remote sensing

MDPI

Article

A Coarse-to-Fine Feature Match Network Using Transformers for Remote Sensing Image Registration

Chenbin Liang [1,2,3], Yunyun Dong [1,*], Changjun Zhao [4] and Zengguo Sun [5]

[1] Northwest Land and Resource Research Center, Shaanxi Normal University, Xi'an 710119, China; liangchenbin2019@ia.ac.cn
[2] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China
[4] School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China
[5] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
* Correspondence: dongyunyun@snnu.edu.cn; Tel.: +86-020-85310659

Abstract: Feature matching is a core step in multi-source remote sensing image registration approaches based on feature. However, for existing methods, whether traditional classical SIFT algorithm or deep learning-based methods, they essentially rely on generating descriptors from local regions of feature points, which can lead to low matching success rates due to various challenges, including gray-scale changes, content changes, local similarity, and occlusions between images. Inspired by the human approach of finding rough corresponding regions globally and then carefully comparing local regions, and the excellent global attention property of transformers, the proposed feature matching network adopts a coarse-to-fine matching strategy that utilizes both global and local information between images to predict corresponding feature points. Importantly, the network has great flexibility of matching corresponding points for any feature points and can be effectively trained without strong supervised signals of corresponding feature points and only require the true geometric transformation between images. The qualitative experiment illustrate the effectiveness of the proposed network by matching feature points extracted by SIFT or sampled uniformly. In the quantitative experiments, we used feature points extracted by SIFT, SuperPoint, and LoFTR as the keypoints to be matched. We then calculated the mean match success ratio (MSR) and mean reprojection error (MRE) of each method at different thresholds in the test dataset. Additionally, boxplot graphs were plotted to visualize the distributions. By comparing the MSR and MRE values as well as their distributions with other methods, we can conclude that the proposed method consistently outperforms the comparison methods in terms of MSR at different thresholds. Moreover, the MSR of the proposed method remains within a reasonable range compared to the MRE of other methods.

Keywords: feature matching; image registration; coarse-to-fine matching; transformer; deep learning

check for updates

Citation: Liang, C.; Dong, Y.; Zhao, C.; Sun, Z. A Coarse-to-Fine Feature Match Network Using Transformers for Remote Sensing Image Registration. *Remote Sens.* **2023**, *15*, 3243. https://doi.org/10.3390/rs15133243

Academic Editor: Chiman Kwan

Received: 10 April 2023
Revised: 3 June 2023
Accepted: 10 June 2023
Published: 23 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the ever-advancing technologies of sensors and continuously increasing information infrastructure, remote sensing images are becoming more and more diverse in terms of spatial, spectral and temporal resolution [1]. Different images of the same observed object or scene can provide different or even complementary information. To obtain a more detailed representation of observed objects, the fusion of different images has drawn extensive attention in recent years. However, accurate image registration is essential for effective image fusion. Image registration is a crucial preprocessing step for further application of remote sensing images, and its accuracy is highly desirable.

Despite the development of numerous registration approaches in the last few decades, it remains a challenge to register images due to complex geometric deformations, radio-

metric discrepancies, and even content changes [2]. Existing image registration methods can be broadly categorized as intensity-based, feature-based and learning-based methods. The main objective of intensity-based methods is to align images by identifying the most similar image block using similarity metrics such as normalized cross-correlation [3] and mutual information [4]. While these methods are easy to implement and robust to linear differences in intensity, they are sensitive to non-linear intensity differences and fail to cope with non-shift geometric deformations. Besides, the image registration based on phase correlation is a promising method. It has drawn extensive attention due to its sub-pixel accuracy and robustness to image gray changes. For example, OS-PC [5] can successfully register Optical and SAR images by combing robust feature representation and 3-D phase correlation and achieve sub-pixel accuracy. For the feature-based method, it is consisted of four steps: feature detection, feature description, feature matching and transformation estimation. The most popular methods are scale-invariant feature transformation (SIFT) [6] and its variants. Their prominent advantage is that it can handle with complex geometric deformations. However, their performance is affected by each individual process, while for the registration of optical and SAR images, due to their large gray differences and appearance differences, phase congruency model based on the local phase information of images and its variants are utilized to extract and describe radiation-robust feature points. For example, HOPC [7] can capture the geometric structure or shape features of images by building a dense descriptor of Histogram of Orientated phase congruency; RIFT [8] can reduce the effect of nonlinear radiation distortions by using phase congruency for feature point detection and using maximum index map for description; the combination of SAR-PC [9], used to extract feature responses for SAR images and an improved phase congruency model [10], can capture local spatial relationship between feature points; and R_2FD_2 [11] has robustness to radiation and rotation differences by combining feature detector of the multichannel autocorrelation of the log-Gabor and feature descriptor of maximum index map of the log-Gabor.

In recent years, with the flourish of deep learning (DL) and its excellent performance in the field of computer vision, such as image classification and object recognition, DL has been introduced to process the task of image registration. The main idea is to replace the handcrafted detectors and descriptors in prior experience or direct perception with powerful representation of convolution neural network (CNN) learned from training data automatically. So, feature detection, feature description based on DL methods, is proposed successively.

Feature detection based on deep learning (DL) aims to mimic the pipeline of handcrafted detectors by utilizing convolutional neural networks (CNNs). CNNs are employed to calculate feature maps with different scales, similar to corner response maps in the Harris corner method [12] and the response map of the Difference of Gaussian (DoG) in SIFT [6]. CNNs possess the inductive bias of translation equivariance [13] and locality, enabling them to recognize similar patterns appearing in different locations in an image. This makes CNNs well-suited for local feature extraction. ResNet [14] and Feature Pyramid Network (FPN) [15] are popular CNN architectures used for extracting local features. Once feature maps are obtained, a feature response map is computed by combining different feature maps, and local maxima points are identified as feature points. LIFT [16] and LFNet [17] are examples of neural networks that implement the feature point extraction pipeline, outputting feature points with orientation. R2D2 [18] calculates repeatability and reliability scores from feature maps to select keypoints with high repeatability and reliability. However, downsampling operations in generating feature scale maps can reduce the accuracy of feature point location, which may impact image registration performance. Additionally, manual annotation of ground-truth correspondences is required for some methods, limiting their ability to learn where handcrafted detectors fail.

For feature description based on deep learning (DL), various methods have been proposed, leveraging the strong representation capacity of deep convolutional neural networks (CNNs) . Learning feature descriptors involves deep metric learning, which aims to pull matching points closer and push non-matching points apart in the distance metric

space [19]. DL-based feature description methods can be classified into two categories: patch-based and dense descriptor methods.

Patch-based methods generate feature descriptors for sparsely distributed feature points detected by a feature detector. These methods use either a single-branch CNN or a Siamese CNN (consisting of two branches with shared weights). For example, L2-Net [20] employs a single-branch CNN where an image patch centered at a keypoint is fed as input to generate a feature descriptor. The generated descriptors are then compared in the metric space, often utilizing efficient methods like KD-trees for nearest neighbor computation. In Siamese CNNs, the inputs are pairs of image patches, and the outputs indicate feature matching or non-matching. A metric network is typically placed after the Siamese CNN, which can be a computation layer for distance calculation or learnable fully-connected layers with non-linearity functions. Different loss functions, such as pairwise contrastive loss [21], triplet loss [22], and listwise ranking loss based on average precision (AP) [23], are used to guide effective network learning. However, these methods require ground-truth correspondences during training, which limits their scalability and generalization.

Compared to patch-based methods, dense descriptor methods aim to generate descriptors for each pixel or uniformly sampled points in feature maps. Manual heuristics, such as local gradient histograms, are insufficient for acquiring robust and discriminative descriptors, especially in small patches with limited information [24]. Recently, dense descriptor methods based on deep learning have gained attention. These methods utilize a feature generation network backbone to process the input image and obtain a series of feature maps. The challenge lies in finding corresponding matches. The most common approach is to calculate correlations between feature descriptors, which can be local or global. Local correlations find short-range correspondences at a low cost, while global correlations find long-range correspondences but with higher computation cost. DGC-Net [25] utilizes global correlation to construct a global cost volume from coarse feature maps. GLU-Net [26] combines global and local correlation layers for accuracy and robustness. NC-Net [27] refines matches by examining consensus patterns in the 4D space of potential correspondences. GOCor [28] improves feature correlation layers to effectively learn spatial matching and disambiguate repeated patterns.

In summary, the existing feature detection based on DL methods suffer from the poor feature localization due to the down sampling in the CNN, which negatively impacts the performance of image registration. Furthermore, the existing dense correspondences methods based on DL focus mainly on the calculation of correlations and some other consensus constraints. However, the feature descriptors $\boldsymbol{f}$ extracted by CNNs have limited receptive field struggle to disambiguate correspondences in indistinctive areas. In contrast, humans use a broader global context in addition to local neighborhoods to find correspondences in these areas. Motivated by this, we propose a local feature match network, named as MatcherTF, where Transformer network [29] with self and cross attention layers are built to process the local feature descriptors to get transformed feature representation. Due to the property of global receptive field of Transformer, the further transformed feature representations are context-independent, improving the ability to disambiguate challenging matches. By integrating the feature descriptor network based on DL into the image matching pipeline, we can generate a much more discriminative descriptor for each feature points detected by handcrafted detector or selected randomly, improving the performance of feature matching. Our main contribution is summarized as follows:

- Transformers with self and cross attention layers are utilized to transform feature vectors to get context-dependent feature descriptors, which greatly improves their determination.
- A schema of coarse-to-fine feature matching is designed, which reduces the computational cost of global correlation to achieve the balance between efficiency and accuracy.
- A novel local feature match network, MatcherTF is proposed, which has great flexibility as its input can be any number of coordinate points on an image. It can be

used alone for remote sensing image registration or combined with existing feature detectors for high-quality image registration tasks.

