Cross-Rejective Open-Set SAR Image Registration

Supplementary Material

1. Supplementary Experiments

1.1. Details of Datasets

In experiments, we use four datasets to validate the performance of the proposed method, including YellowR1, YellowR2, Wuhan, and Yama datasets, and each dataset is composed of two SAR images: one reference image and one sensed image.

The details of the four datasets are as follows:

- **Wuhan dataset**: Two SAR images are captured by the ALOS-PALSAR satellite over Wuhan, China, on 4 June 2006, and 7 March 2009, respectively, as shown in Figure 1. Each image has a size of 400×400 pixels with a resolution of 10 meters.
- **Yama dataset**: Two SAR images are acquired in the Yama region of Australia in 2018 and 2019 by the ALOS-PALSAR, as shown in Figure 2. Each image has a size of 650×530 pixels.
- **YellowR1/R2 dataset**: Two datasets are acquired from the Yellow River dataset which was captured by the Radarsat-2 over the Yellow River in China on 18 June 2008 and 19 June 2009, respectively. The YellowR1 dataset has a size of 700×700 pixels with a resolution of eight meters, as shown in Figure 3. The YellowR2 dataset has a size of 1000×1000 pixels with a resolution of eight meters, as shown in Figure 4.

1.2. Details of Data Construction

Detected Keypoints: In the proposed method, we first use SIFT [4] to pre-detect keypoints on two images for each dataset, as shown in Table 1 of the main paper. Here, we exhibit the visualization of all detected keypoints on two images of each dataset, shown in Figures 1, 2, 3, and 4, where the detected keypoints on the reference image and the sensed image are represented by red dots and blue dots, respectively.

Multi-scale Datasets: After detecting m and n keypoints respectively from two SAR images, each keypoint is used as the centre to capture its image patches and construct a registration dataset for image registration. To enhance the dataset, multiple scales are used to obtain more samples for each keypoint [5]. In experiments, we apply five scales to crop the image patches for each keypoint, and they are 64×64, 68×68, 72×72, 76×76, 80×80 pixels. Moreover, each keypoint's eight nearest neighbours, including up, down, left, right, top-left, bottom-left, top-right, bottom-right directions, are also used to capture the image patches with five scales, and then 45 image patches are obtained for each keypoint. Finally, based on $m + n$ keypoints, we will obtain $45\times(m+n)$ image patches and construct a multi-scale registration dataset for the proposed model. The number of all image patches for each dataset is shown in Table 1. Note that all images are resized to 64×64 to feed into the network.

Table 1. The number of keypoints pre-detected by SIFT

Dataset	YellowR1	YellowR2	Wuhan	Yama
R-Image	1339	1379	840	1223
S-Image	1201	1071	1213	1374
Image Patches	114300	110250	92385	116865

