DeFusion: Aerial Image Matching Based on Fusion of Handcrafted and Deep Features

Xianfeng Song, Yi Zou, Zheng Shi, Yanfeng Yang and Dacheng Li

October 23, 2023

DeFusion: Aerial Image Matching Based on Fusion of Handcrafted and Deep Features

Xianfeng Song[1], Yi Zou[1,*], Zheng Shi[1], Yanfeng Yang[1], and Dacheng Li[2]

[1] South China University of Technology, Guangzhou, China
[2] Gosuncn Technology Group CO., LTD, Guangzhou, China
[*] zouyi@scut.edu.cn

Abstract. Machine vision has become a crucial method for drones to perceive their surroundings, and image matching, as a fundamental task in machine vision, has also gained widespread attention. However, due to the complexity of aerial images, traditional matching methods based on handcrafted features lack the ability to extract high-level semantics and unavoidably suffer from low robustness. Although deep learning has potential to improve matching accuracy, it comes with the high cost of requiring specific samples and computing resources, making it infeasible for many scenarios. To fully leverage the strengths of both approaches, we introduce DeFusion, a novel image matching scheme with a fine-grained decision-level fusion algorithm that effectively combines handcrafted and deep features. We train generic features on public datasets, enabling us to handle unseen scenarios. We use RootSIFT as prior knowledge to guide the extraction of deep features, significantly reducing computational overhead. We also carefully design preprocessing steps by incorporating drone attitude information. Eventually, as evidenced by our experimental results, the proposed scheme achieves an overall 2.5-6x more correct matches with improved robustness when compared to existing methods.

Keywords: Feature Fusion · Image Matching · Neural Network.

1 Introduction

In recent years, the advancements in *Unmanned Aerial Vehicle* (UAV) technology have led to its gradual integration into various national economic industries around the world, such as security, agriculture and logistics. In addition, advances in both vision sensors and image processing technology have established machine vision as the fundamental method for UAVs to perceive their surroundings. Particularly, image matching is considered essential in the field of machine vision for object detection [1,2] and image stitching [3,4].

Handcrafted feature based image matching algorithms are mainly based on expert knowledge and provide strong interpretability. Nevertheless, they lack the ability to extract high-level features that are especially important in tasks such as aerial images, which are often affected by illumination, attitude, rotation and

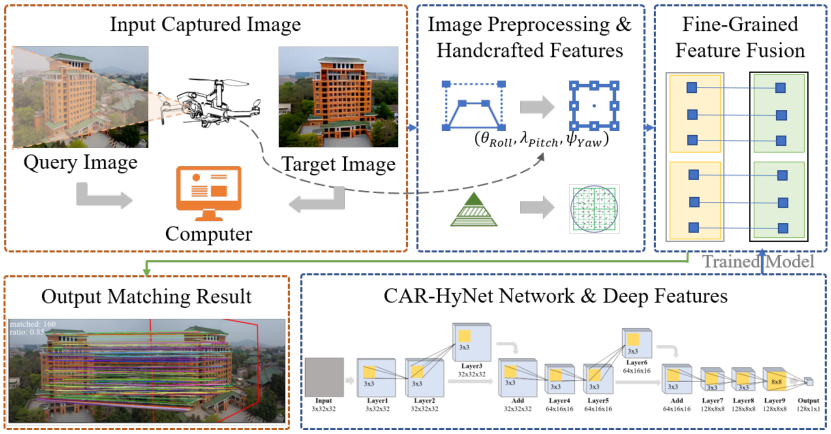


Fig. 1. Overall system architecture. By inputting the *Target Image* and the *Query Image* captured by the drone into the computer, the process begins with *Image Preprocessing and Handcrafted feature extraction* (Section §3). Subsequently, CAR-HyNet is employed to extract deep features (Section §4). Finally, a fine-grained feature fusion algorithm merges and matches the two types of features, yielding the final result (Section §5).

other factors. On the other hand, the recent developments of image matching algorithms based on deep learning have dramatically improved performance and matching accuracy due to their strong feature extraction capability for complex features such as morphology and texture [5]. However, it requires a large amount of specific samples and computing resources for training and inference, which greatly limits its application. It is therefore popular in many fields to combine traditional machine vision techniques with deep learning. This is especially useful where fast implementation is required to provide more reliable feature point matching pairs in image matching [6].

We summarize our major contributions of this paper as below.

(1) We *design* a preprocessing method using drone attitude information. For 2D objects, we take advantage of the drone attitude information to perform an inverse perspective transformation. This improves feature detection while avoiding high latency of simulated perspective transformation.

(2) We *propose* a novel deep learning architecture, named the *Coordinate Attention Residual HyNet* (CAR-HyNet), based on the HyNet [7] architecture. By incorporating coordinate attention, sandglass structure, and residual structure, we effectively enhance the performance of the model.

