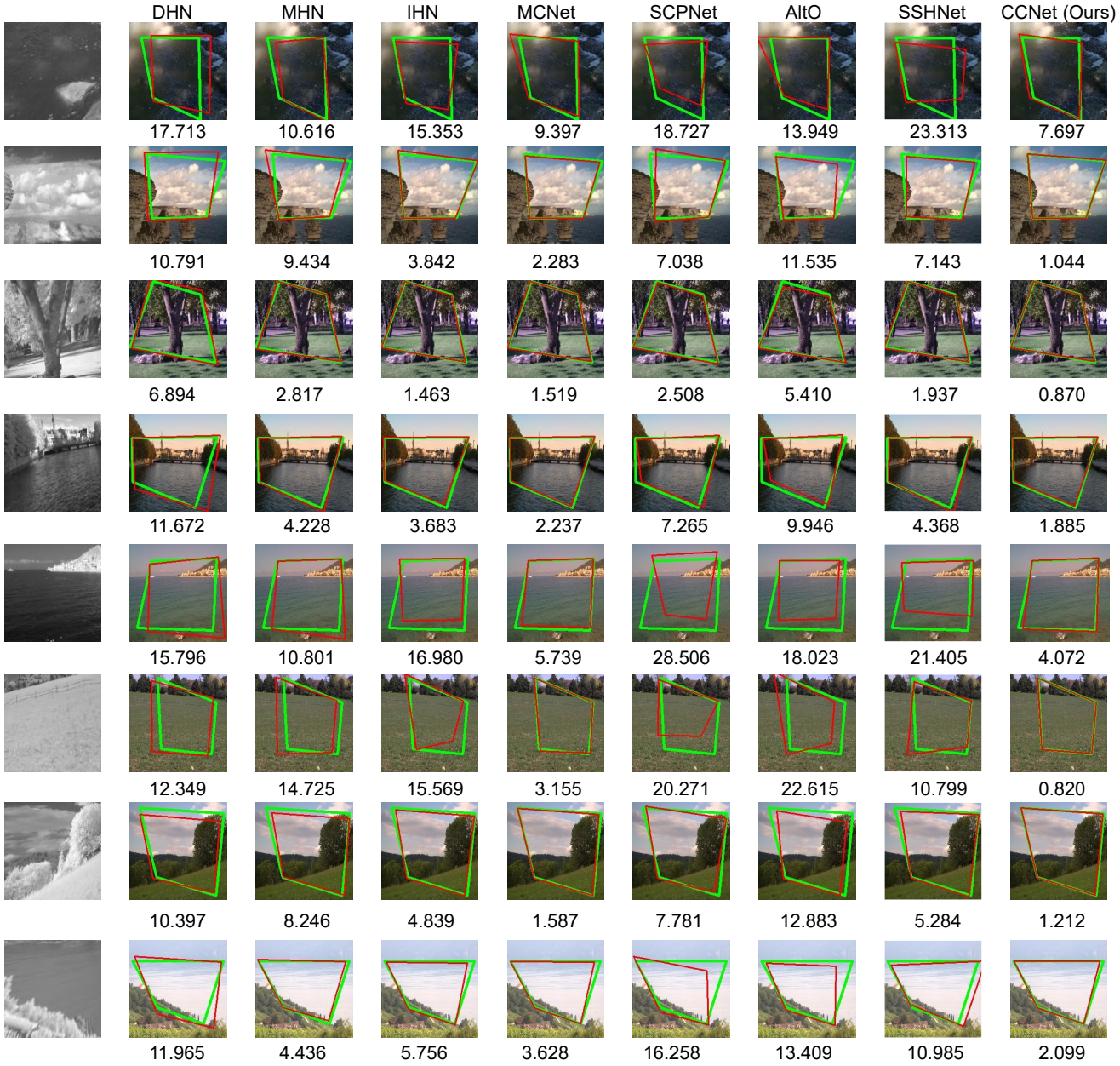

Figure 8. Visualization results of within-dataset evaluation on RGB-NIR. The first column displays the source image to be warped, while the subsequent columns present the corresponding target images. Below each target image, the mean average corner error (MACE) is indicated. Greater similarity between the red and green quadrilaterals signifies higher estimation accuracy.