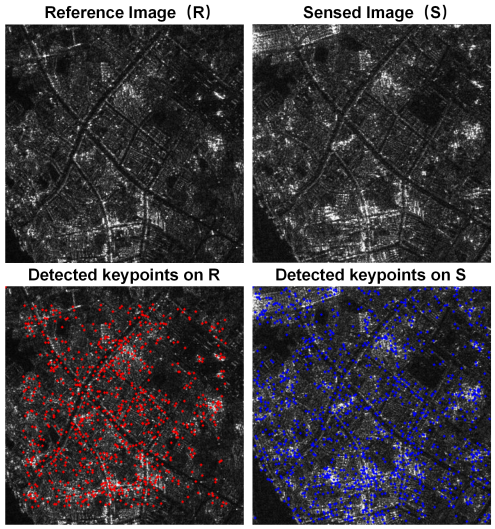

Figure 1. The Wuhan dataset with the size 400×400

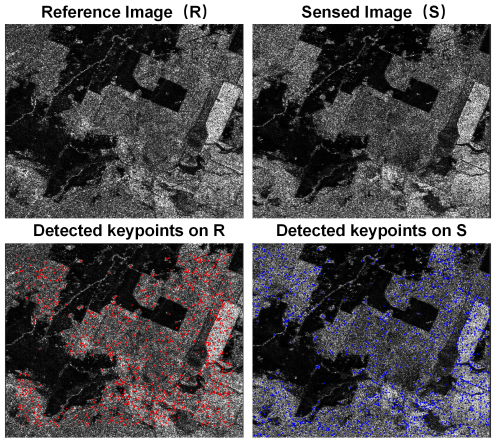

Figure 2. The Yama dataset with the size 650×530

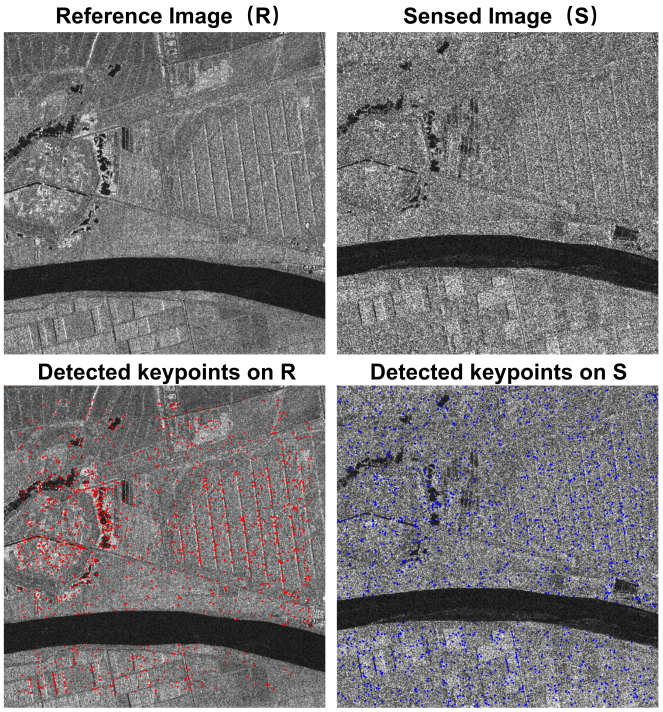

Figure 3. The YellowR1 dataset with the size 700×700

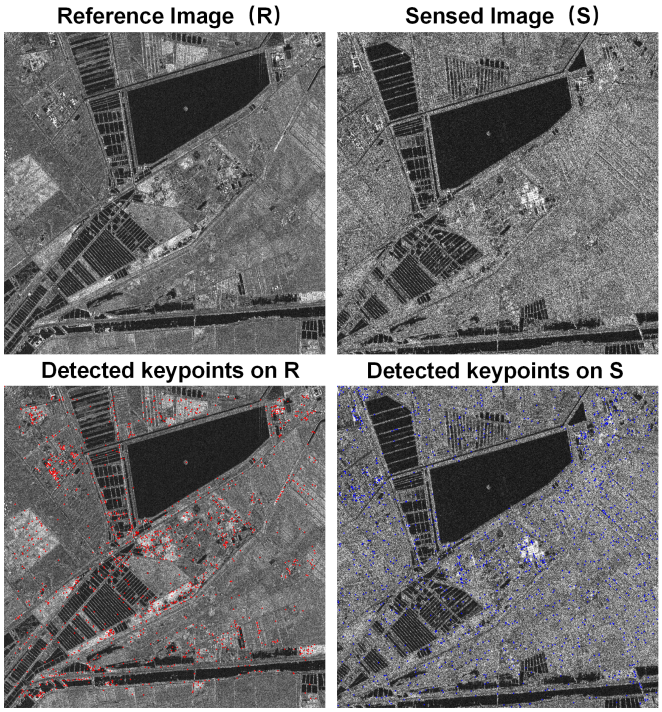

Figure 4. The YellowR2 dataset with the size 1000×1000

In addition, considering that there are inherent and irrelievable diversities between two SAR images, such as the changes in the Earth's surface, noise, etc, we employ the conventional preprocessing referred to [6] which utilizes an initial transformation T_0 to transform n keypoints on the sensed image into the reference image, and then obtain n transformed keypoints. Based on n transformer keypoints, we capture image patches corresponding to n keypoints from the reference images instead of the sensed image, which means the image patches of all $m+n$ keypoints are from the reference images, to alleviate the effect of inherent diversities among reference and sensed images. In experiments, we use a simple registration method (SIFT [4]) to obtain the initial transformation matrix T_0.

1.3. Details of Compared Methods

In experiments, we employ 11 comparative experiments, including SIFT [4], SAR-SIFT [1], DNN [11], SuperPoint [3], Sparse-NCNet [10], MSDF-Net [5], STDT-Net [2], AdaSSIR [6], DALF [8], XFeat [9], and DBMDF [7], where the first two are handcrafted feature-based methods and the last nine are DL-based methods. Especially, DALF and XFeat are two state-of-the-art methods of image registration on natural images. The details of all compared methods are shown as follows:

- **SIFT** [4]: The Scale-Invariant Feature Transform (SIFT) algorithm is a feature detection method that identifies and describes local features in images. SIFT is robust to variations in scale, rotation, and illumination, and it has been a baseline for many image registration algorithms.
- **SAR-SIFT** [1]: this method is an adaptation of the Scale-Invariant Feature Transform (SIFT) specifically designed for SAR images.
- **DNN** [11]: this method employs a Deep Neural Network (DNN) to seek out matched-point pairs, and it marks a milestone in the use of DNNs for SAR image registration.
- **Sparse-NCNet** [10]: this method uses Sparse-NCNet to obtain matched-point pairs and then applies the RANSAC algorithm to remove erroneous pairs for computing the transformation matrix.
- **SuperPoint** [3]: this method proposes a self-supervised keypoint detector and descriptor which are suitable for a large number of multiple-view geometry problems.
- **MSDF-Net** [5]: this method uses a deep forest to seek out the matched-point pairs from the view of a binary classification problem, where a multi-scale dataset is constructed for SAR image registration.
- **AdaSSIR** [6]: this method treats each keypoint as an independent instance and trains a contrastive network for registration to extract the latent features for keypoints and then seek out more correct matched-point pairs.
- **STDT-Net** [2]: this method regards the SAR image registration directly as a multi-classification problem, and then it constructs a dual-transform multi-classification

Table 2. Registration performance on four datasets under different rejection thresholds (λ)

Datasets	YellowR1			YellowR2			Yama			Wuhan		
λ	RMS_{all}	RMS_{LOO}	N_{red}	RMS_{all}	RMS_{LOO}	N_{red}	RMS_{all}	RMS_{LOO}	N_{red}	RMS_{all}	RMS_{LOO}	N_{red}
0.65	0.9400	0.9502	70	1.0041	1.0141	46	1.0211	1.0352	142	1.0226	1.0543	114
0.75	0.7993	0.8021	47	0.8791	0.8804	28	0.8791	0.8845	109	0.8256	0.8476	69
0.85	0.4282	0.4322	16	0.4160	0.4230	8	0.6597	0.6687	48	0.5739	0.5896	31
0.90	0.3085	0.3120	12	0.2846	0.2957	7	0.4274	0.4401	31	0.3295	0.3334	20
0.95	**0.2533**	**0.2565**	11	**0.0952**	**0.1386**	6	**0.3014**	**0.3022**	25	**0.1339**	**0.1402**	16
0.97	**0.2533**	**0.2565**	11	**0.0952**	**0.1386**	6	**0.3014**	**0.3022**	25	**0.1339**	**0.1402**	16
0.99	**0.2533**	**0.2565**	11	**0.0952**	**0.1386**	6	**0.3014**	**0.3022**	25	**0.1339**	**0.1402**	16