(3) We *introduce* a novel approach for fusing handcrafted and deep features at decision level. We use RootSIFT [8] to generate handcrafted features and use them as prior knowledge of CAR-HyNet to extract image patches and generate deep features. Finally, we use a fine-grained fusion method to efficiently fuse these two features.

The architecture of the system is illustrated in Fig. 1. Typically, the drone hovers in the air, takes a picture of the ground as *Query Image* and stores current attitude information in *Exchangeable Image File Format* (Exif). The computer on the ground takes the *Target Image* and *Query Image* as inputs to perform the image matching task using the method proposed in this paper.

More precisely, this paper is organized as follows. We provide a detailed analysis of the related work in Section 2. In Section 3, we describe the image preprocessing. Next, in Section 4, we go through details of the proposed CAR-HyNet network, followed by the feature fusion method in Section 5. We describe experimental setup and comparative studies in Section 6. We further share our thoughts regarding the limitations of current work and areas for future exploration in Section 7. Finally, we conclude this paper in Section 8.

2 Related Work

2.1 Image Matching

Image matching is the process of comparing two images with or without rotation and scaling by a specific algorithm to find the regions with the greatest similarity, in order to determine the geometric relationship between two images [9]. Region-based image matching algorithms perform image matching by comparing differences directly at pixel level or converting images to other information domains for similarity matching. Feature-based image matching algorithms have been widely studied due to their ability to reduce the impact of noise or deformation by selecting invariant features or significant regions for matching.

2.2 Using Handcrafted Features

Handcrafted feature-based image matching is generally divided into three steps: feature point extraction, feature descriptor generation, and feature matching and filtering. Lowe et al. [10] propose the *Scale Invariant Feature Transform* (SIFT) algorithm in 1999. The algorithm uses the Difference-of-Gaussian (DoG) method to approximate LoG, which speeds up feature extraction. The algorithm has invariance to scaling, rotation and translation, as well as a certain degree of illumination and affine invariance.

Many advances have been made based on the SIFT. For instance, the SURF [11] incorporates box filtering and image integration to accelerate gradients. The FAST detector is suitable for real-time video processing, while the BRIEF [12] employs binary descriptors. The ORB [13] is invariant to rotation and scale, and the KAZE [14] preserves edge information. Although these algorithms have improved detection speed, SIFT is still widely used in practice due to its advantages in invariance and robustness to illumination and affine transformations [15].

2.3 Using Deep Features

Deep learning can extract higher level semantic features from images compared to handcrafted features and has been applied in image matching. Verdie et al. [16] introduce the *Temporally Invariant Learned Detector* (TILDE), which demonstrates robustness against changes in lighting and weather conditions. Yi et al. [17] propose an end-to-end algorithm called the *Learned Invariant Feature Transform* (LIFT). Tian et al. [18] show that L2-Net can be combined directly with SIFT by matching features at L2 distance. HardNet [19] is proposed based on L2-Net to maximize the distance between the nearest positive and negative samples. Subsequently, Luo et al. [20] introduce the geometric similarity measure and propose the *Geometry Descriptor* (GeoDesc). Tian et al. [21] propose the SOSNet by introducing second order constraints into feature descriptors and the HyNet [7] which further enhances feature representation.

2.4 Combining Handcrafted Features with Deep Features

In recent years, researchers have focused on the relationship between handcrafted and deep features [22]. Combining multiple features can often achieve superior performance over a single feature. Barroso et al. [23] propose *Key.Net*, which combines handcrafted features with CNN. Zhou et al. [24] combine CNN with color feature HSV, shape feature HOG, and local feature SIFT for image classification. Rodriguez et al. [25] propose SIFT-AID by combining SIFT and CNN to produce affine invariant descriptors. However, the proposed algorithm is time consuming due to simulated perspective transformations. Song et al. [26] propose a multi-data source deep learning object detection network (MS-YOLO) based on millimeter-wave radar and vision fusion. Nevertheless, most existing methods focus on the direct combination of handcrafted and deep features, which inevitably leads to inferior results after feature fusion.

3 Attitude-Oriented Image Preprocessing

In aerial scenes, the UAV may be at a tilt angle, causing the shape of the target in the oblique image to undergo geometric changes compared to the rectified image, resulting in perspective transformation. Most feature matching algorithms are not robust to perspective transformation. A more classical and widely used approach is to simulate perspective transformation by generating multiple images for matching [27], as shown in Fig. 2. Although the number of feature matches under perspective changes can be greatly improved by matching multi-view images separately, it is inefficient and imposes high latency.

Note that two images in space can be transformed using a transformation matrix and UAV attitude information is available. Therefore, for image matching of 2D targets, we propose to correct the oblique image to a bird's eye view using the attitude based inverse perspective transformation to improve the performance of feature point extraction and matching. More importantly, this approach does not incur high latency from simulating viewpoints as it only performs transformation and matching once. However, for 3D object image matching, inverse perspective transformation is less effective and therefore not recommended.

