MOATNET: REGISTRATION FOR MULTI-TEMPORAL OPTICAL REMOTE SENSING IMAGES USING DEEP CONVOLUTIONAL FEATURES

Chao Li, Yanan You, Jingyi Cao, Wenli Zhou

School of Artificial Intelligence, Beijing University of Posts and Telecommunications

IGARSS 2021 - 2021 IEEE International Geoscience and Remote Sensing Symposium | 978-1-6654-0369-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/IGARSS47720.2021.9553197

ABSTRACT

Image registration is an important technique that has been widely used in many areas. It is an indispensable premise for remote sensing image tasks like change detection and image fusion. In this paper, we propose a deep learning framework to generate descriptors for key points and then combine the descriptors constructed with FAST key points for accurate image registration. During the process of training, we adopt a novel loss function named Moat Loss (ML) to train our model, which is accordingly called MoatNet. Experiments show that our method is more robust than traditional algorithms like SIFT and is more accurate than the end-to-end deep learning methods in much more complex cases.

Index Terms— Image registration, CNN, MoatNet, FAST

1. INTRODUCTION

Image registration refers to the process of aligning two or more images with certain spatial relationship. It is an essential prerequisite for many other remote sensing image tasks such as change detection, image fusion, etc.

So far, many approaches have been proposed to address the issue of image registration, and these methods can be divided into two categories, one of which is based on intensity information and the other is based on features extracted from the images. The intensity-based methods, such as mutual information (MI), cross correlation (CC) and sequential similarity detection algorithm (SSDA), try to use the intensity information to measure the similarity between two images and find the optimal mapping that maximizes it. These methods may be accurate in a way, but are sensitive to intensity changes and are limited to high computational complexity. In contrast, the feature-based methods utilize the features refined from the images including points, lines, edges, contours, etc., and then register two images by matching these features. Compared to the intensity-based methods, the feature-based methods like scale-invariant feature transform (SIFT) [1] and speeded up robust features (SURF) [2] have proved to be more robust and effective, and are more popular in the field of image registration.

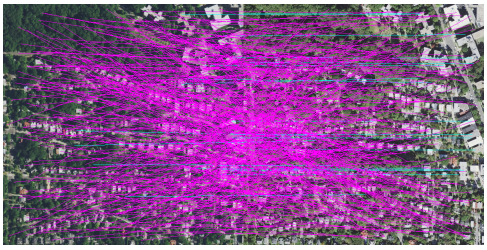


Fig. 1: The result of SIFT key points matching using the nearest neighbor algorithm. The cyan lines refer to the correct correspondences while the magenta lines refer to the incorrect ones. The outliers are far more than inliers.

However, for multi-temporal remote sensing image registration, due to the non-linear discrepancies caused by factors like weather, season, illumination, etc., if we directly use traditional feature-based algorithms, there can be numerous outliers tending to cause registration error, as shown in Fig.1. In recent years, with the success of convolutional neural network (CNN) in computer vision, it has been used to cover the shortages of traditional handcrafted features or even replaces them totally. Yang *et al.* [3] used VGG-16 [4] to generate descriptors and designed a gradually increasing selection of inliers to improve robustness. Ma *et al.* [5] proposed a method using deep learning features for prior coarse registration, which aimed to correct the SIFT outliers. Ye *et al.* [6] combined the deep CNN features with the SIFT features as new descriptors. Except for the above methods that tried to make up the shortcomings of handcrafted features with CNN, Park *et al.* [7] proposed an end-to-end network for aerial image matching without any assistance of handcrafted features. Nevertheless, common CNN features are usually not distinguishable enough when the overlapping areas of the input image patches are large, whereas the end-to-end deep learning methods are neither easy to train nor accurate enough.

To make the descriptors more distinguishable, we develop a novel loss function to force the model to learn how to discriminate the correct correspondent point from its incorrect neighbors. Furthermore, we draw a buffer zone on the feature map which does not contribute to the loss function to improve the robustness and make training easier. Because it's like a *moat* surrounding the center castle, we name the loss func-

tion Moat Loss (ML), and the framework is called MoatNet. Experiments show that MoatNet features outperform handcrafted features like SIFT when applied in multi-temporal remote sensing image registration.

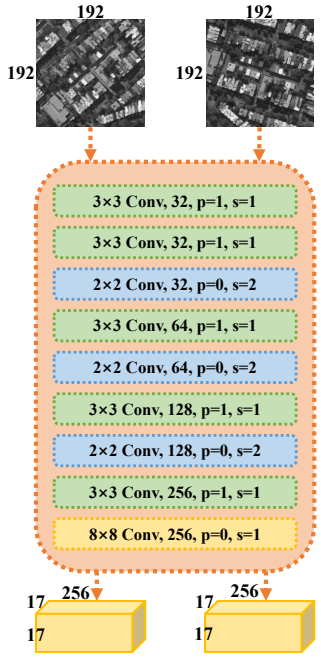


Fig. 2: The architecture of MoatNet, where p refers to padding and s refers to stride.