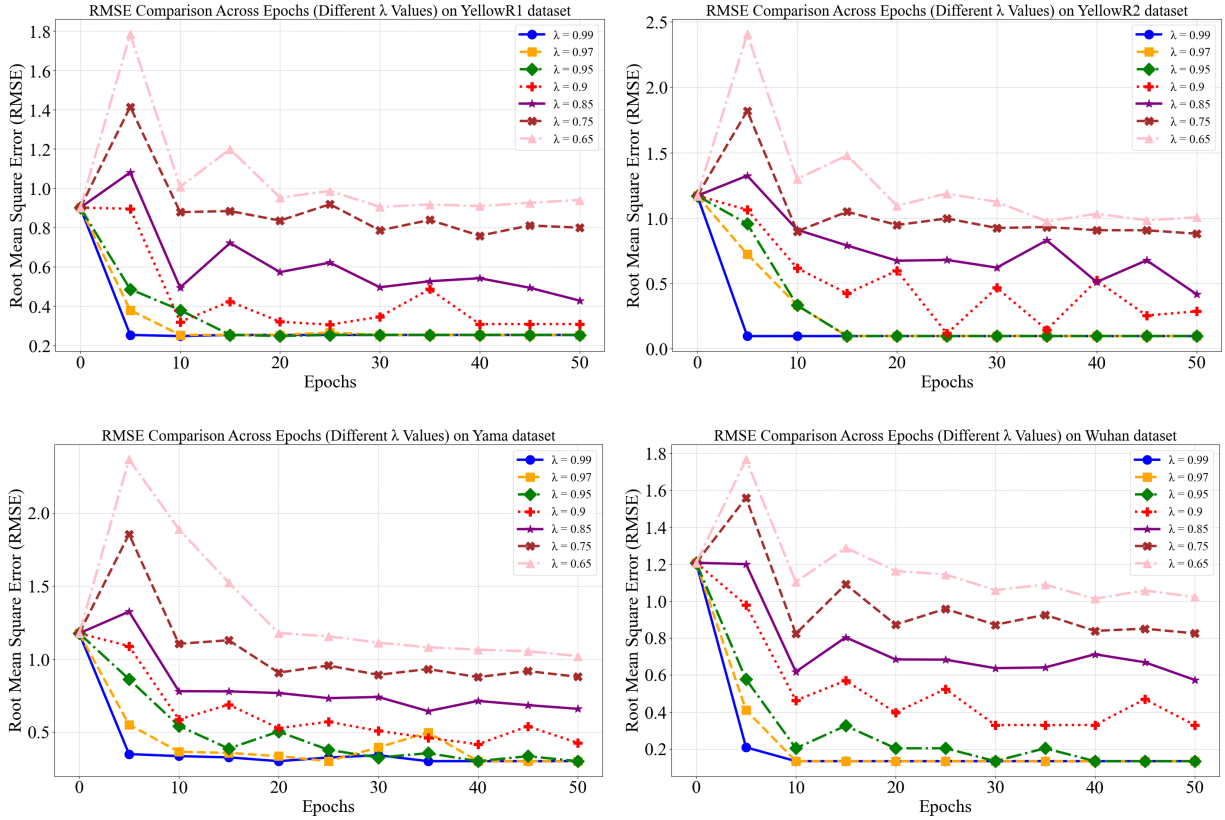

Figure 5. The registration performances (RMS_{all}) under different λ (0.65, 0.75, 0.85, 0.90, 0.95, 0.97, 0.99) for four datasets.

network to seek the matched-point pairs.

- **DALF** [8]: this method proposes a deformation-aware network to jointly detect and descript keypoints, which moderates the difficulty of matching deformable surfaces.
- **XFeat** [9]: this method designs a lightweight convolutional neural network architecture by adjusting the channel distribution of the convolutional network, which utilizes a novel match refinement module to achieve semi-dense matching efficiently the first time.
- **DBMDF** [7]: this method proposes a deep learning-based detector and descriptor for keypoint for SAR image registration, meanwhile constructing a coarse-to-fine registration method to obtain more robust matched points.

1.4. Evaluation Metrics

In experiments, we apply three evaluation metrics to evaluate the performance of SAR image registration, including N_{red}, RMS_{all} and RMS_{LOO}, and their details are given as follows:

- N_{red}: it represents the number of matched-point pairs sought by the registration method. Under the condition of a comparable root mean square error, a higher number of matched-point pairs indicates better registration ability.
- RMS_{all}: it represents the root mean square error of the

registration results, calculated by

$$RMS_{\text{all}} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(p_i^R - \hat{p}_i^S)^2}, \quad (1)$$

where $\hat{p}_i^S = T(p_i^S)$, and $(p_i^R, \hat{p}_i^S)$ is a pair of matched points sought by the registration method. T is the transformation of image registration obtained by the registration method, and k is the number of matched-point pairs, $k = N_{\text{red}}$. If the value of RMS_{all} is less than or equal to 1, it indicates that the registration performance attains sub-pixel accuracy.

- RMS_{LOO}: it denotes the error obtained based on the leave-one-out strategy and root mean square error. For each matched-point pair in N_{red}, RMS_{LOO} is the average of all errors of all $N_{\text{red}} - 1$ matched-point pair.

1.5. Detailed Results on Rejection Thresholds (λ)

In Section 4.4, we presented the variation of the RMS_{all} metric under different rejection thresholds ($\lambda = 0.65, 0.75, 0.85, 0.90, 0.95, 0.97, 0.99$). In this section, we provide more details of RMS_{all} under different rejection thresholds λ. The detailed results are shown in Table 2, and the enlarged plots are also redisplayed here and shown in Figure 5.