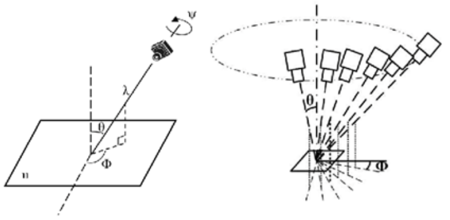


Fig. 2. Simulate perspective transformation [28].

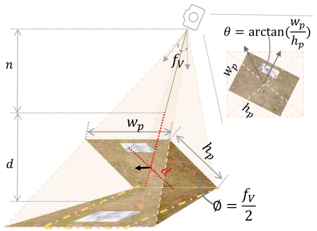


Fig. 3. Perspective transformation model.

We provide an analytical model of perspective transformation. Suppose that $F(w_p, h_p, \theta, \phi, f_V)$ represents the perspective transformation matrix, where w_p and h_p are the pixel width and pixel height of the image, θ is the rotation of the camera, ϕ is the rotation of the image plane, and f_V is the vertical perspective. Fig. 3 shows the aforementioned variables and their relationships.

With the center of the image as the center of the circle, upward as the positive direction of the y-axis, rightward as the positive direction of the x-axis, and inward as the positive direction of the z-axis, the coordinates of the four endpoints of the image are defined as: $(-\frac{w_p}{2}, \frac{h_p}{2}, 0)$, $(\frac{w_p}{2}, \frac{h_p}{2}, 0)$, $(\frac{w_p}{2}, -\frac{h_p}{2}, 0)$, $(-\frac{w_p}{2}, -\frac{h_p}{2}, 0)$. Define the perspective transformation matrix F as below,

$$F = PTR_\phi R_\theta, \tag{1}$$

where R_θ is the rotation matrix around the z-axis, R_ϕ is the rotation matrix around the x-axis, T is the translation matrix that moves the coordinate system along the z-axis, and P is the projection matrix of the vertical field of view f_V.

To calculate T, we define $d = \sqrt{w_p^2 + h_p^2}$ as the side length of the square containing any rotated portion of the image. As shown in Fig. 3 , using f_V from camera parameters, we calculate $h = \frac{d}{2\sin(\frac{f_V}{2})}$, which describes the degree of translation of the object along the negative z-axis. Thus, the matrix T is

$$T = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,-h \\ 0\,0\,0\,1 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,-\frac{d}{2\sin(\frac{f_V}{2})} \\ 0\,0\,0\,1 \end{bmatrix}. \tag{2}$$

Correspondingly, the projection matrix P is given by

$$P = \begin{bmatrix} \cot\left(\frac{f_V}{2}\right) & 0 & 0 & 0 \\ 0 & \cot\left(\frac{f_V}{2}\right) & 0 & 0 \\ 0 & 0 & -\frac{(f+n)}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}, \tag{3}$$

where $n = h - \frac{d}{2}$ and $f = h + \frac{d}{2}$. The perspective transformation matrix can be obtained by substituting Equation (3) into Equation (1).

An illustrative example of the effect after inverse perspective transformation is given in Fig. 4. However, note that this method has its limitations when the tilt angle is too large, resulting in the transformed image being too small to retain sufficient information. To alleviate this problem, we can transform it in a smaller angle to prevent excessive loss of information.

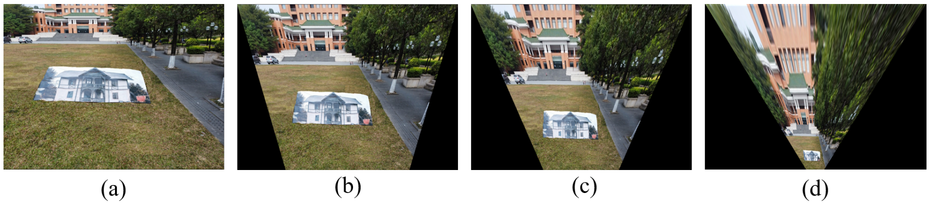


Fig. 4. Inverse perspective transformation at (a) 0°, (b) 30°, (c) 45°, and (d) 60°.

4 Design and Improvement of the CAR-HyNet Network

By combining handcrafted features with deep features, more accurate and adaptable features can be extracted and described. The convolutional network HyNet [7] evolves from L2-Net [18] and introduces a new triplet loss function from the perspective of optimizing feature descriptors, which makes the image match up to the state-of-the-art.