2. METHOD

Common CNN features are usually not distinguishable enough when the overlapping areas of the input image patches are large. To relieve the issue, we propose a novel loss function ML in this paper to train the convolutional neural network which is used to construct descriptors, and force it to learn to discriminate the correct correspondent point from its incorrect neighbors. Furthermore, we combine the FAST [8] key points with the MoatNet descriptors for image registration process. More details will be introduced in the following parts.

2.1. MoatNet

The architecture of MoatNet is shown in Fig.2, which is used to describe key points along with the image patches centered on them. There are two streams that share the same parameters during the training process. Each input sample consists of two image patches whose sizes are 192×192. For each input image patch, the MoatNet will output a tensor of $256 \times 17 \times 17$. As shown in Fig. 2, the two input image patches are from different scenes but share the same geographic center. To ensure the scale-invariance and rotation-invariance, there exist random rotation and scaling transformation around the geometric centers of the two input image patches.

There is a batch normalization layer after each convolutional layer. In addition, except for the last convolutional layer, each one is followed by a ReLU activation function. For each $256-d$ feature vector on the output tensor, we use L2 normalization to unitize it so that its length is fixed to 1.

2.2. Sample labeling

Suppose that there are two registered remote sensing images acquired at different times, denoted as I_1 and I_2. For each key point denoted as $(x, y)_k$ where k refers to the sample index, we randomly affine the two images with two random scaling-rotation matrixes M_1 and M_2 of 2×2, denoted as $I_1^{'}$ and $I_2^{'}$ respectively. Simultaneously, we can find the corresponding points of $(x, y)_k$ under M_1 and M_2, denoted as $(x', y')_k^1$ and $(x', y')_k^2$ respectively.

$$\begin{aligned} ((x', y')_k^1)^T &= M_1((x, y)_k)^T \\ ((x', y')_k^2)^T &= M_2((x, y)_k)^T \end{aligned} \tag{1}$$

If $(x', y')_k^1$ and $(x', y')_k^2$ are far enough from the image borders, image patches P_k^1 and P_k^2 are then cropped from $I_1^{'}$ and $I_2^{'}$ with sizes of 192×192, which are centered on $(x', y')_k^1$ and $(x', y')_k^2$ respectively. P_k^1 and P_k^2 make up a sample of our dataset.

The two image patches share the same geometrical center, therefore the center feature vector of the output tensor of patch P_k^1 should be mapped to that of the output tensor of patch P_k^2. In other words, $F(W, P_k^1)_{9,9}$ is corresponding to $F(W, P_k^2)_{9,9}$, where W refers to the weights of the MoatNet.

2.3. MoatNet Loss (ML)

Suppose the input batch of the MoatNet is denoted as $X = (P_k^1, P_k^2),\ k = 1, 2, \cdots, N$, where N is batch size. The output of (P_k^1, P_k^2) through the MoatNet is denoted as $(F(W, P_k^1), F(W, P_k^2))$. According to the model construcure shown in Fig.2, we can infer that the sizes of $F(W, P_k^1)$ and $F(W, P_k^2)$ are both $256 \times 17 \times 17$, and we use $F(W, P_k^l)_{i,j}$ to represent the 256-d feature vector on the i-th row and j-th column of the output tensor. Obviously, $F(W, P_k^l)_{9,9}$ is the center feature vector on $F(W, P_k^l)$. Therefore, we can define,

$$\begin{aligned} D_1(k) &= d(F(W, P_k^1)_{9,9}, F(W, P_k^2)_{9,9}) \\ D_2(k, q, r) &= \min_{i,j \in A \cap \mathbb{N}} d(F(W, P_k^q)_{9,9}, F(W, P_k^r)_{i,j}) \\ \delta(k, q, r) &= \min(0, \varepsilon + D_1(k) - D_2(k, q, r)) \end{aligned} \tag{2}$$

where, $\varepsilon = 1.0,\ q, r \in \{1, 2\}$, and,

$$d(x, y) = \left\| \frac{x}{\|x\|_2} - \frac{y}{\|y\|_2} \right\|_2 \tag{3}$$

$$A = \{n \mid 1 \le n \le 17, n - 9 < -m \vee n - 9 > m\} \tag{4}$$

then ML is defined as,

$$ML = \frac{1}{N}\sum_{k=1}^{N}(\delta(k,1,2)+\delta(k,2,1)) \tag{5}$$

where N is the batch size.