2. Related Work

2.1. Image Registration of Natural Images

In the community of computer vision, image matching is a fundamental problem in many 3D computer vision tasks, such as visual localization, structure from motion (SfM), and simultaneous localization and mapping (SLAM). In terms of image matching methods, they are the most advanced and cutting-edge. Many classical and popular matching methods are proposed firstly to solve the natural image matching, such as the most classical and successful method of SIFT [6] and other popular deep-learning-based methods, SuperPoint [30], R2D2 [11], DGC-Net [25] and so on. Although these methods can be directly used for matching remote sensing images, the matching performance may suffer some degree of degeneration. This is mainly because the resolution difference between remote sensing images and natural images are large, namely, natural images mainly contain the fine geometric and textural features of many scenes, while the features captured in remote sensing images are more representative of the scene as a whole. Besides, for the data-driven method, the availability and size of training data are also significant factors that can not be overlooked. For natural images, they are captured by common cameras, and the intrinsic and extrinsic parameters of a camera can be adjusted independently. So, it is easy to acquire many pairs of images with the ground-truth geometric transformation. For example, HPatches dataset [31] is created by adjusting the parameters of cameras to acquire the ground-truth homography matrix, while for remote sensing images, they are captured by satellites in the space and it is difficult for researchers to adjust the attitude of satellites to acquire image pairs with ground-truth geometrical transformation at the present stage. So, there is hardly few remote sensing image registration datasets with ground-truth geometrical transformation label, which hinders the development of remote sensing image registration based on deep learning. To alleviate the problem of remote sensing image registration dataset, we propose an almost automatic way of creating remote sensing image registration dataset with ground-truth homograph transformation label by leveraging the geometric consistency of level-1C data product of sentinel-2 and level-2 data product of Landsat-8 images.

2.2. Image Registration of Remote Sensing Images

Recently, some image registration methods based on deep learning have also been introduced in the community of remote sensing. In terms of registration methods themselves, they are similar to the ones used by natural images. However, about the creation of training dataset, there are significant differences. In essence, the existing remote sensing image registration methods based on deep learning belongs to strongly supervised learning. They need the corresponding feature point pairs as supervised signal [32–34]. However, because a large number of feature point pairs are required to used as training data and manual selection is time-consuming, the commonly used method is extracting feature point pairs by using existing mature methods, such as SIFT [6]. As a result, the deep learning model trained by these data fails to learn anything in the area of images where the existing method fails. Thus, the trained model rarely yields performance improvement compared to existing algorithms. To alleviate the problem, we proposed a way of weakly supervised learning, which only requires the ground-truth homograph transformation between image pairs and some feature points selected randomly, combined with some feature points extracted by SIFT, to feed into our designed network to extend the ability of SIFT.

2.3. Transformer in the Computer Vision Tasks

Transformer [29] has become the de facto standard for sequence modeling in the field of natural language processing due to its prominent property of capturing long-term dependencies between words in a sequence. In the field of computer vision, Transformers

are drawing more attention, such as semantic segmentation [35] and object detection [36]. They and their cross-attention mechanism have been shown to bring performance improvements to these tasks. For feature descriptor [37], Transformer also can be utilized to fuse local and global context information to make them more discriminative. However, for vanilla Transformer, the memory cost is quadratic to the length of sequences due to the matrix multiplication, which has become a bottleneck of dealing with long sequences using Transformer. Especially for the task of feature description, it limits the dimension of feature vector and the number of feature vectors. Recently, some variants of Transformer [38–40] are proposed to improve its efficiency. Among them, Linear Transformer [40] reduces to the linear computational complexity by utilizing the kernel feature maps and the associativity characteristic of matrix multiplication. In this work, we adopt the linear Transformer to maintain a manageable computational cost.

3. Methods

3.1. The Overall Process of the Proposed Method

The flowchart of the proposed method is shown as Figure 1. It includes fours feature maps as input and two matching processes, namely coarse feature matching and fine feature matching. Among them, the coarse feature matching and fine feature matching are our main work, we will detail them as follows. Here, we briefly introduce how these feature maps are generated.

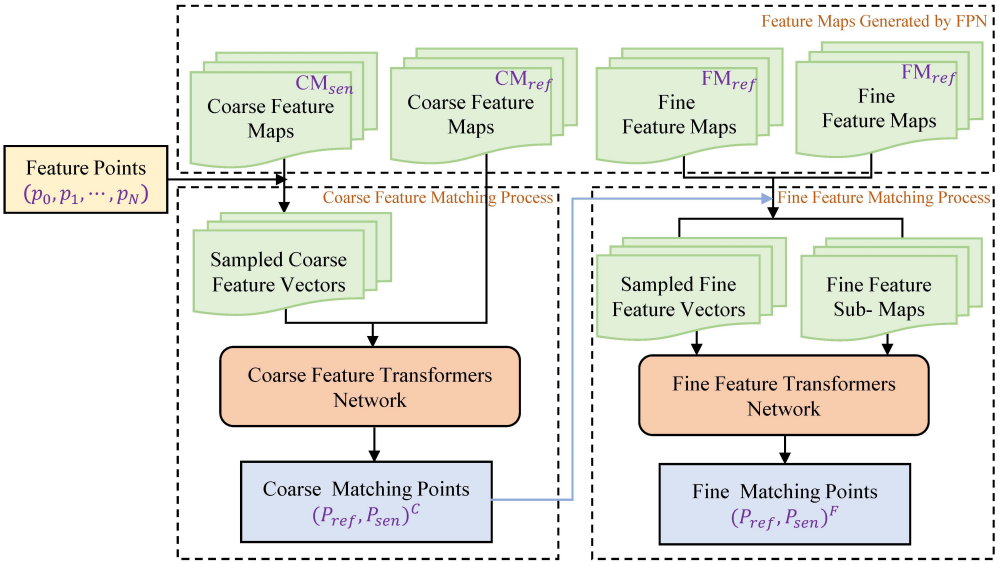


Figure 1. The Flowchart of The Proposed Method

Because CNN has the locality and the inductive bias of translation equivariance, namely, shifting the image and then feeding it through a number of convolutional layers is the same as feeding the original image through the same convolutional layers and then shifting the resulting feature maps [13], it is able to recognize similar patterns appearing in different locations in an image. Here, a standard convolutional neural network with feature pyramid network [15] is adopted to generate feature maps with different resolutions. In order to better explain the generation of the feature maps, we take as an example a pair of remote sensing images with reference image I_{ref} and sensed image I_{sen}. For the reference image I_{ref}, it will be fed into the feature pyramid network and get coarse feature maps CM_{ref} and fine feature maps FM_{ref}. Meanwhile, the sensed image I_{sen} is also be fed into the same feature pyramid network and get its coarse feature maps CM_{sen} and fine feature maps FM_{sen}. Notably, the feature pyramid networks used to process reference image and sensed image are the same network by sharing weights. The size of the coarse feature map is $\frac{1}{8}$ of the size of the original image and the size of fine feature maps is $\frac{1}{2}$ of the size of the original image. The coarse feature maps are utilized to generate global context-dependent

feature descriptors. So, its smaller size can reduce the length of input of the Transformer and ensure the computation cost is within an acceptable range. The fine feature maps are utilized to fine-tune the coarse matched points, so its larger size can ensure the higher position accuracy of final matched points.

3.2. Coarse Feature Matching Process

In this process, a set of feature points $(p_0, p_1, \cdots, p_N)$ from a sensed image are utilized to sample coarse feature vectors from coarse feature maps of sensed image. These feature points can be extracted by the off-the-shelf feature extraction algorithm or selected randomly. For each feature point p_i in the sensed image, we sample its coarse feature vector cd_i from the coarse feature map CM_{ref}, then we combine the vector cf_i with the entire coarse feature map CM_{ref} as input to a coarse feature transformer network, obtaining the transformed vector cft_i and the transformed coarse feature maps CMT_{ref}. Finally, we select the position p_i^c in the maps of CMT_{ref} that has the highest correlation with vector cft_i as the coarse matching points for the point p_i.

3.2.1. The Architecture of Coarse Feature Transformer Network

Compared to Convolutional Neural Network (CNN), Transformer can easily capture the global information. Take as an example establishing connections between points that are far apart on the image. For CNN, due to its local connectivity, it requires many stacked convolution layers to build the connection. In fact, the required number of layers is also related to the size of convolution kernel. The smaller the kernel size, the more convolution layers are required, while for Transformer, only one attention layer is enough to build the connection due to its global receptive field. So, the coarse feature Transformer network mainly leverages the excellent characteristic of global attention Transformer to make the feature vectors more context-dependent. Its whole architecture is shown as in Figure 2.

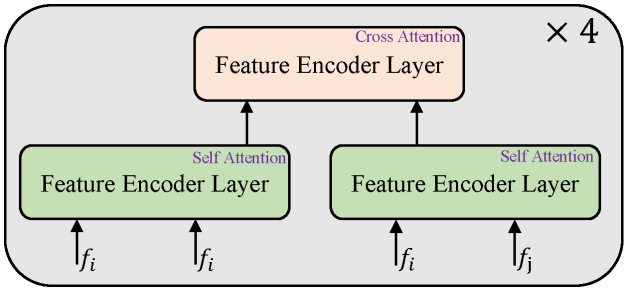


Figure 2. The whole architecture of the coarse feature Transformer network

The structure is primarily comprised of four stacked groups of feature transformation. Each group consists of three feature encoder layers. Within each group, the first two layers are responsible for executing self-attention tasks, while the final layer performs cross-attention tasks. The self-attention task receives input from feature vectors from the same image, while the cross-attention task receives input from feature vectors from different images, namely the reference image and the sensed image. Regarding the architecture of each encoder layer, they are identical and are shown in Figure 3.