4.1 CAR-HyNet Network Structure

To address the challenges in aerial image processing, we propose a new improved multi-channel *Coordinate Attention Residual HyNet* (CAR-HyNet) network based on HyNet, as shown in Fig. 5. More precisely, we introduce *Coordinate Attention* (CoordAtt) [29] and design a *Coordinate Attention Sandglass Network* (CA-SandGlass), and modify HyNet to apply CA-SandGlass for aerial image processing. In addition, we take full advantage of RGB three-channel as inputs to further improve overall image matching performance.

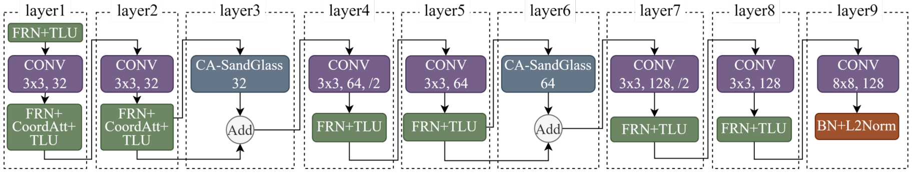


Fig. 5. Overall network structure of CAR-HyNet.

Coordinate attention sandglass network We notice that conventional convolutional operations can only capture local positional relationships, a significant drawback for processing aerial images from UAVs. To address this limitation, we propose using *Coordinate Attention* (CoordAtt) [29] by embedding position information into channel attention to capture remote dependencies for accurate feature descriptors.

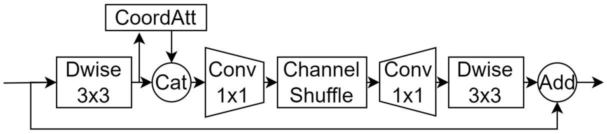


Fig. 6. Structure of CA-SandGlass.

Furthermore, since SandGlass [30] is a lightweight module. Considering that CoordAtt focuses on long-range dependencies and SandGlass focuses on feature information at different scales, combining these two techniques allows the network to generate more comprehensive and discriminative feature representations. Additionally, as the residual connection needs to be built on high-dimensional features, we further combine them to form the CA-SandGlass block. The structure is shown in Fig. 6.

Increasing nonlinearity of the model The original HyNet structure is the same as L2-Net, which consists of 6 feature extraction layers and 1 output layer. In contrast, we propose adding 2 layers of CA-SandGlass to increase non-linearity. Our experiments indicate that adding more than 2 layers of CA-SandGlass does not result in any improvement in matching accuracy. To avoid potential adverse effects such as gradient dispersion, we connect the 2 CA-SandGlass layers to the backbone using a residual connection. Overall, this design offers the best balance between performance and complexity.

RGB three-channel image input Another important improvement we introduced in CAR-HyNet is leveraging the full RGB three-channel as inputs. Compared to grayscale images, color images contain much richer information at a negligible computational cost. The absence of color information in processing can result in incorrect matching, particularly in regions with the same grayscale and shape but different colors.

4.2 CAR-HyNet Performance Evaluation

To evaluate the performance of the proposed CAR-HyNet, we compare it with several widely used models. For fairness, we use the unified Brown [31] dataset for evaluation, which includes Liberty (LIB), Notre Dame (ND), and Yosemite (YOS), and experimental results for existing models are taken from their papers. We perform standard *False Positive Rate at 95%* (FPR95) measurements across 6 training and test sets, as shown in Table 1. As we can see, CAR-HyNet outperforms the other models with a notable improvement in detection performance.

Table 1. Patch verification performance on the Brown dataset (FPR@95) [7].

Train	ND	YOS	LIB	YOS	LIB	ND	Mean
Test	LIB		ND		YOS		
SIFT [32]	29.84		22.53		27.29		26.55
TFeat [33]	7.39	10.13	3.06	3.80	8.06	7.24	6.64
L2-Net [18]	2.36	4.70	0.72	1.29	2.57	1.71	2.23
HardNet [19]	1.49	2.51	0.53	0.78	1.96	1.84	1.51
DOAP [34]	1.54	2.62	0.43	0.87	2.00	1.21	1.45
SOSNet [21]	1.08	2.12	0.35	0.67	1.03	0.95	1.03
HyNet [7]	0.89	**1.37**	0.34	0.61	0.88	0.96	0.84
CAR-HyNet	**0.77**	1.53	**0.30**	**0.57**	**0.69**	**0.64**	**0.75**

5 Decision Level Fusion for Image Matching

One challenge we face in this work is how to fuse handcrafted features with deep features appropriately. The vast majority of the existing literature focuses on feature level fusion using weighted fusion of multiple features. For example, color features, corner features, histogram features, and convolution features of the image are fused directly using different weights. However, directly weighting different features for superposition ignores the fact that different features have different degrees of sensitivity in different scenarios. This inescapably leads to poor performance as some features inadvertently suppress others, introducing a large number of incorrect matches.

To this end, we propose a new fine-grained decision level fusion method that combines handcrafted features with deep features. The method fully considers the correlation of different feature extraction methods on feature points, effectively improving the number of correct matching pairs.