In this paper, m in Eq. (4) is an integer less than 9, which represents the width of the buffer zone, also known as *moat*. It determines the matching error tolerance. When m is smaller, the constraint of ML is more strict and the model is easier to be overfitted. In this paper, we set $m = 2$, which represents 16 pixels error tolerance mapping to original images.

2.4. Training

During the process of training, the two streams share the same weights. After obtaining the outputs of our model, we then calculate the ML using the method in 2.3 and use stochastic gradient descent (SGD) to optimize the parameters of our model. In this paper, we set the learning rate $lr = 0.005$ and the momentum $mmt = 0.9$. Besides, dropout layer and weight decay are also adopted to inhibit overfitting.

2.5. Registration using MoatNet descriptors

Suppose that there are two images I_m and I_s to be registered, where I_m refers to the master image and I_s refers to the slave image. We firstly utilize FAST algorithm with NMS to detect the key points in the two images, and eliminate the points that are too close to the image borders, which are not suitable for cropping. The two key point sets are denoted as,

$$\begin{aligned} S_m &= \{(x,y)_1^1, \cdots, (x,y)_Q^1\} \\ S_s &= \{(x,y)_1^2, \cdots, (x,y)_R^2\} \end{aligned} \tag{6}$$

In order to relieve the computational burden, NMS is applied to ensure that there is certain distance between each two key points in S_m and S_s. For each $(x,y)_q^1$ in S_m, we crop an image patch from I_m with the center of $(x,y)_q^1$ and size of 96×96, denoted as B_q. The size of B_q is the same as the size of the receptive field of the output feature. Feed the MoatNet with B_q, and we have $F(W, B_q)_{2,2}$, which is the center feature vector of $F(W, B_q)$. Feed the MoatNet with I_s and we have $F(W, I_s)$. Let,

$$D_{q,i,j} = d(F(W,B_q)_{2,2}, F(W,I_s)_{i,j}) \tag{7}$$

Suppose that,

$$\begin{aligned} a_q, b_q &= \arg\min_{i,j} \ D_{q,i,j} \\ c_q, d_q &= \arg\min_{\substack{i<a_q-m \vee i>a_q+m \\ j<b_q-m \vee j>b_q+m}} D_{q,i,j} \end{aligned} \tag{8}$$

where m is the *moat* width, which equals 2 here. If,

$$D_{c_q,d_q} - D_{a_q,b_q} \geq \epsilon \tag{9}$$

we then calculate the center point of the receptive field of feature vector $F(W, I_s)_{a_q,b_q}$, which is,

$$(\hat{x}, \hat{y}) = (32 + 8(b_q - 1), 32 + 8(a_q - 1)) \tag{10}$$

Then, we have a subset C of S_s, where for any $(x,y)_r^2$ in C, the Euclidean distance between $(x,y)_r^2$ and $(\hat{x}, \hat{y})$ is no more than 8. Subsequently, select the $(x,y)_r^2$ in C with the highest response, and assume that $(x,y)_q^1$ and $(x,y)_r^2$ make up a correspondence. With all correspondences collected, we finally use RANSAC to remove possible outliers and calculate the homographic matrix H for image registration.

Table 1: The RMSE using three methods

RMSE \ Group / Method	Group 1	Group 2	Group 3
SIFT+RANSAC	N/A	2.7761(px)	**1.2238**(px)
[7]	14.1487(px)	49.5706(px)	75.8712(px)
Ours+RANSAC	**3.4180(px)**	**1.7252(px)**	1.9282(px)