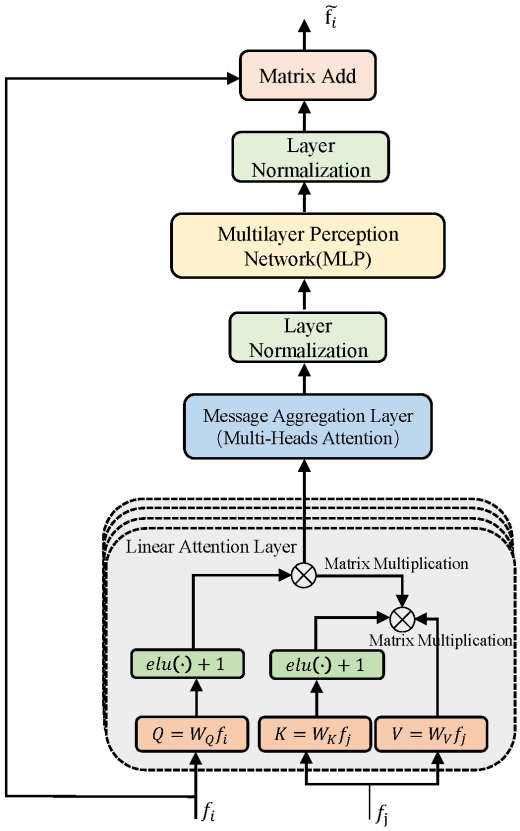


Figure 3. The architecture of feature encoder layer

In the feature encoder layer, its core component is linear attention layer [40], which is an efficient variant of the vanilla attention layer in Transformer. For the sake of simplicity in expression, the input sequences of the attention layers are denoted as query $Q \in \mathcal{R}^{\mathcal{N}\times\mathcal{D}}$, key $K \in \mathcal{R}^{N\times D}$ and value $V \in \mathcal{R}^{N\times D}$, borrowed from the terminology in the field of information retrieval. Here, N denotes the length of sequences and D is the feature dimension. Its math expression is shown in Equation (1):

$$Attention(Q, K, V) = softmax(QK^T)V \tag{1}$$

For the vanilla version of Transformer, its computation cost grows quadratically $(\mathcal{O}(N^2))$ with the length of input sequence N, while for the linear attention layer, its computation cost is reduced to $\mathcal{O}(N)$ by replacing the exponential kernel with an alternative kernel function sim(O,K)=$\phi(Q) \cdot \phi(K)^T$, where $\phi(X) = elu(X) + 1$.

To improve the representation power of the model further, a linear attention layer with d heads is designed. So, each input feature vector f_i, f_j to the attention layer is divided into d parts and fed them into the attention layer simultaneously and outputs d parts, then aggregate them together. The math expression of linear attention layer is shown in Equation (2):

$$LinearAttention(Q, K, V) = \phi(Q)(\phi(K)^T V) \tag{2}$$

In the cross-attention layers of our model, we treat the feature vectors from one image as the query Q, while the feature vectors from the other image are treated as the key K and value V. In contrast, for the self-attention layers, we use the same image's feature vectors for Q, K, and V. After passing through the multi-head linear attention layers, we aggregate the output and normalize the inputs to each layer independently using layer normalization. This helps improve gradient stability during training. We then use a multi-layer perceptron (MLP) network to transform the output further. To incorporate the residual connection

idea of ResNet, we add the original feature vector f_j to the output of the MLP. This final transformed coarse feature vector $\hat{f}_i$ is our model's output.

3.2.2. Coarse Feature Matching

In the stage of coarse feature matching, we design a differential matching layer to address the challenge of picking the nearest neighbor matches in the hand-crafted design model. To determine the corresponding of a picked point $\boldsymbol{x_i}$ in sensed image I_{sen}, we correlated the feature descriptor at $\boldsymbol{x_i}$, denoted by $cft_{sen}^{(i)}$, with all the feature descriptors in CFT_{ref} in image I_{ref}. The result is a set of correlation values that measure the similarity between the feature descriptor $cft_{sen}^{(i)}$ and each feature descriptor in CFT_{ref}. To obtain a spatial distribution over the 2D pixel locations of image I_{ref} that indicate the probability of each location being the correspondence of $\boldsymbol{x_i}$, we apply a 2D softmax operation [41] to the correlation values. We denote this probability distribution as $p(\boldsymbol{x}|x_{sen}^{(i)}; CFT_{ref}; CFT_{sen})$.

$$p(\boldsymbol{x}|x_i; CFT_{ref}; CFT_{sen}) = \frac{\exp({cft_{sen}^{(i)}}^T CFT_{ref}(x))}{\sum_{\boldsymbol{y}\in I_{ref}} \exp({cft_{sen}^{(i)}}^T CFT_{ref}(\boldsymbol{y}))} \tag{3}$$

where $\boldsymbol{y}$ varies over the pixel grid of I_{ref}. The coarse corresponding of $\boldsymbol{x_i}$ in sensed image is calculated as Equation (4):

$$\hat{\boldsymbol{x}}_j = \sum_{\boldsymbol{x}\in I_{ref}} x \cdot p(x|x_i; CFT_{ref}; CFT_{sen}) \tag{4}$$

The differential matching layer makes the entire network end-to-end trainable. Furthermore, because the location of corresponding is calculated by the correlation between feature descriptors, the requirement of its correctness would facilitate coarse feature Transformer network to learn more powerful descriptors. By gathering all the coarse matches, we get a set of coarse matching points $(P_{ref}, P_{sen})^C$.

3.3. Fine Feature Matching Process

In this process, for each pair of coarse matching points (p_i, p_i^c), we refine the location of p_i^c by using a lightweight transformer network to get the fine position of p_i^f finally.

3.3.1. The Architecture of Fine Feature Transformers Network

The architecture of the fine feature transformer network is similar to that of the coarse feature transformer network. Furthermore, the only difference is that the number of stacked groups of feature transformation is 1 while the number of stacked groups of feature transformation is 4 in the fine feature transformers network.

3.3.2. Fine Feature Matching

The flowchart of fine feature matching is shown as Figure 4.

First, for each coarse match (i_c, j_c), we locate its position on the fine feature maps FM_{sen} and FM_{ref}, respectively. Second, we crop out two windows of the same size FM_{ref}^s and FM_{sen}^s, centered at (i_c, j_c), respectively. Third, we use a fine feature Transformer network to generate two fine transformed feature maps FTF_{ref}^s and FTF_{sen}^s. Fourth, we correlate the center vector of FTF_{sen}^s, f_i with all the vectors in FTF_{ref}^s and produce a matching probability map that represents the probability of each pixel in the neighborhood of j_c being matched with i_c. Finally, the finely corresponding $\hat{j}_c$ is calculated as the spatial coordinates expectation over the matching probability distribution. The methods for computing matching probability map and spatial coordinates expectation are analogous to the ones in the coarse feature matching. By Collecting all the matches of $(i_c, \hat{j}_c)$, we obtain the final set of fine matches $(P_{ref}, P_{sen})^F$.

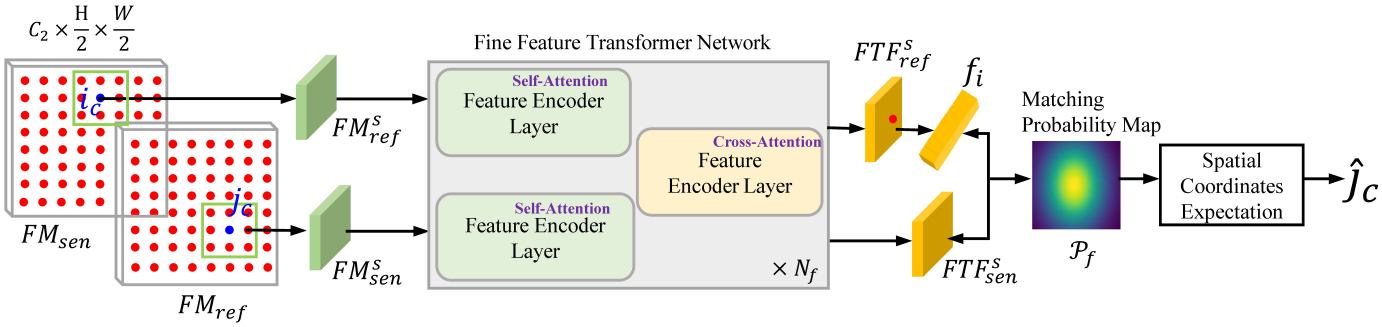


Figure 4. The flowchart of Fine Feature Matching

3.4. Loss function

The process of training a deep neural network involves integrating supervision information into a loss function, which is then minimized to guide the network's learning. Considering that our training data consisted of image pairs with ground-truth homograph transform H, we use two categories of supervision signals—one is the re-projection loss and the other is cycle consistency loss—which encourages the spatial proximity between a point and its forward–backward mapping [42]. Additionally, because our matching process is a coarse-to-fine process, each type of loss function naturally consists of both coarse-level and fine-level losses. So, the total loss function $\mathcal{L}$ has four different terms, namely, $\mathcal{L}^c_{rpt}$, $\mathcal{L}^f_{rpt}$, $\mathcal{L}^c_{cyc}$, $\mathcal{L}^f_{cyc}$.

Coarse-level re-projection loss $\mathcal{L}_c$: for a point x_i in the sensed image, we get a predicted coarse corresponding point x'_i in the reference image, then we calculate the coarse-level re-projection loss as follows (5):

$$\mathcal{L}^c_{rpt}(x_i) = ||x'_i - Hx_i||_2 \tag{5}$$

where H is the ground-truth homograph transformation matrix between image pairs.

Fine-level re-projection loss $\mathcal{L}^f_{rpt}$: for a point x_i in the sensed image, we get a predicted fine corresponding point $\hat{x}_i$ in the reference image, then we calculate the re-projection distance as follows (6):

$$\mathcal{L}^f_{rpt}(x_i) = ||\hat{x}_i - Hx_i||_2 \tag{6}$$

where H is the ground-truth homograph transformation matrix between image pairs.

Coarse-level cycle consistency loss $\mathcal{L}^c_{cyc}$**:** for a point x_i in the sensed image, $\hat{x}_i$ is a fine predicted corresponding point in the reference image, then we calculate the coarse-level cycle consistency loss as follows (7):

$$\mathcal{L}_f(x_i) = ||H_{R\to S}(\hat{x}_i) - x_i||_2 \tag{7}$$

where $H_{R\to S}(x'_i)$ denotes the coarse predicted corresponding point in the sensed image for the point $\hat{x}_i$ in the reference image.