5.1 Extracting Handcrafted Features

To prepare for decision level fusion, we first extract handcrafted features using the RootSIFT algorithm [8]. RootSIFT is an extended mapping algorithm to the *Scale-Invariant Feature Transform* (SIFT) algorithm, but achieves a higher number of correct matches of feature descriptors. In our tests with DEGENSAC and NNDR=0.85, the number of correct matches of feature descriptors is improved by approximately 19.4% after RootSIFT mapping.

5.2 Extracting Deep Features

To achieve fine-grained control over the correspondence of feature points during the fusion process and reduce the computational workload of deep learning, we use handcrafted features as prior knowledge for deep feature extraction. We first reconstruct the scale pyramid of the color image based on the image processed in the previous section and feature points extracted by RootSIFT. We then generate patches by intercepting the image at a size of 64x64 in the corresponding scale space and then scaling them down to 32x32.

Since data augmentation techniques can introduce noisy data and CNNs are not invariant to rotation [35], we further rotate patches to primary orientation. Finally, we feed patches into CAR-HyNet to eventually generate 128-dimensional deep features. This method fully leverages the feature points extracted by RootSIFT, mitigates the lack of rotational invariance in CNNs, and successfully generates deep features for fusion.

5.3 Fine-grained Decision Level Feature Fusion

Next, we present the design of the proposed algorithm for decision level fusion. As a high level fusion, decision level fusion offers global optimal decision with high accuracy and flexibility [36].

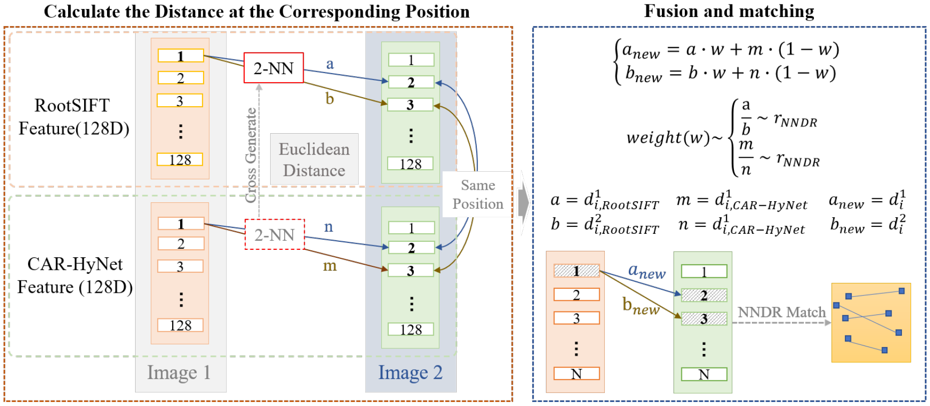


Fig. 7. Fine-grained Decision level fusion using Euclidean distance. *NNDR* represents *Nearest Neighbor Distance Ratio*, *2-NN* represents the nearest neighbor and second-nearest neighbor based on Euclidean distance.

An illustrative example of the algorithm flow is shown in Fig. 7. For a pair of input images to be matched, denoted as *Image1* and *Image2*, we first extract their RootSIFT feature descriptors $D^1_{RootSIFT}$, $D^2_{RootSIFT}$, and then extract the CAR-HyNet feature descriptors $D^1_{CAR-HyNet}$, $D^2_{CAR-HyNet}$ with rotation and scale invariance, respectively. We calculate the Euclidean distance between the feature descriptors of each feature point in the two images under RootSIFT and CAR-HyNet respectively, and calculate the two nearest points, $d^m_{i,RootSIFT}$ and $d^m_{i,CAR-HyNet}$, for each feature point, where $m = 1, 2$ indicates the first and second nearest neighbors. For the two nearest neighboring points of each feature point, we find the distances of these two points at the corresponding positions of the CAR-HyNet feature points by traversing the RootSIFT feature points in turn. We then use the NNDR method to determine the success of the matching from the two feature extraction algorithms.

To retain more implicit matching point pairs, we use the NNDR threshold α with a lenient strategy. The Euclidean distance d of the feature descriptor is calculated in Equation (4), where *dim* represents the dimension of the feature descriptor, $D^i_{type,k}$ represents the k-th dimensional value of the feature descriptor of the i-th feature point of $type = RootSIFT, CAR - HyNet$, m represents the m-th closest distance. When $i = j$, it is further simplified as $d^m_{i,type}$.

$$d^m_{ij,type} = \sqrt{\sum_{k=1}^{dim} (D^i_{type,k} - D^j_{type,k})} \tag{4}$$

Furthermore, using the weight $w \in [0, 1]$ to fuse the Euclidean distances of two points and generate new distances as the nearest neighbor $d^1_{i\_new}$ and second nearest neighbor distance $d^2_{i\_new}$ of the feature point. In our experiments, we set w to 0.75. The fusion equation is calculated as Equation (5).