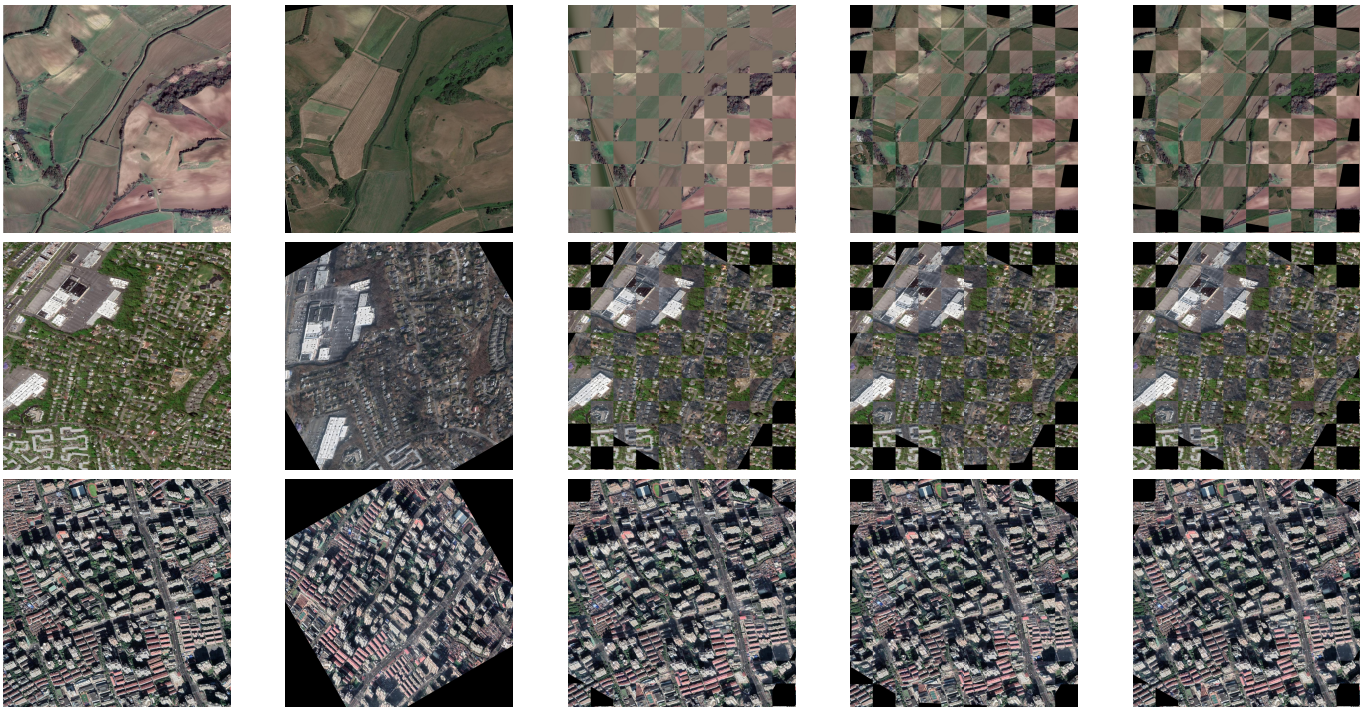
3. EXPERIMENTS

We use 3 groups of remote sensing images to test our method. Each group contains two images acquired from different times. For each group, the two images are both of size 1024×1024. The registration results using our method as well as SIFT and the method in [7] are shown in Fig.3. Besides, we also use root mean square error (RMSE) to measure the registration accuracy of our methods along with SIFT and the method in [7], which is shown in Table 1. During the experiments, we fix our threshold $\epsilon = 0.15$ in Eq. (9). For evaluation of SIFT, we fix the ratio of the minimal distance and the next minimal distance $ratio = 0.75$.

As can be seen in Fig. 3, we can infer that our method outperforms conventional handcrafted methods like SIFT in more complicated multi-temporal remote sensing image registration problems such as Group 1. In other words, learnable features can be more rational than handcrafted features in a way. In addition, with the ML loss function and FAST key points, we can see that our method performs closely to SIFT in relatively simple cases such as Group 2 and Group 3. When compared to the end-to-end method like [7], both our method and SIFT are more accurate, which indicates that feature-based methods are more accurate than end-to-end deep learning methods in general.

4. CONCLUSION

In this paper, we proposed a deep convolutional framework named MoatNet and use it to describe FAST key points instead of traditional descriptors for multi-temporal optical remote sensing image registration. In order to consolidate the



(a) Master (b) Slave (c) SIFT (d) [7] (e) MoatNet (Ours)

Fig. 3: Registration results using SIFT, method in [7] and MoatNet. Row 1-3 refer to Group 1-3 respectively.

distinguishability for descriptors when the overlapping areas of the input image patches are large, we design a novel ML loss function to force the MoatNet to learn to discriminate the correctly matched correspondences from the incorrect neighbors. Experiments show that our methods performs better than traditional feature-based methods like SIFT in complicated muti-temporal remote sensing image registration cases and are more accurate than end-to-end deep learning methods.

5. ACKNOWLEDGEMENT

This work is Supported by Beijing Natural Science Foundation, China (Grant No. 4214058).

6. REFERENCES

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[3] Zhuoqian Yang, Tingting Dan, and Yang Yang, "Multi-temporal remote sensing image registration using deep convolutional features," *IEEE Access*, vol. 6, pp. 38544–38555, 2018.

[4] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] Wenping Ma, Jun Zhang, Yue Wu, Licheng Jiao, Hao Zhu, and Wei Zhao, "A novel two-step registration method for remote sensing images based on deep and local features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4834–4843, 2019.

[6] Famao Ye, Yanfei Su, Hui Xiao, Xuqing Zhao, and Weidong Min, "Remote sensing image registration using convolutional neural network features," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 232–236, 2018.

[7] Jae-Hyun Park, Woo-Jeoung Nam, and Seong-Whan Lee, "A two-stream symmetric network with bidirectional ensemble for aerial image matching," *Remote Sensing*, vol. 12, no. 3, pp. 465, 2020.

[8] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.