Fine-level cycle consistency loss $\mathcal{L}^f_{cyc}$**:** for a point x_i in the sensed image, $\hat{x}_i$ is a predicted fine corresponding point in the reference image, then we calculate the fine-level cycle consistency loss as follows (8):

$$\mathcal{L}^f_{cyc}(x_i) = ||H'_{R\to S}(\hat{x}_i) - x_i||_2 \tag{8}$$

where $H'_{R\to S}(\hat{x}_i)$ denotes the fine predicted corresponding point in the sensed image for the point x_i in the reference image.

The total loss: For each image pair, its total loss is a weighted sum of coarse-level and fine-level loss, totaled over N sampled points in sensed image. Its expression is shown as Equation (5):

$$\mathcal{L} = \sum_{i=1}^{N} \omega_{rpt}\mathcal{L}^{c}_{rpt}(x_i) + \omega_{rpt}\mathcal{L}^{f}_{rpt}(x_i) + \omega_{cyc}\mathcal{L}^{c}_{cyc}(x_i) + \omega_{cyc}\mathcal{L}^{f}_{cyc}(x_i) \tag{9}$$

where ω_{rpt} and ω_{cyc} are the weight of re-projection loss and cycle consistency terms, respectively.

4. Experiments and Analysis

In this part, we mainly detailed the generation of training data, implementation of network, qualitative experiment and quantative experiments.

4.1. The Generation of Training Data

Here, we propose an almost fully automated method to generate image pairs with ground-truth homograph transformation as label. The cornerstone of automation is that the property of well geometrically alignment between the level-1C products of Sentinel-2 and Level-2 products of Landsat-8. The detailed process is shown as follows:

(1) We deliberately chose a variety of Sentinel-2 images, captured in different years and seasons, and covering diverse scenes such as mountains, farmlands, buildings, roads, and more. Once the selection process was complete, we could use a python script to automatically download the chosen Sentinel-2 images from the Copernicus Open Access Hub.

(2) For each scene in the Sentinel-2 image dataset, we utilized a Python script to download the corresponding Landsat-8 image based on the coordinates of the Sentinel-2 image's center point. After obtaining the pairs of Sentinel-2 and Landsat-8 images, we clipped out their overlapping areas and divided them evenly into 1500×1500 image tiles based on their geographic coordinates using the Geographic Data Abstraction Library (GDAL) [43]. As a result, each pair of image patches is precisely aligned, as the level-1C product of Sentinel-2 and the level-2 product of Landsat-8 undergo rigorous geometric correction.

(3) For each pair of image tiles, we applied random scaling, rotation, and shifting to one tile to obtain a transformed image. A smaller patch with size of 640×640 was then cropped from the transformed tile, which serves as the sensed image and can more realistically mimic the geometric transformation of the image. Because it can avoid the introduction of black areas in the corners. Similarly, a 640×640 image patch centered at the other tile was cropped as the reference image. Finally, we calculated the homography matrix H between the reference and sensed images based on the applied scale, rotation, and displacement. The range of rotation was $(-30°, 30°)$, the range of scaling was $(0.8, 1.25)$, and the range of shift was $(-100, 100)$ pixels.

What is worth noting is that in the real-world scenarios, especially in mountainous areas, the geometric deformation of images can be quite complex due to the influence of terrain variations. Theoretically, a single homography transformation may not fully describe the geometric deformations encountered in such scenarios. However, for remote sensing images, the complexity of geometric deformations between them is significantly reduced after geometric correction using high-resolution Digital Elevation Models (DEMs) [44] and precise positioning systems onboard satellite platforms. In particular, for local area geometric deformations, similarity transformations are often sufficient to describe the geometric transformations [45]. Moreover, for the registration of a whole scene image, a commonly used strategy is block matching, which provides a solid foundation for describing real geometric deformations using homography transformations in practical applications.

4.2. The Implementation Details

Architecture details. The size of input image is 640×640. The number of channels of coarse feature maps and fine feature maps are $N_c = 128$ and $N_f = 128$, respectively. The number of stacked groups of feature transformation is 4 in coarse feature Transformer and 1 in the fine feature Transformer.

Training Dataset. A total of 117 Sentinel-2 remote sensing images are selected and 1704 pairs of image patches are generated. For each sensed image patch, 800 points are extracted. Among these points, 10% of the points are randomly selected, while the remaining 90% of the points are extracted using the SIFT algorithm [6].

Training scheme. Two RTX-TITAN GPUs, each with 24 GB of GPU memory, are utilized to train the network. The Adam optimizer is used with a decay of 0.9 to minimize the loss. The initial learning rate is set to 5×10^{-5}, with a linear warm-up for the first 500 iterations. Gradient clipping is applied with a value of 0.5 to prevent gradient explosions. The weights of ω_{rpt} and ω_{cyc} are set 1.0 and 0.1, respectively. The total number of epochs is set to 200.

4.3. Qualitative Experiments

To validate the effectiveness of the proposed method, two pairs of images (Sentinel-2 and Landsat-8 images) with size of 640×640 pixels are used to test. One pair mainly covers the scene of mountain, and they have large gray differences. The other mainly contains the human-made buildings, and some contents are obscured by cloud. For each pair of images, we adopted two different methods to extract image feature points. One method is to uniformly select 20×20 grid points from the sensed image (Sentinel-2), with a minimum distance of 60 pixels from the image boundary, in order to avoid the possibility of some boundary points not having corresponding points on the reference image. The other method is to extract 400 feature points from the image using the SIFT algorithm. Then, for each extracted image feature point, we used the proposed method to match it with the reference image, and the matching results are shown in Figure 5.

For each pair of images, we calculate the re-projection error of their each matched points based on the true geometric transformation matrix H between them. Here, we set an error threshold σ to be 2. If the re-projection error is less than this threshold, we consider the feature point matching successful, otherwise it is considered a failure. We compute the matching success rate for each pair of images, which is defined as the ratio of the number of correctly matched points to the total number of feature points, and is displayed in the top left corner of the image. In order to visualize the matching effect of feature points better, we connect each pair of matched points with a line, and the color of the line represents the error of the matched points. As the re-projection error gradually increases, the color of the line changes from green to red. Green color represents a very small re-projection error, while red color represents a re-projection error that exceeds twice the error threshold.

From the perspective of the matching success rate metric, the matching success ratio of the proposed method is almost above 80%, especially for the feature points extracted by SIFT, with a matching success rate exceeding 85%. For the uniformly selected feature points, their matching accuracy can also reach or approach to 80%, which demonstrates the effectiveness of the proposed method. At the same time, from the distribution of the matching points, the correctly matched points are almost uniformly distributed on the image, while some matching failures or large matching errors are mostly distributed near the edge of the image or where the image is occluded. This excellent distribution of correctly matched points is also very conducive to the final geometric correction of the image. So, the high matching success rate and uniform distribution of matched points illustrate the effectiveness of the proposed method.

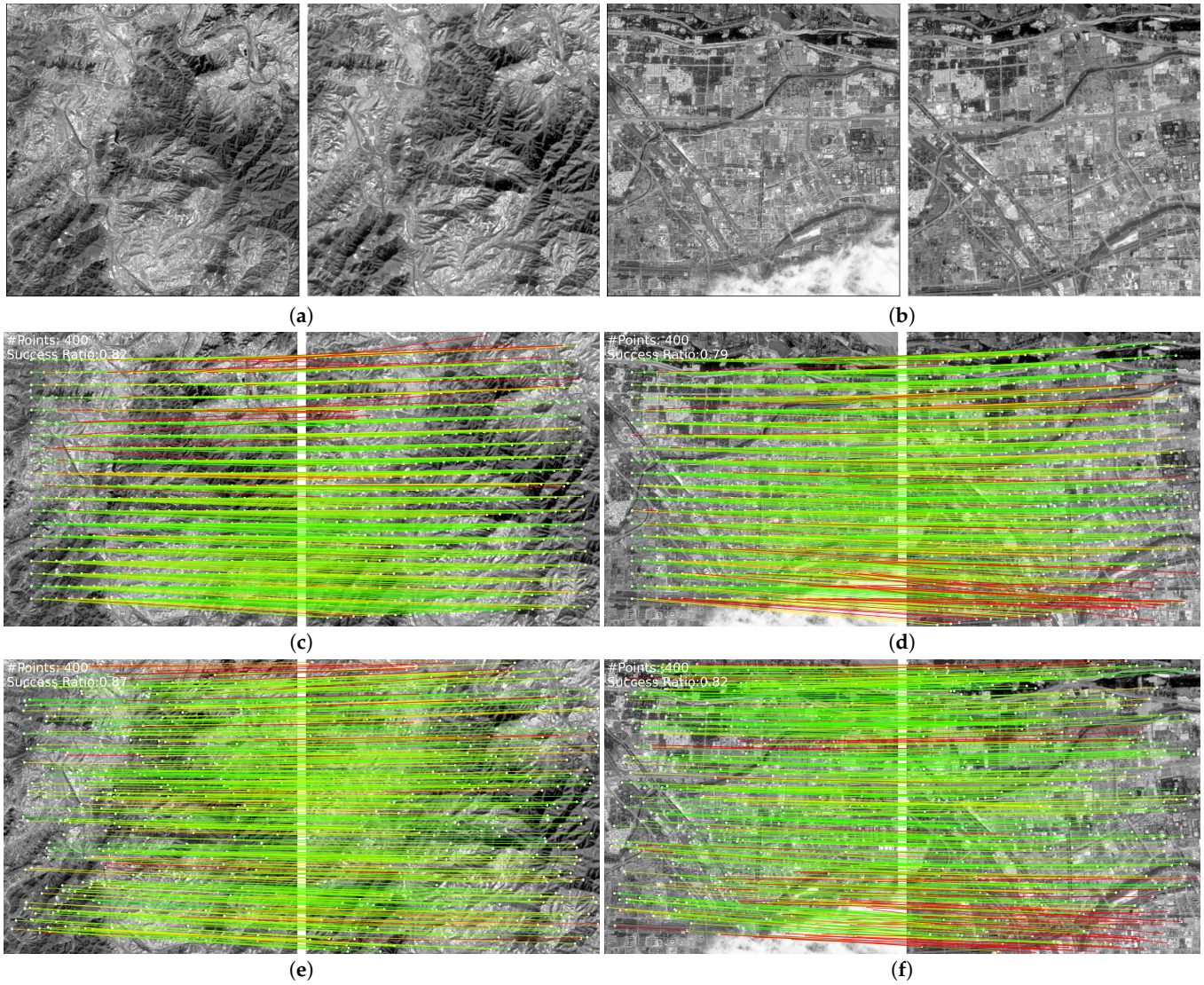


Figure 5. The qualitative experiments of two pairs of images. The color of the line connecting the matching points gradually changes from green to red when its re-projection error gets larger. Green lines represent correctly matched points, while red lines represent incorrectly matched points. Here, the incorrect matched points are defined as these matched points whose re-projection error is larger twice than the error threshold, $\sigma = 2$ pixels. The yellow lines represent errors within the range of $0 \sim 2\sigma$ pixels. At the same time, for each figure with matched points, the total number of points and the matching success rate are shown at the left corner. (**a**) A pair of Sentinel-2 and Landsat-8 Images mainly cover the mountain. (**b**) A pair of Sentinel-2 and Landsat-8 Images mainly cover the man-made buildings. (**c**,**d**) The matching result for uniformly distributed grid points. (**e**,**f**) The matching result for feature points extracted by SIFT algorithm.