$$\begin{cases} d^1_{i\_new} = d^1_{i,RootSIFT} \cdot w + d^1_{i,CAR-HyNet} \cdot (1 - w) \\ d^2_{i\_new} = d^2_{i,RootSIFT} \cdot w + d^2_{i,CAR-HyNet} \cdot (1 - w) \end{cases} \tag{5}$$

To further improve the reliability of the results, we traverse the feature points of CAR-HyNet again to repeat the above steps for cross-generation and eventually merge them into a new set of feature points. To filter out any potential incorrect matches as much as possible, we take another NNDR screening with a stricter threshold α. For duplicate matching feature points, we empirically retain the one with the smaller nearest neighbor distance. Finally, we use the DEGENSAC algorithm [37] to refine the screening to obtain the final matched feature points.

5.4 Fine-grained Decision Level Feature Fusion Evaluation

We present our evaluation of the proposed decision level fusion method. We test the performance of different algorithms under direct weighted fusion and our method. For a fair comparison, we select RootSIFT+HyNet, RootSIFT+HardNet and RootSIFT+CAR-HyNet for our experiments at different heights and perspectives. The results are shown in Table 2.

Table 2. Correct matching numbers of different feature fusion methods with different heights and perspectives (NNDR threshold=0.85, fusion weight=0.75), where DWF refers to traditional *Direct Weighted Fusion* and DLF refers to our proposed *Decision Level Fusion*. Note that due to limited space, we only show results of 9 typical images of different heights and perspectives respectively.

Algorithm	**Method**	**Heights**						**Perspectives**					
		d1	d3	d5	d6	d7	d9	a1	a3	a5	a7	a9	a15
HardNet	DWF	1013	196	87	48	35	18	295	165	189	160	109	20
	DLF	**1086**	**230**	**94**	**56**	**44**	**20**	**351**	**216**	**236**	**206**	**175**	**85**
HyNet	DWF	1003	192	83	47	35	18	296	170	188	159	108	23
	DLF	**1073**	**227**	**93**	**57**	**41**	**22**	**358**	**221**	**238**	**208**	**181**	**84**
CAR-HyNet	DWF	1027	216	91	57	35	18	400	275	267	223	208	32
	DLF	**1099**	**244**	**109**	**64**	**46**	**22**	**483**	**341**	**308**	**269**	**292**	**89**

As can be seen from Table 2, the improved feature fusion method shows excellent performance with all three algorithms. Moreover, our proposed RootSIFT+CAR-HyNet method outperforms other methods and yields a higher number of correct matches.

6 Experiments

In this section, we first describe our experimental setup, datasets, and evaluation criteria. We conduct matching experiments on real world aerial images, and compare the proposed method with existing methods in detail. [3]

Since CNNs are not rotation invariant, which leads to detect more matching feature points only when two images are similar. To make the results more comparable, patches in the following experiments are generated after the operations described in Section 5.2. The utilization of different combinations of algorithms is presented in Table 3.

Table 3. Combination of different algorithms

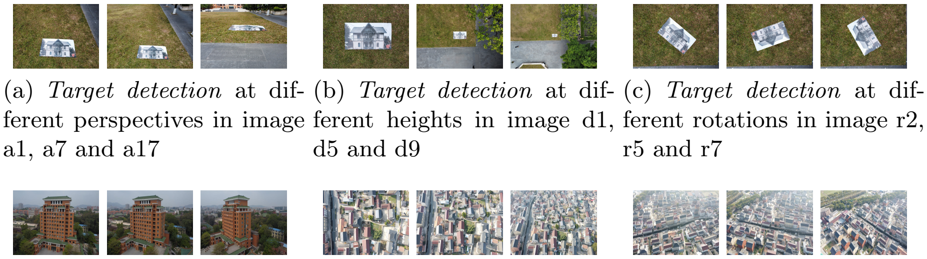
Algorithm	Feature points	Feature descriptor	Image pyramid
SIFT	SIFT	SIFT	Gray
KAZE	KAZE	KAZE	Gray
KeyNet	KeyNet	KeyNet	Gray
HardNet	RootSIFT	HardNet	Gray
HyNet	RootSIFT	HyNet	Gray
CAR-HyNet	RootSIFT	CAR-HyNet	Color

6.1 Lab Setup and Datasets

For the experimental setup, we use a Supermicro server in our lab with an Intel(R) Xeon(R) Gold 6230 CPU, NVIDIA RTX 3090 GPU, and 128GB of memory. The software environment consists of Ubuntu 20.04, Python 3.7 and Pytorch framework. We use a DJI Mini2 to capture aerial images over an open area and the target object in the target detection task is a cloth of size 100cm×177cm. The original images captured by the DJI Mini2 are 4000×3000 in size and compressed to 800×600. Due to space limitations, we only display a selection of typical images, as shown in Fig. 8.