1.6. Analyses on Obtained Matched-Point Pairs

To provide a more qualitative analysis of the point pair variations during the registration process for each dataset, we supply more results by systematically evaluating the changes in the obtained matched-point pairs for four datasets. All changes are recorded from Epoch 5 to Epoch 50 with 5-epoch intervals (the 5th, 10th, 15th, 20th, 25th, 30th, 35th, 40th, 45th and 50th epochs), as detailed in Table 3. From Table 3, it is obvious that the numbers of the obtained matched-point pairs exhibit a downward trend from the beginning to convergence for four datasets, and the registration accuracies are gradually improved with the modification of these sought matched-point pairs. Furthermore, to visually illustrate the changes in the sought matched-point pairs during the iterative process, we draw the registration connecting-line charts in these iterations for four datasets, as shown in Figures 8, 9, 10, and 11. Note that the yellow dotted line expresses the deleted matched-point pairs and the purple line expresses the added matched-point pair.

From the four figures, it is seen that the matching lines are mostly parallel, which indicates the proposed method can effectively seek out more accurate matched-point pairs. Especially, compared with these epochs before convergence, the final epoch obtains more accurate pairs. For example, for the YellowR2 dataset, after two pairs {(865, 85), (773, 161)} (A) and {(510, 870), (417, 948)} (B) are removed in the 15th epoch, the obtained registration result is stabilized in the subsequent iterations, only remaining six more accurate matched-point pairs. Among six pairs, five pairs have an x-coordinate difference of 92 and one pair has a difference of 93, when all have a y-coordinate difference of 77. The removed pairs have y-coordinate differences of 76 and 78, and the pair (B) has an x-coordinate difference of 93. Compared with the remained pairs, there are more diversities among the removed pairs.

For the Yama dataset, it is obvious that 37 matched-point pairs obtained in the 5th epoch are deleted in the 10th epoch, as shown in the yellow dotted lines, which also brings the 0.11 improvement (RMS_{all}). Moreover, it is observed that although the matched-point pair {(545, 275), (557, 277)} (C) is re-obtained in the 45th epoch but still removed in the 50th epoch. In contrast, the matched-point pair {(341, 386), (351, 388)} is sought in the 35th epoch and remained until the end of the 50th epoch. According to the analyses on the coordinate values of all matched-point pairs, it is known that the pair D is more consistent with the final matched-point pairs than the pair C, which means D is higher confidence than C. It indicates that the proposed method can obtain more accurate and consistent matched-point pairs to obtain more precise registration results.

For the Wuhan dataset, it is observed that although the matched-point pair {(176, 223), (195, 203)} (E) is modified to {(177, 224), (195, 203)} in the 10th epoch, it is reverted to {(176, 223), (195, 203)} in the 25th epoch but still removed entirely after the 30th epoch. It illustrates that this matched-point pair is unstable and with low confidence. And our method can effectively remove some matched-point pairs with low confidence by improving in feature mapping and rejection. For the final sought matched-point pairs, the deviation in the vertical and horizontal axes between matched points is consistent, specifically 18 pixels in the x-coordinate and 20 pixels in the y-coordinate. In short, the results of the change of matched-point pairs sought by the proposed method in iterations illustrate that the proposed method obtains more accurate matched-point pairs by enhancing the feature of keypoint and cross-rejective open-set recognition for precise registration.

1.7. Registration on the Yellow River dataset

To further demonstrate the effectiveness of the proposed method, we also conduct registration on the Yellow River dataset. For the Yellow River dataset, the reference image and the sensed image are collected by the Radarsat-2 on 18 June 2008 and 19 June 2009, respectively. The resolution of two SAR images is 7666×7692 pixels, as shown in Figure 6. Compared with the previous four datasets, this dataset exhibits pronounced rotational transformations, as well as extensive regions devoid of detectable keypoints, and these characteristics bring more challenges for the registration. Note that YellowR1 and YellowR2 are captured

Table 3. The quantity changes of matched-point pairs obtained by CroR-OSIR in iterations

Epochs		5	10	15	20	25	30	35	40	45	50
YellowR1	N_{red}	15	12	11	11	10	11	11	11	11	11
	RMS_{all}	0.4859	0.3797	0.2533	0.2533	0.2533	0.2533	0.2533	0.2533	0.2533	0.2533
YellowR2	N_{red}	23	8	6	6	6	6	6	6	6	6
	RMS_{all}	0.9543	0.3309	0.0952	0.0952	0.0952	0.0952	0.0952	0.0952	0.0952	0.0952
Yama	N_{red}	69	32	27	31	26	25	27	25	26	25
	RMS_{all}	0.8629	0.5429	0.3875	0.5039	0.3790	0.3260	0.3562	0.3014	0.3354	0.3014
Wuhan	N_{red}	31	17	19	17	17	16	17	16	16	16
	RMS_{all}	0.5768	0.2036	0.3254	0.2036	0.2036	0.1339	0.2036	0.1339	0.1339	0.1339