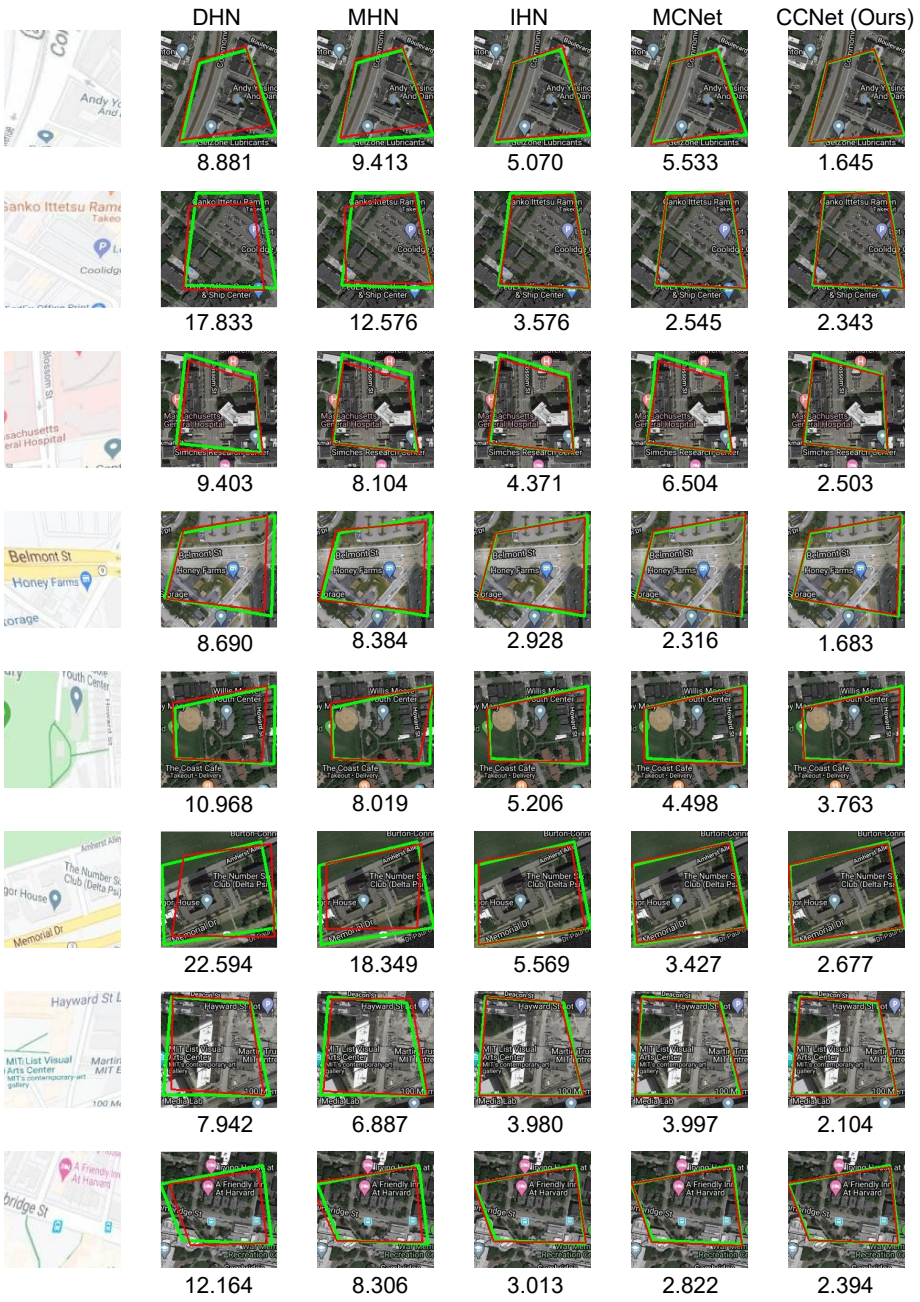

Figure 9. Visualization results of zero-shot evaluation on GoogleMap. The first column displays the source image to be warped, while the subsequent columns present the corresponding target images. Below each target image, the mean average corner error (MACE) is indicated. Greater similarity between the red and green quadrilaterals signifies higher estimation accuracy.