We choose the widely used Brown [31] and HPatches [38] datasets for the joint training of CAR-HyNet. The Brown dataset includes three sub-datasets: Liberty (LIB), Notre Dame (ND), and Yosemite (YOS). We employ standard *False Positive Rate at 95%* (FPR95) as the evaluation metric. In addition, we employ *Nearest Neighbor Distance Ratio* (NNDR) as the feature matching method. For practical drone applications, we set the NNDR threshold at 0.85. Meanwhile, to prevent too many feature points, we employ a feature intensity filter to retain the first 4000 feature points.

[3] Our code is publicly available on Github: https://github.com/songxf1024/DeFusion



(a) *Target detection* at different perspectives in image a1, a7 and a17

(b) *Target detection* at different heights in image d1, d5 and d9

(c) *Target detection* at different rotations in image r2, r5 and r7

(d) *Building* at different perspectives in image h2, h5 and h9

(e) *Aerial* at different perspectives in image ha1, ha5 and ha9

(f) *Aerial* at different rotations in image hr1, hr2 and hr6

Fig. 8. Samples from aerial image datasets, captured by the DJI Mini2.

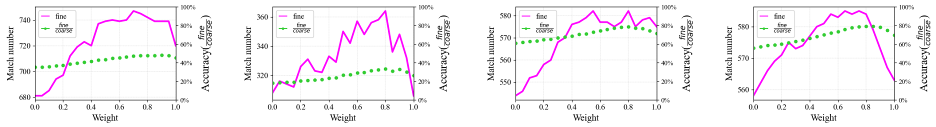
6.2 Impact of NNDR Thresholds and Fusion Weights

In our experiments, we observe that different NNDR thresholds and feature fusion weights have a significant impact on matching performance. To understand the potential correlations, we conduct a series of experiments using a typical set of images from the dataset. Fig. 9 represents the trend of the number of matches and accuracy of CAR-HyNet with different weights $(\frac{RootSIFT}{CAR-HyNet} = \frac{w}{1-w})$, where *fine* represents the number of correct matches, and *coarse* represents the maximum number of matches including incorrect matches. In addition, Fig. 10 represents the influence of different algorithms by NNDR thresholds in perspective and height scenarios. To ensure practical applicability for UAVs, we empirically set the NNDR threshold at 0.85 and the feature fusion weight at 0.75 after extensive experimental comparisons, as it provides an appropriate balance between fusing handcrafted and deep features, enabling us to maximize the number of correct matches while achieving high matching accuracy as much as possible.

Another observation we find is that, as shown in Fig. 10(b), the matching accuracy of the proposed method is highest when the NNDR threshold is set to a small value, and then decreases as the NNDR threshold increases. This can be explained by the fact that with an increased threshold, the proposed method is able to include more potential matching points. Therefore, the matching accuracy begins to decrease even though the number of correct matches remains maximal.

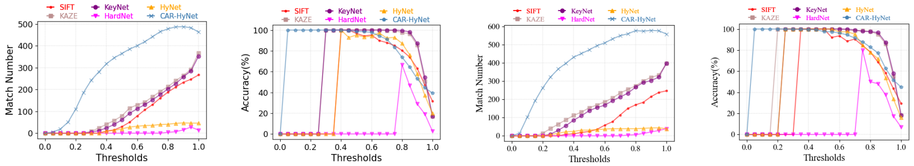
6.3 Time Consumption Evaluation

In addition, we evaluate the time consumption of different algorithms with a maximum limit of 4000 feature points, as shown in Table 4. We use $a0$ and $a4$ from our dataset for the evaluation. Experiments show that the proposed method offers the best performance in time and number of correct matches. As shown in Table 4, it takes 5500ms for ASIFT to detect 260 correct matches, while the proposed method takes 1096ms to detect 300 correct matches. Also note that other methods in comparison take about 1100ms, but with less than 120 correct matches. By analyzing our method, we also note an interesting observation,



(a) Image *h1* (b) Image *h4* (c) Image *r1* (d) Image *r8*

Fig. 9. Number of matches and matching accuracy at different feature fusion weights ($\frac{RootSIFT}{CAR-HyNet} = \frac{w}{1-w}$), where *fine* is the number of correct matches and *coarse* is the maximum number of matches including incorrect ones.



(a) Match Number vs NNDR Thresholds in image *a1* (b) Accuracy vs NNDR Threshold in image *a1* (c) Match Number vs NNDR Threshold in image *r1* (d) Accuracy vs NNDR Threshold in image *r1*

Fig. 10. Performance on sample image *a1* and *r1* at different NNDR thresholds.

where the stages of *generate patches* and *feature matching* accounted for most of the time, as shown in Table 5.