4.4. Quantitative Experiments

To evaluate the registration accuracy and performance quantitatively of the proposed method, a total of 340 pairs of images are used as the test data. These images have different acquired time, large gray difference, and diverse content, including mountains, rivers, roads, human-made buildings, farmland, etc., and some image content is partially obscured by thin cloud. Six samples of these image pairs are shown in Figure 6.

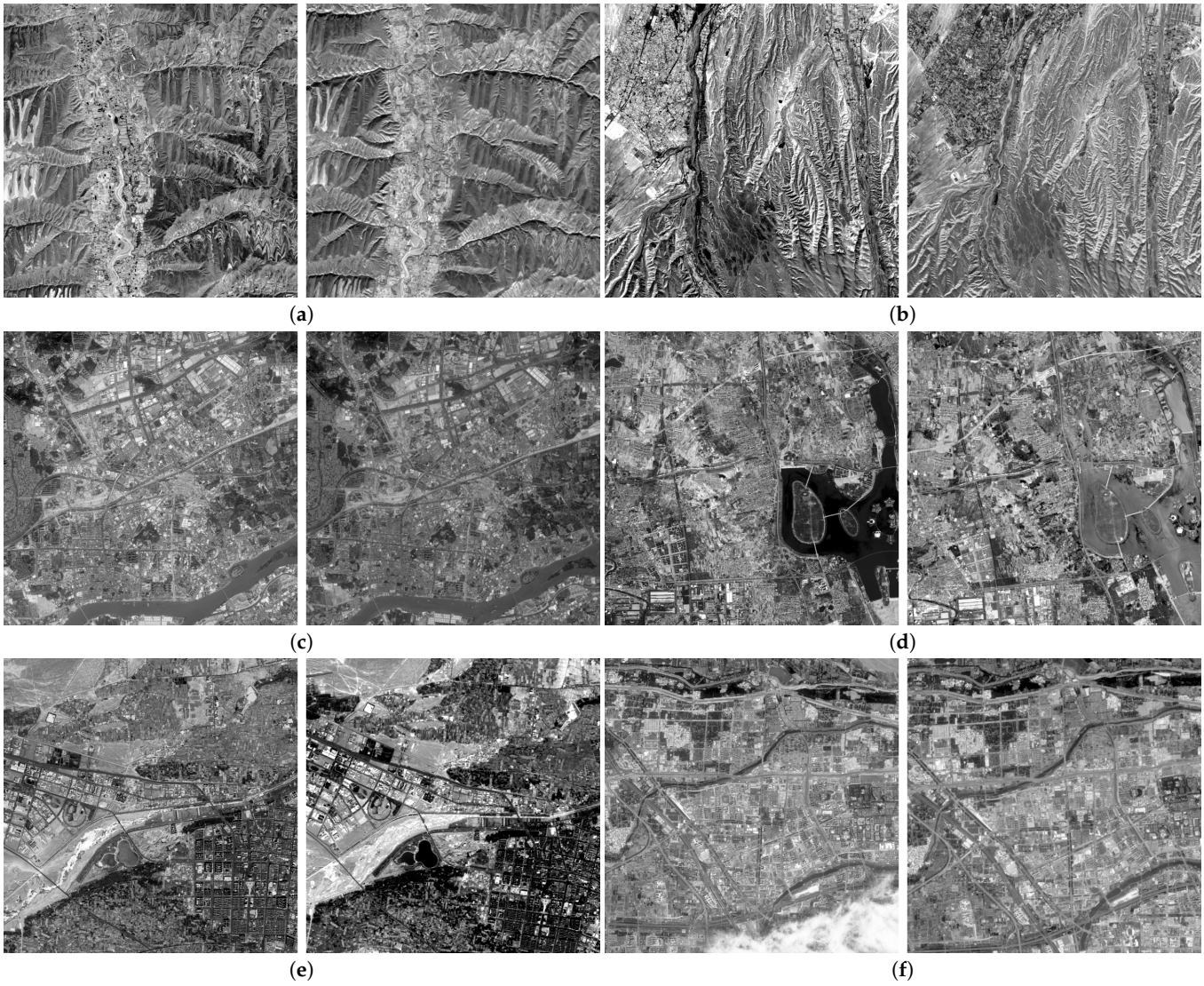


Figure 6. The six samples of test dataset. Image pair (**a**) primarily features large mountains and valleys, while image pair (**b**) depicts smaller mountains with some content changes. Image pairs (**c**) and (**e**) showcase urban areas with rivers and roads, whereas image pair (**d**) depicts farmland with significant grayscale differences across images. Image pair (**f**) focuses mainly on human-made structures, although one of the images is partially obscured by thin cloud.

To quantitatively evaluate the performance of the proposed method, two measurements of mean re-projection error MRE and mean success ratio, SR are adopted. The mean re-projection error of a pair of feature points $(x_S^{(i)}, y_S^{(i)})$ and $(x_R^{(i)}, y_R^{(i)})$ is calculated as follows:

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{(x_{S'}^{(i)} - x_R^{(i)})^2 + (y_{S'}^{(i)} - y_R^{(i)})^2}{2}} \tag{10}$$

where $(x_S^{(i)}, y_S^{(i)})$ is a feature point from the sensed image and $(x_R^{(i)}, y_R^{(i)})$ is a feature point from the reference image, and the re-projected feature point of $(x_S^{(i)}, y_S^{(i)})$ in the sensed image is $(x_{S'}^{(i)}, y_{S'}^{(i)})$ in the reference image. The success ratio SR is calculated as the ratio of

the number of successfully matched feature points $\#MatchedPoints$ to the total number of feature points used to match $\#TotalFeaturepoints$:

$$SR = \frac{\#MatchedPoints}{\#TotalFeaturepoints} \tag{11}$$

The standard for determining the success of feature point matching is whether their re-projection error falls below a specific threshold. A pair of feature points is considered as a successful match when their re-projection error is less than the threshold. For the MRE metric, a smaller value indicates better performance, while for the SR metric, a larger value indicates better performance.

At the same time, three methods are utilized to compare. One is the most popular traditional method SIFT [6] , and the other two algorithms are deep learning based methods superGlue [46] and LoFTR [37]. Although for the SIFT algorithm, given a feature point, can generate its descriptor, but in the actual SIFT matching process, the scale and orientation information of this feature point are also required, and then a corresponding block is cropped based on the position information to generate the descriptor. For the other two deep-learning-based algorithms, they are even more closely related to feature extraction because they are trained through end-to-end training. Specifically, the SIFT algorithm extracts feature points using the Differene of Gaussian (DoG) operator and performs matching based on the ratio of the nearest neighbor distance to the second nearest neighbor distance. The SuperGlue method, on the other hand, utilizes the SuperPoint network [30] to extract feature points and then employs the SuperGlue network for feature matching. As for the LoFTR method [37], its feature points and matching are obtained through joint optimization learning. Therefore, the feature matching process of these three methods is essentially related to feature point extraction. In contrast, our proposed method works completely independently of feature extraction. Given the position information of a point in the image to be registered, we can find its corresponding feature point in another reference image. The weights of SuperPoint, SuperGlue and LoFTR are provided by their respective authors, and their hyper-parameters are also set to default values provided by the authors. The implementation of SuperPoint and SuperGlue are from their authors and the implementation of LoFTR are from the kornia library [47].

However, for each algorithm among these three, the number, distribution, and location of the extracted feature points for the same pair of images are different. To avoid potential performance degradation in the three matching algorithms caused by differences in the extracted feature points, the proposed method, when comparing with each algorithm individually, takes the feature points extracted by each algorithm as the set of candidate matching points and then employs the proposed method for feature points matching. The comparison results of the mean success ratio and mean re-projection error of the proposed method and each algorithm under different threshold settings of 1, 3, 5, 7, and 9 pixels are shown in Table 1. Further, the respective comparison of the boxplot of mean success ratio and mean re-projection Error between the Proposed Method and SIFT, SuperGlue, and LoFTR under different thresholds are shown in Figure 7.

Based on the comparison with the SIFT algorithm, the average SR of the SIFT algorithm is about 0.24, while the average SR of the proposed method is at least 0.39, and as the threshold gradually increases, the average SR of the proposed method will further increase and can exceed 0.5. As expected, the MRE value of the proposed method is larger than that of SIFT, mainly because under the same threshold setting, the proposed method can correctly match far more feature points than SIFT. When the threshold is set to one pixel, the SR of the proposed method is nearly 70% higher than that of SIFT, while its MRE is not higher than 30% of SIFT's MRE. Therefore, this fully demonstrates that the performance of the proposed method is significantly better than that of the traditional SIFT algorithm, that is, the descriptor obtained by comprehensively considering the local and global information of feature points is more discriminative than the descriptor obtained by simply cropping small regions based on the scale and orientation information of feature

points. Compared to the SuperGlue method, a deep learning-based approach, the proposed method has a significantly higher mean success ratio (SR) value, especially at the same threshold setting where the proposed method's average SR can be up to 5 times higher than that of SuperGlue. Furthermore, the proposed method has a smaller mean re-projection error (MRE) than that of SuperGlue, which fully demonstrates that the proposed method outperforms SuperPoint comprehensively. As for the poor performance of the SuperGlue, it is mainly due to its input being two sets of feature points, resulting in its attention to be focused only within the neighborhood of detected feature points.