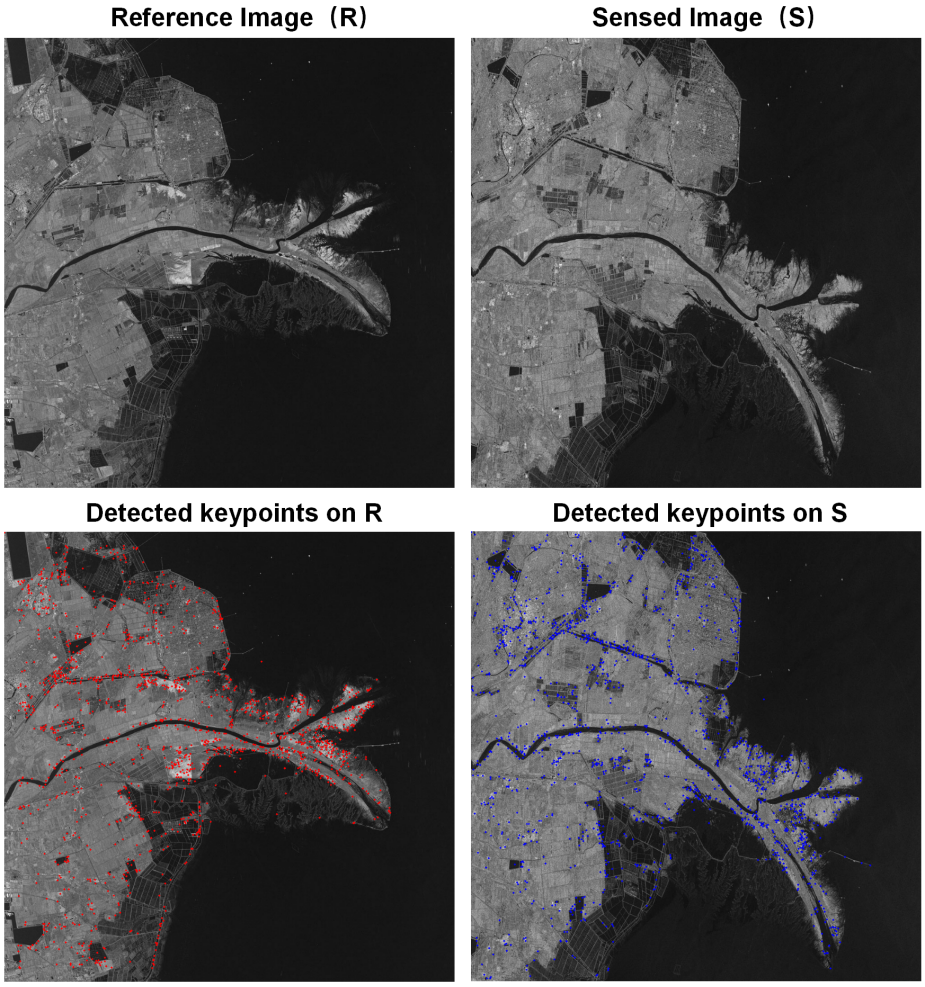

Figure 6. The Yellow River dataset with the size 7666×7692

from the Yellow River dataset, but the rotational transformation between their images is eliminated before cropping. In this experiment, we still pre-detect keypoints from two images by SIFT, and the detected keypoints are shown in Figure 6. The same settings in Section 4.1 of our main paper are implemented, and the rejection threshold is set as $\lambda = 0.95$. The experimental results are shown in Table 4. Here, MSDF-Net[5] is not used in this experiment, since its code is not published.

From Table 4, it is seen that the proposed method obtains higher registration performance than these compared methods, meanwhile seeking out more matched-point pairs (N_{red}), without any post-processing strategies that assist in filtering out some incorrectly matched-point pairs. In com-

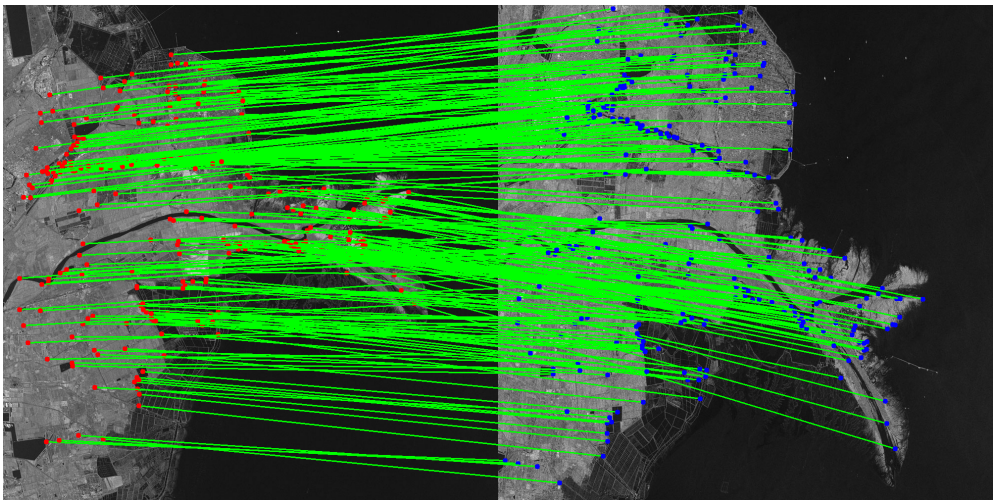

Figure 7. The visualization of the registration results on the Yellow River dataset

Table 4. The Registration Performance on Yellow River Dataset

Methods	$RMS_{\textbf{all}}$	$RMS_{\textbf{LOO}}$	$N_{\textbf{red}}$
SIFT [4]	1.3936	1.3996	21
SAR-SIFT [1]	2.9355	2.9422	11
DNN [11]	1.0525	1.2616	19
SuperPoint [3]	1.9874	2.0866	20
Sparse-NCNet [10]	1.0505	1.0618	93
AdaSSIR [6]	0.9945	0.9998	62
STDT-NET [2]	1.1431	1.1523	43
DALF [8]	<u>0.8357</u>	<u>0.8489</u>	63
XFeat [9]	0.9244	0.9297	176
DBMDF [7]	1.3887	1.4655	19
CroR-OSIR	**0.8275**	**0.8276**	202

pared methods, only three methods (AdaSSIR, DALF and XFeat) obtain a performance of less than 1, and the rest methods are not effective on Yellow River as in the previous four datasets. The experimental result demonstrates that our method (CroR-OSIR) is more effective for handling the affine transformation than these existing methods.

Moreover, the visualization of the registration obtained by our method is shown in Figure 7. From Figure 7, it is seen that, compared with other datasets, the matching lines on this dataset exhibit slight intersections, which may be caused by the rotation. The proposed method obtains abundant matched-point pairs. It indicates the proposed method can adapt well to the image's rotation angle effects. The matching points are accurately positioned and evenly distributed, demonstrating the robustness of our method against rotation transformations.