Figure 10. Visualization results of zero-shot evaluation on GoogleEarth. The first column displays the source image to be warped, while the subsequent columns present the corresponding target images. Below each target image, the mean average corner error (MACE) is indicated. Greater similarity between the red and green quadrilaterals signifies higher estimation accuracy.

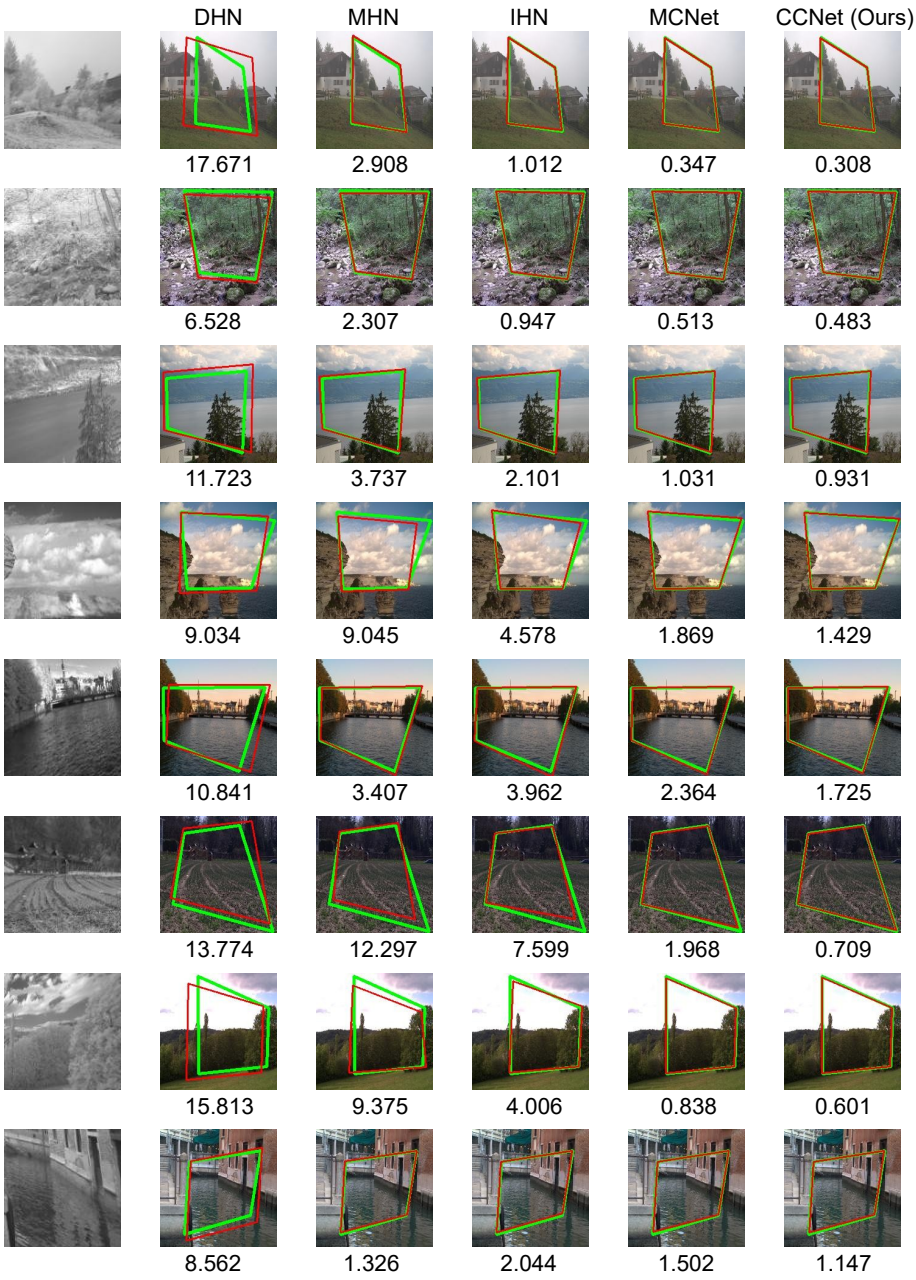

Figure 11. Visualization results of zero-shot evaluation on RGB-NIR. The first column displays the source image to be warped, while the subsequent columns present the corresponding target images. Below each target image, the mean average corner error (MACE) is indicated. Greater similarity between the red and green quadrilaterals signifies higher estimation accuracy.