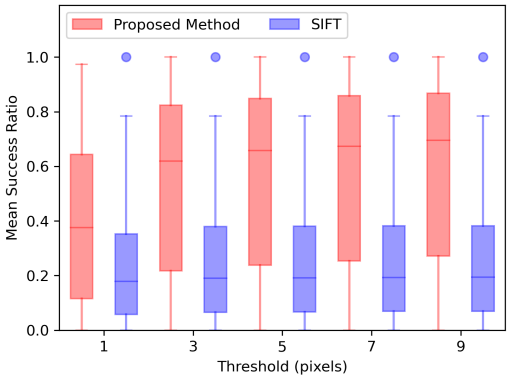
Table 1. The Comparison of Mean Success Ratio and Mean Reprojection Error Under Different Thresholds for Different Methods.

Comparison Experiment	Method	The Mean Success Ratio Under Different Thresholds (pixels)					The Mean Reprojection Error Under Different Thresholds (pixels)				
		@1px	@3px	@5px	@7px	@9px	@1px	@3px	@5px	@7px	@9px
Comparison between SIFT and The Proposed Method	The Proposed Method	0.39	0.53	0.55	0.57	0.58	0.57	1.01	1.27	1.53	1.84
	SIFT	0.23	0.24	0.24	0.24	0.24	0.44	0.67	0.88	1.10	1.33
Comparison between SuperPoint and The Proposed Method	The Proposed Method	0.44	0.64	0.67	0.69	0.70	0.62	1.17	1.54	1.89	2.25
	SuperPoint	0.08	0.11	0.12	0.12	0.13	0.56	0.96	1.17	1.36	1.58
Comparison between LoFTR and The Proposed Method	The Proposed Method	0.32	0.49	0.52	0.53	0.55	0.56	1.01	1.26	1.47	1.69
	LoFTR	0.27	0.38	0.38	0.38	0.38	0.57	1.00	1.13	1.21	1.28

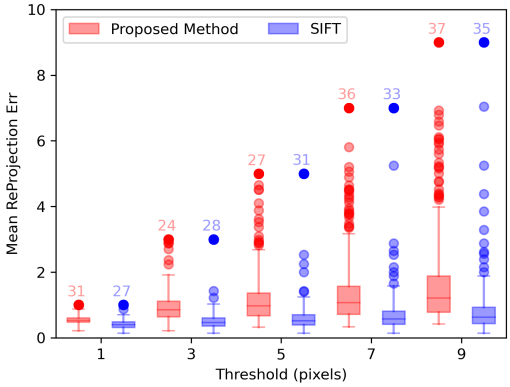
Compared to the latest deep learning method LoFTR, the proposed method has a higher SR value, approximately 20% higher than that of LoFTR. Regarding the MRE metric, when the threshold is larger than 3, the proposed method's MRE is higher than that of LoFTR because the proposed method has a higher average SR, as expected. However, when the threshold is less than or equal to 3, the MRE values of both methods are very close, which fully demonstrates that the proposed method is better than LoFTR.

Based on the boxplot graphs of mean success ratio from various comparative experiments, it can be observed that regardless of the method or threshold setting, there are cases of complete matching failure in the test dataset consisting of 340 image pairs. This indicates that the chosen dataset is quite challenging. Upon examining the test dataset, it is found that the complete matching failures are mainly due to a large extent of cloud coverage in the images or significant content variations caused by temporal differences between the image pairs. Although efforts are made to select images with minimal cloud coverage for the dataset, the subsequent local cropping process inevitably led to situations where a significant portion of the images are obscured by clouds.

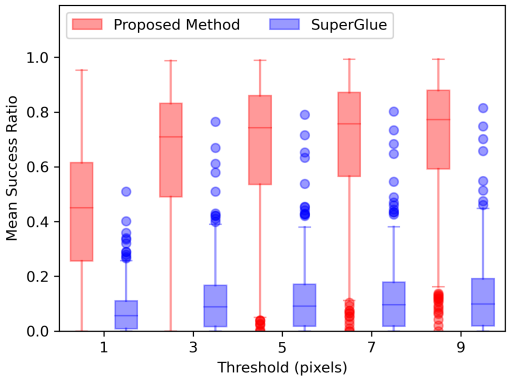
Furthermore, comparing the proposed method with the SIFT method, although the proposed method exhibits a larger interquartile range, the 25th, 50th, and 75th percentiles of the proposed method are significantly higher than those of the SIFT method. A similar trend is observed in the comparison between SuperPoint and SIFT methods. In the case of comparing LoFTR with the proposed method, their quartile sizes are almost similar, but the 25th, 50th, and 75th percentiles of the proposed method are higher than those of LoFTR. Therefore, based on the boxplot of mean success ratio, it can be concluded that the proposed method's mean success ratio is superior to the mean success ratios of the other three methods.



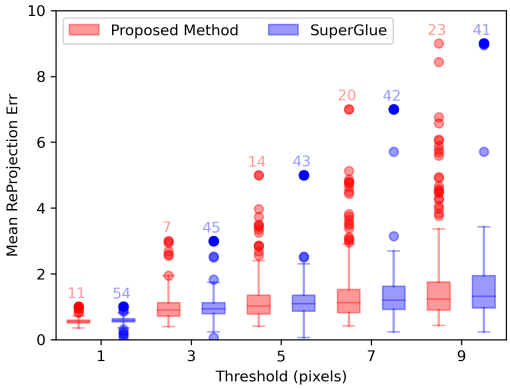
(**a**) The boxplot of mean success ratio between the proposed method and SIFT



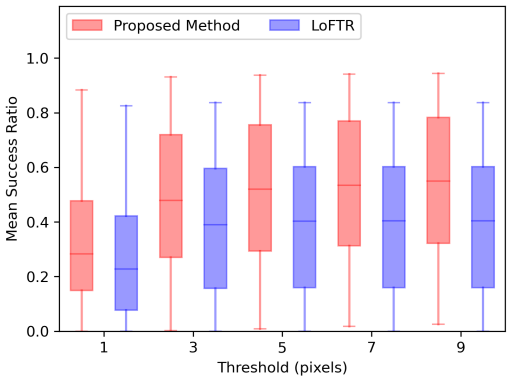
(**b**) The boxplot of mean reprejection error between the proposed method and SIFT



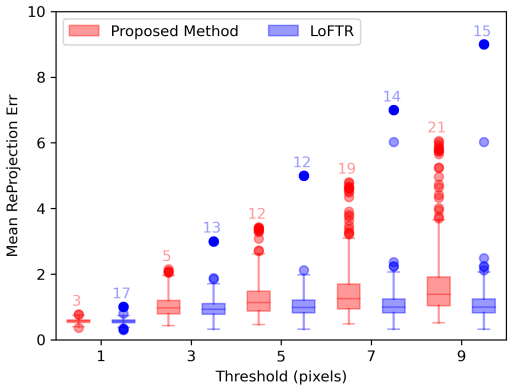
(**c**) The boxplot of mean success ratio between the proposed method and SuperGlue



(**d**) The boxplot of mean reprejection error between the proposed method and SuperGlue



(**e**) The boxplot of mean success ratio between the proposed method and LoFTR



(**f**) The boxplot of mean reprejection error between the proposed method and LoFTR

Figure 7. The respective comparison of the boxplot of mean success ratio and mean re-projection Error between the Proposed Method and SIFT, SuperGlue, and LoFTR under different thresholds. The number appearing above each box in the boxplot of mean Reprejection error represents the number of outliers. (**a**) The boxplot of mean success ratio between the proposed method and SIFT (**b**) The boxplot of mean reprejection error between (**c**) The boxplot of mean success ratio between the proposed method and SuperGlue (**d**) The boxplot of mean reprejection error between (**e**) The boxplot of mean success ratio between the proposed method and LoFTR (**f**) The boxplot of mean reprejection error between the proposed method and LoFTR.

Analyzing the boxplot graphs of mean reprojection error from various comparative experiments, in comparison to the SIFT method, the proposed method exhibits a slightly smaller mean reprojection error, which aligns with our expectations since the proposed method generates more matched points under the same threshold. Comparing the proposed method with SuperGlue, it can be observed that not only does the proposed method have lower 25th, 50th, and 75th percentiles across different thresholds, but it also has significantly fewer outliers exceeding the maximum value, with at least half the number of outliers compared to the SuperGlue method. In the comparison between LoFTR and the proposed method, for thresholds greater than 3, the 25th, 50th, and 75th percentiles of the proposed method are higher than those of LoFTR. However, when it comes to outliers beyond the maximum value, the proposed method has slightly more outliers than LoFTR. Overall, the boxplot graphs of mean reprojection error provide consistent information with the Mean reprojection error values presented in Table 1.

Besides, analyzing the three separate comparison experiments as a whole, we find that the proposed method's mean *SR* and *MRE* values are different because the number and location of the selected feature points are different for the same test dataset. For the proposed method, each experiment's feature points are detected by the compared method. To investigate this further, we conducted another experiment where we uniformly sampled 900 points from each image as feature points and then performed feature matching, the mean success matching success ratio and the mean reprojection error of the proposed method are shown in Table 2 and their boxplot graph of the are shown in Figure 8.

Table 2. The Mean Matching Success Ratio and Mean Reprojection Error of The Proposed Method Under Different Thresholds for 900 Points Sampled Uniformly.