References

[1] Flora Dellinger, Julie Delon, Yann Gousseau, Julien Michel, and Florence Tupin. Sar-sift: a sift-like algorithm for sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):453–466, 2014. 2, 6

[2] Xiaozheng Deng, Shasha Mao, Jinyuan Yang, Shiming Lu, Shuiping Gou, Youming Zhou, and Licheng Jiao. Multi-class double-transformation network for sar image registration. *Remote Sensing*, 15(11):2927, 2023. 2, 6

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 6

[4] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 1, 2, 6

[5] Shasha Mao, Jinyuan Yang, Shuiping Gou, Licheng Jiao, Tao Xiong, and Lin Xiong. Multi-scale fused sar image registration based on deep forest. *Remote Sensing*, 13(11):2227, 2021. 1, 2, 5

[6] Shasha Mao, Jinyuan Yang, Shuiping Gou, Kai Lu, Licheng Jiao, Tao Xiong, and Lin Xiong. Adaptive self-supervised sar image registration with modifications of alignment transformation. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023. 2, 6

[7] Javid Norouzi, Mohammad Sadegh Helfroush, Alireza Liaghat, and Habibollah Danyali. A deep-based approach for multi-descriptor feature extraction: Applications on sar image registration. *Expert Systems with Applications*, 254: 124291, 2024. 2, 3, 6

[8] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Enhancing deformable local features by jointly learning to detect and describe keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1306–1315, 2023. 2, 3, 6

[9] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2682–2691, 2024. 2, 3, 6

[10] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 605–621. Springer, 2020. 2, 6

[11] Shuang Wang, Dou Quan, Xuefeng Liang, Mengdan Ning, Yanhe Guo, and Licheng Jiao. A deep learning framework for remote sensing image registration. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:148–164, 2018. 2, 6

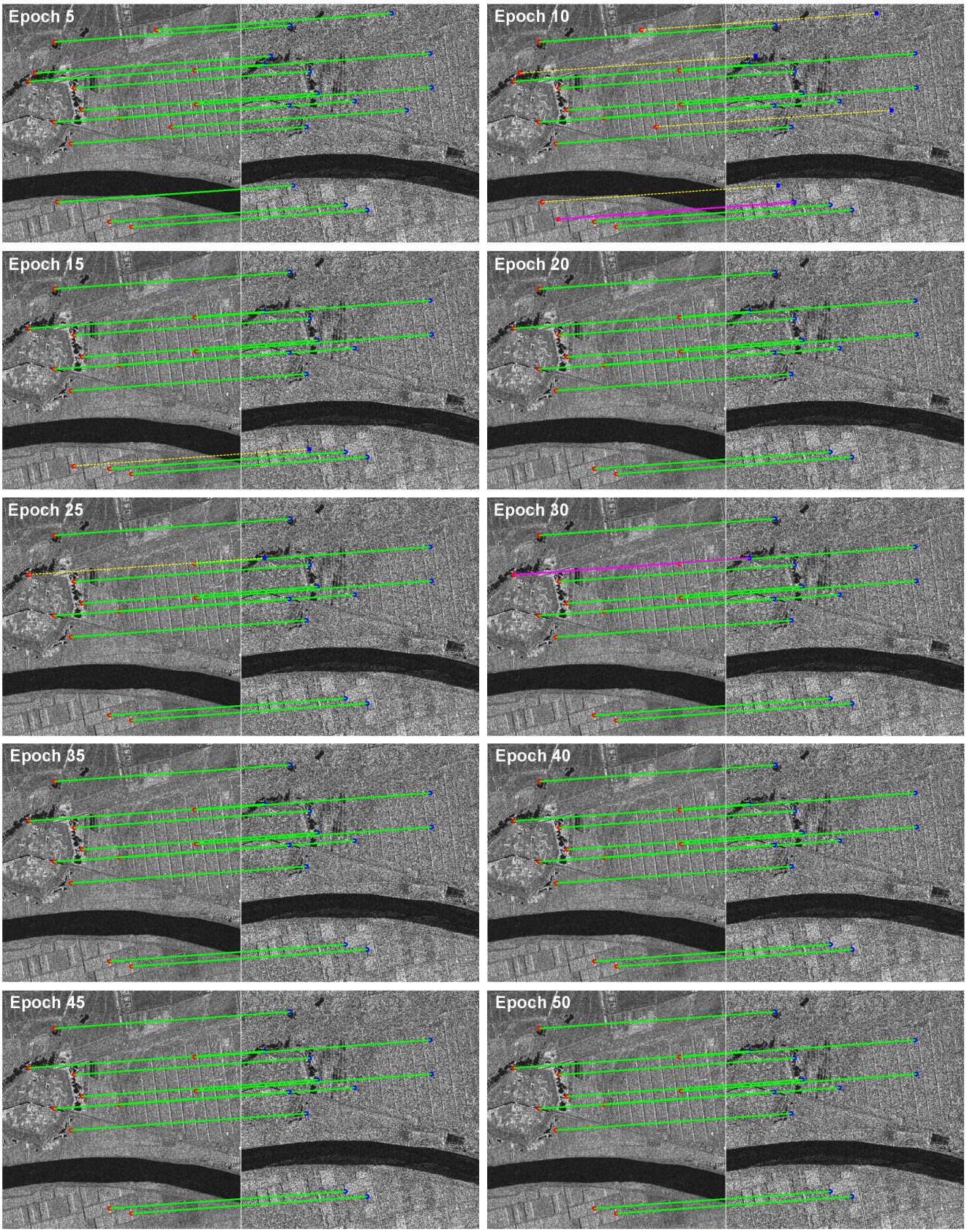

Figure 8. The registration connecting-line charts on YellowR1 dataset during the iterative process, where the yellow dotted line expresses the deleted matched-points pair and the purple connecting-line expresses the added matched-points pair.