Method	The Mean Success Ratio Under Different Thresholds (pixels)					The Mean Reprojection Error Under Different Thresholds (pixels)				
	@1px	@3px	@5px	@7px	@9px	@1px	@3px	@5px	@7px	@9px
The Proposed Method	0.23	0.35	0.37	0.39	0.40	0.57	1.03	1.30	1.54	1.79

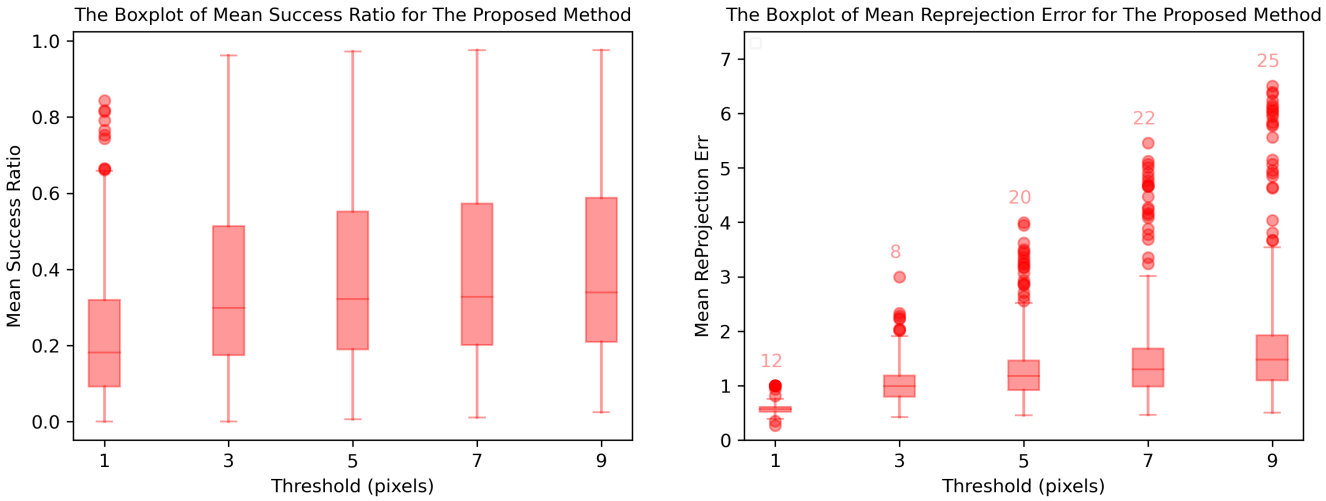


Figure 8. The boxplot of mean matching success ratio and mean reprojection error of the proposed method under different thresholds for 900 points sampled uniformly. The number appearing above each box in the boxplot of mean Reprejection error represents the number of outliers.

Based on the comprehensive comparison of the proposed method's matching success rate under different conditions of obtaining feature points, we find that the proposed method's mean *SR* is the lowest when uniformly selecting feature points, and the highest when SuperPoint extracts feature points. After further analysis, we find that when using uniformly selected feature points, the quality is not high due to differences in the image content, such as partial cloud coverage, which can lead to a low-quality feature point

selection, resulting in the lowest mean matching success rate. From the analysis of the *MRE* metric, we can also conclude that the proposed method's *MRE* is the lowest when using SuperPoint's extracted feature points, which proves that the feature points extracted by SuperPoint and the proposed method are the best combination.

5. Discussion

5.1. Sensitivity Analysis of Feature Points

In the quantitative experiment, we compared the average matching success ratio and average re-projection error to show that the proposed method outperforms the classical SIFT algorithm and current deep learning methods. However, the performance of the proposed method varied under different feature point acquisition ways for the same test dataset, indicating its sensitivity to feature points. During the training, we used mixed feature points, namely with 50% from SIFT algorithm and 50% randomly selected to train the proposed method.

Besides, we compared other three different test feature points acquisition methods, namely, random selection, SIFT extraction and SuperPoint network extraction, and their extracted feature points are named as random, sift and SuperPoint, respectively. At the same time, we trained the proposed method with the three different categories of feature points, namely random, sift and SuperPoint. The proposed method trained by these three different categories of feature points are named as random-trained, sift-trained and SuperPoint-trained method, respectively. The proposed method trained with mixed feature points are referred as mixed-trained methods. For each proposed method trained with different categories of feature points, we calculate its average matching success ratio and average re-projection error under different thresholds for different categories of test feature points. The results are shown in Table 3 and Table 4.

Table 3. The Average Matching Success Ratios of Three Proposed Methods Trained with Different Feature Points in The Case of Four Types of Test Feature Points

Method Trained by Different Feature Points	**The Categories of Test Feature Points**																			
	Random					**Sift**					**Mixed**					**Superpoint**				
	@1px	**@3px**	**@5px**	**@7px**	**@9px**	**@1px**	**@3px**	**@5px**	**@7px**	**@9px**	**@1px**	**@3px**	**@5px**	**@7px**	**@9px**	**@1px**	**@3px**	**@5px**	**@7px**	**@9px**
random-trained	0.80	0.90	0.92	0.93	0.94	0.77	0.86	0.89	0.90	0.91	0.80	0.89	0.92	0.93	0.94	0.40	0.52	0.55	0.57	0.59
sift-trained	0.53	0.72	0.78	0.82	0.84	0.55	0.73	0.78	0.81	0.84	0.52	0.72	0.78	0.81	0.84	0.14	0.25	0.29	0.33	0.36
mixed-trained	0.74	0.86	0.88	0.90	0.91	0.74	0.85	0.87	0.89	0.90	0.74	0.85	0.88	0.90	0.91	0.39	0.51	0.55	0.57	0.59
superpoint-trained	0.74	0.87	0.89	0.90	0.91	0.75	0.88	0.90	0.91	0.92	0.74	0.87	0.89	0.91	0.91	0.74	0.84	0.86	0.87	0.88

Table 4. The Average Re-projection Error of Three Proposed Methods Trained with Different Feature Points in The Case of Four Types of Test Feature Points

Method Trained by Different Feature Points	**The Categories of Test Feature Points**																			
	Random					**Sift**					**Mixed**					**Superpoint**				
	@1px	**@3px**	**@5px**	**@7px**	**@9px**	**@1px**	**@3px**	**@5px**	**@7px**	**@9px**	**@1px**	**@3px**	**@5px**	**@7px**	**@9px**	**@1px**	**@3px**	**@5px**	**@7px**	**@9px**
random-trained	0.41	0.59	0.69	0.77	0.85	0.42	0.61	0.73	0.83	0.93	0.41	0.59	0.69	0.77	0.85	0.52	0.91	1.19	1.46	1.76
sift-trained	0.51	0.92	1.20	1.44	1.67	0.51	0.91	1.20	1.45	1.68	0.52	0.94	1.22	1.46	1.71	0.6	1.32	1.89	2.45	3.06
mixed-trained	0.43	0.65	0.78	0.89	1.00	0.43	0.65	0.79	0.91	1.02	0.43	0.65	0.78	0.89	0.99	0.52	0.93	1.21	1.48	1.79
superpoint-trained	0.45	0.66	0.78	0.88	0.97	0.45	0.66	0.77	0.87	0.96	0.45	0.66	0.78	0.88	0.98	0.41	0.60	0.72	0.84	0.94

When it comes to matching SuperPoint feature points, the SuperPoint-based method stands out by achieving higher matching success rates and smaller reprojection errors. However, the other methods exhibit relatively lower matching success rates. This discrepancy can be attributed to the SuperPoint method's tendency to extract a varying number

of feature points depending on the image content. In our test, we standardized the number of different types of feature points for each image to 900. If the actual number of extracted feature points fell short of 900, we randomly duplicated points to reach the target count. Conversely, if the number exceeded 900, we randomly selected only 900 points. Amongst the methods evaluated, the random-trained approach consistently achieved the highest average matching success rate and the smallest reprojection error when matching non-SuperPoint feature points. The SuperPoint-trained method followed closely behind, obtaining the second-highest matching success rate and the second-smallest reprojection error. Notably, the difference between the SuperPoint-trained and random-trained methods was minimal, with both demonstrating comparable performance across various types of feature points. In contrast, the SIFT-trained method exhibited the lowest matching success rate and the largest reprojection error. The mixed-trained method's performance was slightly inferior to that of the SuperPoint-trained method. This indicates that the proposed network is sensitive to the feature points in the training dataset and that using feature points extracted by the traditional SIFT algorithm as training data hinders the improvement of the network's feature description ability. In fact, this also agrees with the previous understanding that using feature points extracted by existing feature extraction algorithms as training data, especially for feature extraction networks, will not cause the network to learn the areas where the existing feature detection algorithm fails. For our proposed method, using randomly selected feature points as training data has the advantage of being easy to implement and can increase the network's feature description ability, which further illustrates the practicality of the proposed method.

5.2. Efficiency Analysis

To evaluate the efficiency of the proposed method, we primarily analyzed the number of parameters and compared the average time consumed for matching each pair of images on the test dataset. The number of learnable parameters of the proposed method is shown in Table 5 and the average time consumed for matching each pair of images in the test dataset for different methods is shown in Table 6.

Table 5. The number of learnable parameters of the proposed method.

Module Name	Parameters (Floating Point Operations, FLOPs)
Feature Representation Network	5.9 M
Coarse Feature Transformer	5.3 M
Fine Feature Transformer	328 K
Total	11.53 M

Table 6. The average time consumed for matching each pair of images in the test dataset for different methods.

Methods	SIFT	SuperPoint	LoFTR	Proposed Method
Elapsed Time (s)	0.183	0.981	4.552	6.001

In terms of the number of learnable parameters, the model has a relatively small number of model parameters, and the majority of the learnable parameters come from the Feature Representation Network and the Coarse Feature Transformer sub-modules. From the average elapsed time, it can be observed that the traditional SIFT algorithm has the lowest computational cost, significantly lower than the other deep-learning-based methods by an order of magnitude. Among these three deep learning methods, SuperPoint demonstrates notably lower computational cost compared to the other two. This is mainly due to the fact that both LoFTR and the proposed method employ the Transformer network architecture, which involves highly dense computations, particularly in the operation of

scaled dot product attention on query, key, and value tensors, which is computationally expensive. However, considering the excellent matching performance of the proposed method, it remains competitive. Additionally, with the growing popularity of Transformer network architecture, popular software and hardware optimizations are expected to further improve the computational efficiency of this algorithm. For instance, the recently released deep learning framework PyTorch 2.0 has specifically optimized this algorithm, thereby further enhancing the computational efficiency of Transformers.