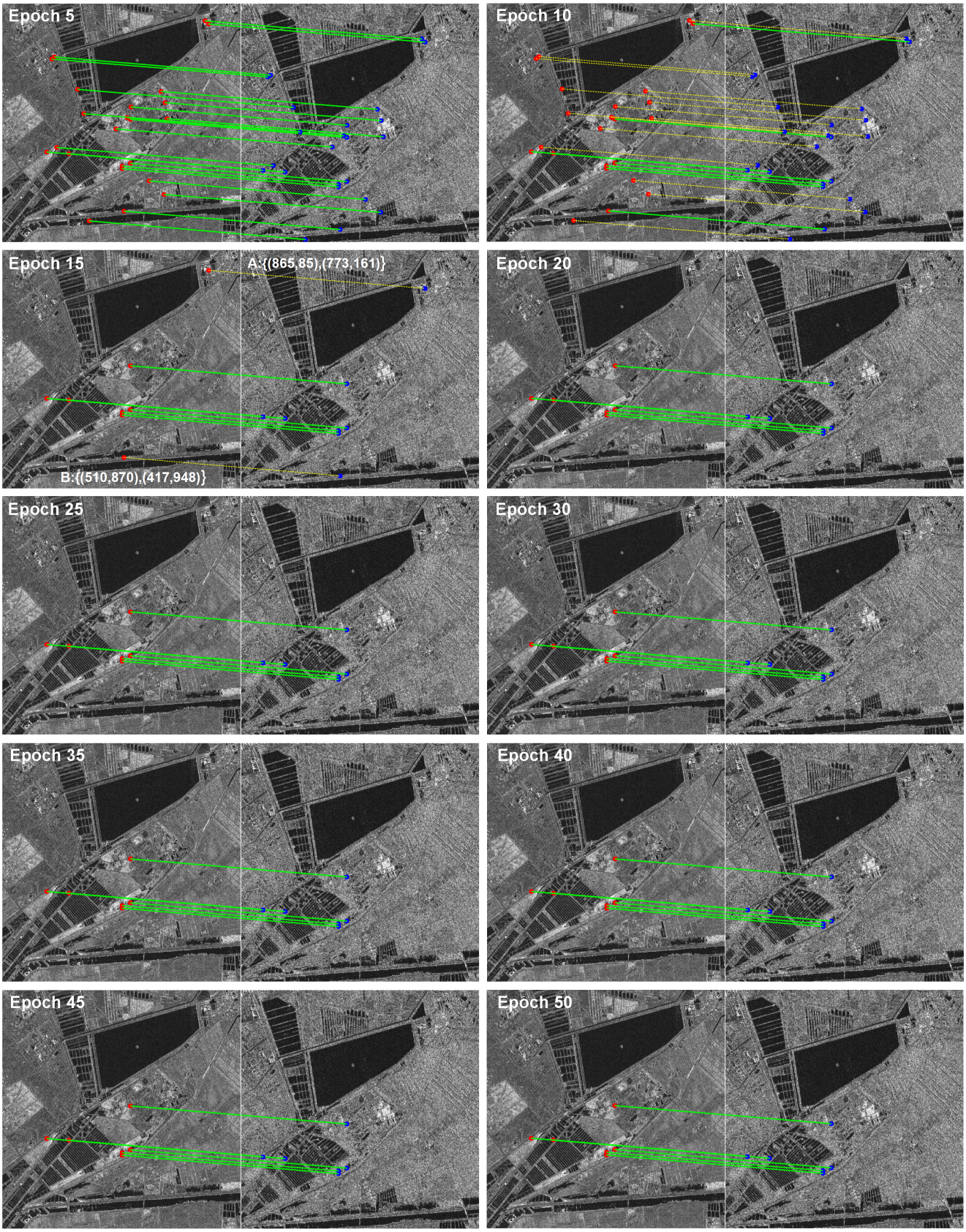

Figure 9. The registration connecting-line charts on YellowR2 dataset during the iterative process, where the yellow dotted line expresses the deleted matched-points pair and the purple connecting-line expresses the added matched-points pair.

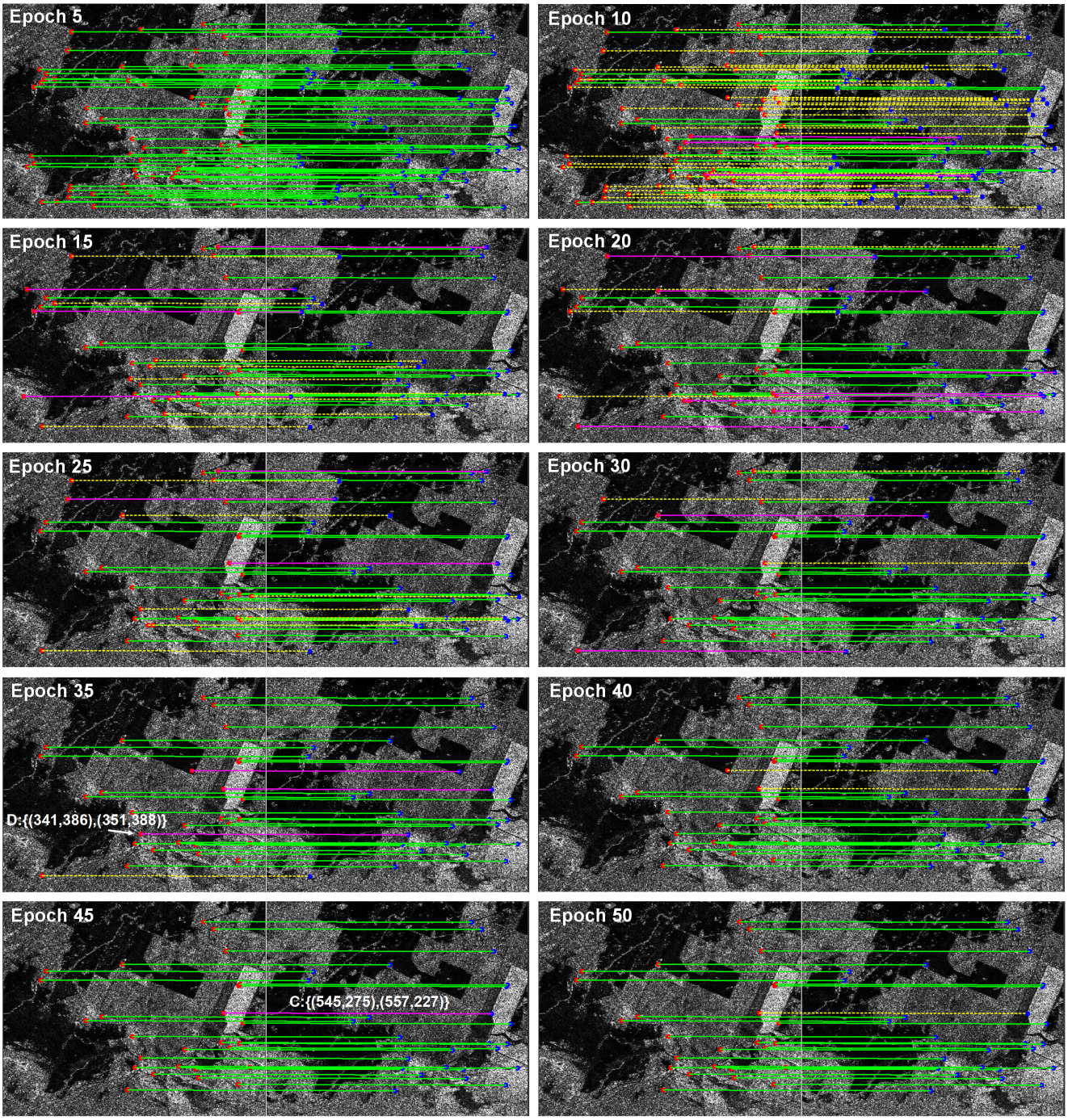

Figure 10. The registration connecting-line charts on Yama dataset during the iterative process, where the yellow dotted line expresses the deleted matched-points pair and the purple connecting-line expresses the added matched-points pair.

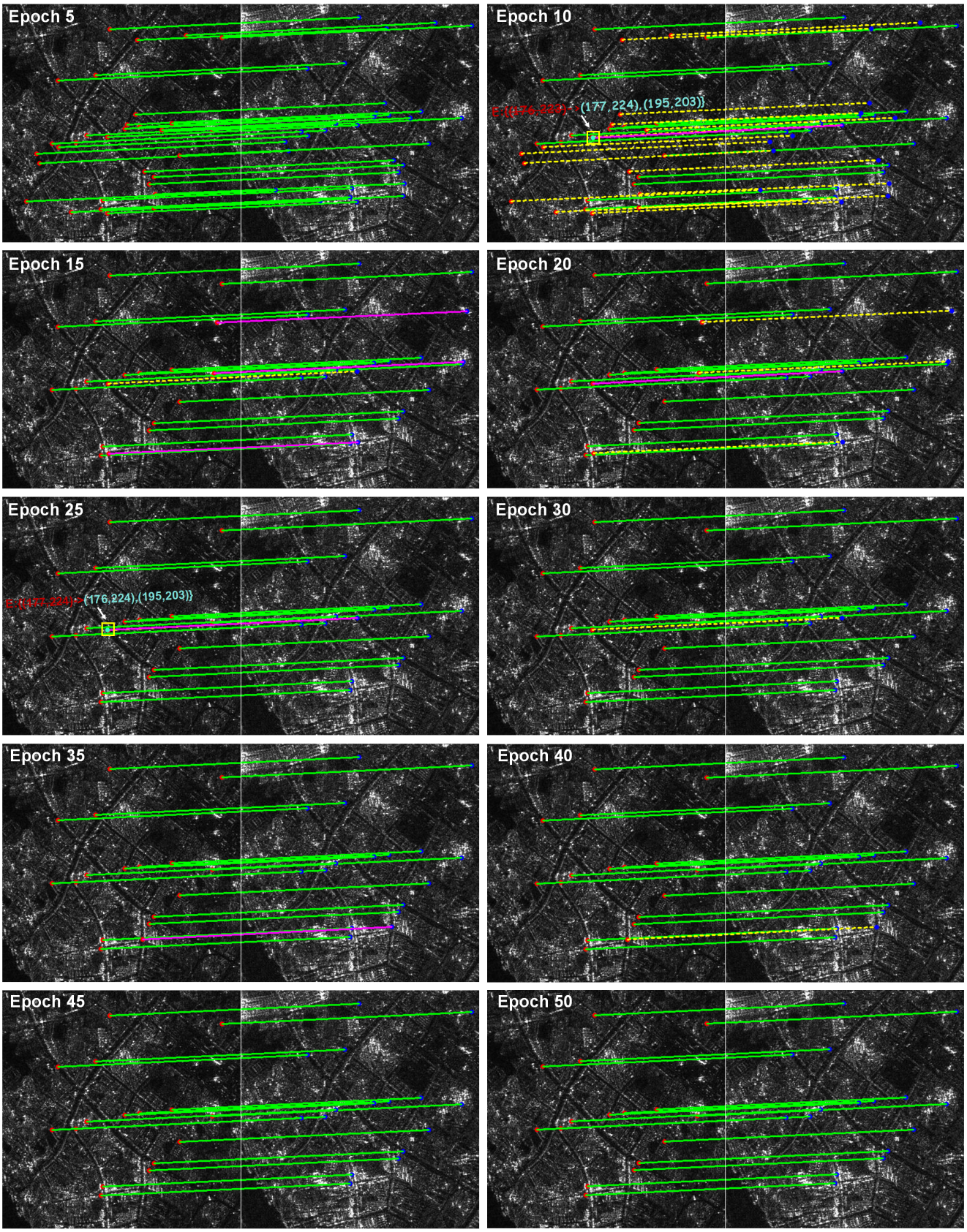

Figure 11. The registration connecting-line charts on Wuhan dataset during the iterative process, where the yellow dotted line expresses the deleted matched-points pair and the purple connecting-line expresses the added matched-points pair.