5.3. Pros and Cons Analysis

The advantage of the proposed method is its ability to incorporate a larger range of neighborhood information, and even global information, into the generation of feature descriptors by leveraging the excellent global contextual capabilities of the transformer. This results in feature descriptors that have higher robustness and discriminability, significantly improving the success rate of feature matching. Additionally, the method offers high flexibility as it can be combined with any specific feature extraction algorithm independently. However, in our experiments, we found that the matching success rate of the proposed method dramatically decreases when the overlap between images is low, especially when it is below 30%. Through in-depth analysis, we discovered that when the image overlap is small, incorporating a larger range of neighborhood information introduces more noise to the generated descriptors. This leads to low signal-to-noise ratio in the generated descriptors, ultimately resulting in matching failures. Furthermore, the computational efficiency of the proposed method needs further improvement, particularly regarding the large computational cost of scale-dot operations in the transformer for query, key, and value computations.

6. Conclusions

In this paper, we propose a flexible feature matching network that utilizes the excellent global attention capability of the Transformer to address the low matching success rate caused by feature descriptors generated solely from local regions in existing feature matching processes. We also introduce an almost fully automated remote sensing image matching dataset, which effectively trains the network, enabling it to achieve significantly improved feature matching success rates without being limited by specific feature extraction algorithms. Qualitative experiments validate the effectiveness of the proposed method, while quantitative experiments, using average matching success rate and average reprojection error as evaluation metrics, compare it with traditional classical algorithms and state-of-the-art deep learning-based algorithms, demonstrating its superior performance. Finally, further ablative experiments are conducted to analyze the sensitivity of the proposed network to feature selection, concluding that randomly selecting feature points as training data is more beneficial for enhancing the network's feature matching ability, further confirming the practicality of the proposed method. In future research, we will investigate how to further improve the network's feature matching performance under larger geometric transformations, especially in situations where there is minimal overlap between images. In addition, we will also explore how to adapt this network for matching SAR and Optical images.

Author Contributions: Conceptualization, C.L. and Y.D.; methodology, Y.D. and C.L.; validation, C.L., C.Z. and Y.D.; formal analysis, Y.D. and C.L.; investigation, C.L., Y.D. and Z.S.; writing—original draft preparation, Y.D. and C.L.; visualization, C.L., Y.D. and Z.S.; supervision, Y.D.; funding acquisition, Y.D. All authors have contributed significantly and have participated sufficiently to take responsibility for this research. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Nature Science Foundation of China (No. 62001275).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; Yan, J. Image matching from handcrafted to deep features: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 23–79. [CrossRef]
2. Paul, S.; Pati, U.C. A comprehensive review on remote sensing image registration. *Int. J. Remote. Sens.* **2021**, *42*, 5396–5432. [CrossRef]
3. Liang, J.; Liu, X.; Huang, K.; Li, X.; Wang, D.; Wang, X. Automatic registration of multisensor images using an integrated spatial and mutual information (SMI) metric. *IEEE Trans. Geosci. Remote. Sens.* **2013**, *52*, 603–615. [CrossRef]
4. Maes, F.; Collignon, A.; Vandermeulen, D.; Marchal, G.; Suetens, P. Multimodality image registration by maximization of mutual information. *IEEE Trans. Med. Imaging* **1997**, *16*, 187–198. [CrossRef]
5. Xiang, Y.; Tao, R.; Wan, L.; Wang, F.; You, H. OS-PC: Combining feature representation and 3-D phase correlation for subpixel optical and SAR image registration. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *58*, 6451–6466. [CrossRef]
6. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
7. Ye, Y.; Shen, L. HOPC: A novel similarity metric based on geometric structural properties for multi-modal remote sensing image matching. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2016**, *3*, 9. [CrossRef]
8. Li, J.; Hu, Q.; Ai, M. RIFT: Multi-modal image matching based on radiation-variation insensitive feature transform. *IEEE Trans. Image Process.* **2019**, *29*, 3296–3310. [CrossRef]
9. Xiang, Y.; Wang, F.; Wan, L.; You, H. SAR-PC: Edge detection in SAR images via an advanced phase congruency model. *Remote Sens.* **2017**, *9*, 209. [CrossRef]
10. Xiang, Y.; Tao, R.; Wang, F.; You, H.; Han, B. Automatic registration of optical and SAR images via improved phase congruency model. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2020**, *13*, 5847–5861. [CrossRef]
11. Zhu, B.; Yang, C.; Dai, J.; Fan, J.; Qin, Y.; Ye, Y. R_2FD_2: Fast and Robust Matching of Multimodal Remote Sensing Images via Repeatable Feature Detector and Rotation-invariant Feature Descriptor. *IEEE Trans. Geosci. Remote. Sens.* **2023**. [CrossRef]
12. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; Volume 15, pp. 10–5244.
13. Cohen, T.; Welling, M. Group Equivariant Convolutional Networks. In Proceedings of the 33rd International Conference on Machine Learning, PMLR, 20–22 June 2016; Volume 48, pp. 2990–2999.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, USA 2016; pp. 770–778.
15. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
16. Yi, K.M.; Trulls, E.; Lepetit, V.; Fua, P. LIFT: Learned invariant feature transform. In Proceedings of the European Conference on Computer Vision, Springer, Amsterdam, The Netherlands, 11–14 October 2016; pp. 467–483.
17. Ono, Y.; Trulls, E.; Fua, P.; Yi, K.M. LF-Net: Learning local features from images. In Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018.
18. Revaud, J.; De Souza, C.; Humenberger, M.; Weinzaepfel, P. R2D2: Reliable and repeatable detector and descriptor. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
19. Choy, C.B.; Gwak, J.; Savarese, S.; Chandraker, M. Universal correspondence network. In Proceedings of the Advances in Neural Information Processing Systems 29 (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
20. Tian, Y.; Fan, B.; Wu, F. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 661–669.
21. Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Fua, P.; Moreno-Noguer, F. Discriminative learning of deep convolutional feature point descriptors. In Proceedings of the IEEE international Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 118–126.
22. Mishchuk, A.; Mishkin, D.; Radenovic, F.; Matas, J. Working hard to know your neighbor's margins: Local descriptor learning loss. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
23. He, K.; Lu, Y.; Sclaroff, S. Local descriptors optimized for average precision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 596–605.
24. Schmidt, T.; Newcombe, R.; Fox, D. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robot. Autom. Lett.* **2016**, *2*, 420–427. [CrossRef]
25. Melekhov, I.; Tiulpin, A.; Sattler, T.; Pollefeys, M.; Rahtu, E.; Kannala, J. Dgc-net: Dense geometric correspondence network. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, Waikoloa Village, HI, USA, 7–11 January 2019; pp. 1034–1042.
26. Truong, P.; Danelljan, M.; Timofte, R. GLU-Net: Global-local universal network for dense flow and correspondences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6258–6268.
27. Rocco, I.; Cimpoi, M.; Arandjelović, R.; Torii, A.; Pajdla, T.; Sivic, J. Neighbourhood consensus networks. In Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018.

28. Truong, P.; Danelljan, M.; Gool, L.V.; Timofte, R. GOCor: Bringing globally optimized correspondence volumes into your neural network. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 14278–14290.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
30. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA 18–22 June 2018; pp. 224–236.
31. Balntas, V.; Lenc, K.; Vedaldi, A.; Mikolajczyk, K. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5173–5182.
32. Yuan, X.; Yuan, X.; Chen, J.; Wang, X. Large Aerial Image Tie Point Matching in Real and Difficult Survey Areas via Deep Learning Method. *Remote Sens.* **2022**, *14*, 3907. [CrossRef]
33. Liu, Y.; Gong, X.; Chen, J.; Chen, S.; Yang, Y. Rotation-invariant siamese network for low-altitude remote-sensing image registration. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2020**, *13*, 5746–5758. [CrossRef]
34. Ye, F.; Su, Y.; Xiao, H.; Zhao, X.; Min, W. Remote sensing image registration using convolutional neural network features. *IEEE Geosci. Remote. Sens. Lett.* **2018**, *15*, 232–236. [CrossRef]
35. Xu, Z.; Zhang, W.; Zhang, T.; Yang, Z.; Li, J. Efficient transformer for remote sensing image segmentation. *Remote Sens.* **2021**, *13*, 3585. [CrossRef]
36. Li, Q.; Chen, Y.; Zeng, Y. Transformer with transfer CNN for remote-sensing-image object detection. *Remote Sens.* **2022**, *14*, 984. [CrossRef]
37. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-free local feature matching with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8922–8931.
38. Krzysztof, C.; Valerii, L.; David, D.; Xingyou, S.; Andreea, G.; Tamas, S.; Peter, H.; Jared, D.; Afroz, M.; Lukasz, K.; et al. Rethinking attention with performers. In Proceedings of the of ICLR, Virtual Event, 3–7 May 2021.
39. Zaheer, M.; Guruganesh, G.; Dubey, K.A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big bird: Transformers for longer sequences. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17283–17297.
40. Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In Proceedings of the International Conference on Machine Learning. PMLR, Virtual, 13–18 July 2020; pp. 5156–5165.
41. Bengio, Y.; Goodfellow, I.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2017; Volume 1.
42. Wang, X.; Jabri, A.; Efros, A.A. Learning correspondence from the cycle-consistency of time. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2566–2576.
43. GDAL/OGR Contributors. *GDAL/OGR Geospatial Data Abstraction Software Library*; Open Source Geospatial Foundation: Chicago, IL, USA, 2022.
44. Franks, S.; Storey, J.; Rengarajan, R. The new landsat collection-2 digital elevation model. *Remote Sens.* **2020**, *12*, 3909. [CrossRef]
45. Fraser, C.S.; Dial, G.; Grodecki, J. Sensor orientation via RPCs. *ISPRS J. Photogramm. Remote Sens.* **2006**, *60*, 182–194. [CrossRef]
46. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4938–4947.
47. Riba, E.; Mishkin, D.; Ponsa, D.; Rublee, E.; Bradski, G. Kornia: An open source differentiable computer vision library for pytorch. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020; pp. 3674–3683.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.