Table 4. Time-consumption and number of correct matches of different algorithms with a maximum limit of 4000 feature points.

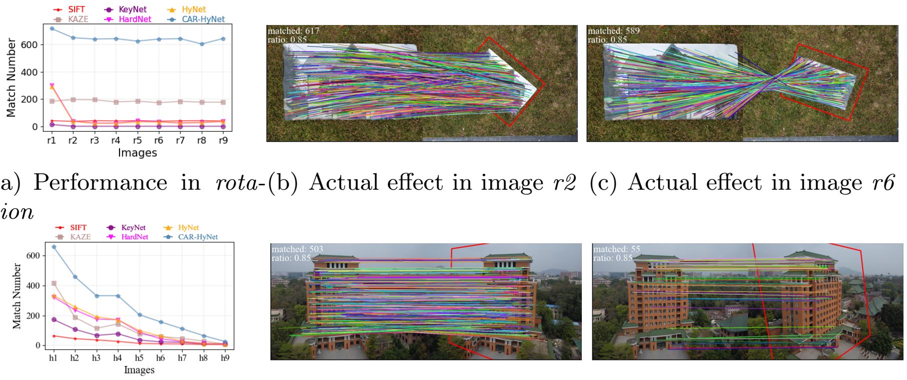
Algorithm	Elapsed Time(ms)	Correct Matches
SIFT	1166.7	70
ASIFT	5006.2	274
KAZE	898.0	79
KeyNet	493.6	5
HardNet	1575.8	117
HyNet	1605.0	122
CAR-HyNet	1096.1	**300**

Table 5. Time elapsed at each stage of CAR-HyNet with a maximum limit of 4000 feature points.

Stage	NMS	Preprocess	IPM	Filter	RootSIFT	Pyramid	Patches	CAR-HyNet	Match
Elapsed Time(ms)	0.001	77.638	7.318	0.005	147.221	83.511	340.984	59.418	380.013

6.4 Overall Performance Evaluation

Fig. 11 provides an overall comparative perspective of the proposed method along with other algorithms and the actual matching effect of our method. As shown on the left side of Fig. 11, the proposed method can provide significantly better overall matching performance than other algorithms with robustness in all scenarios. Our method even outperforms the well-performing RootSIFT+HardNet combination and KAZE by 2.5x-6x. The right side of Fig. 11 provides a real world perspective of the actual matching results for the testing images. Due to space limitations, we only give a part of the results here, but experimental results on other data show the same trend.



(a) Performance in *rotation* (b) Actual effect in image *r2* (c) Actual effect in image *r6*

(d) Performance in *building match* (e) Actual effect in image *h2* (f) Actual effect in image *h8*

Fig. 11. Overall matching performance comparison and actual matching results in the real world.

7 Discussion and Future Work

The proposed method shows excellent performance in image matching. However, we also notice that there are several areas for potential improvement. We share our thoughts below for discussion.

Rotation invariance. We rotate patches according to the primary orientation of feature points computed by SIFT, which gives our method rotation invariant that can be compared with SIFT. However, the accuracy of the primary orientation in SIFT is inherently inaccurate, implying errors in the matching of certain feature points. Since rotation is very common in the real world, we plan to investigate methods to improve rotation invariance and reduce errors caused by primary orientation.

Real-time. Currently, the proposed method takes an average of 1 second to complete an image matching, which is acceptable for offline applications but too slow to meet real-time operational demands. In future work, we plan to investigate the use of faster feature descriptors, dimension reduction techniques, employing lightweight models, and other strategies to minimize matching latency.

Operating platform. Our experiments are conducted on a computer in the lab. In future work, we plan to explore transferring the system to a drone platform, thereby achieving an end-to-end and real-time image matching system.

8 Conclusion

In this paper, we propose a novel image matching scheme. The proposed image preprocessing improves detection performance by using drone attitude information. We design the CAR-HyNet network that is more suitable for feature representation and generate deep features using SIFT as prior knowledge. Finally, we propose a fine-grained decision level fusion algorithm to effectively combine handcrafted features and deep features. Experimental results show that our proposed RootSIFT+CAR-HyNet combination provides the best overall matching performance. The effectiveness of our method is further demonstrated through experiments, where it takes an average of only 1 second for 4000 feature points and achieves 2.5-6x more matches than existing methods. In addition, it is trivial to generalize the proposed method to other datasets. In summary, we believe that the proposed image matching scheme shows great potential. As we shared in the previous section, there are still several aspects for further exploration, and we hope this paper will pave the way for more active exploration in the field of image matching.

9 Acknowledgment

This research was supported in part by the South China University of Technology Research Start-up Fund No. X2WD/K3200890, as well as partly by the Guangzhou Huangpu District International Research Collaboration Fund No. 2022GH13. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

1. Sharma, M., Singh, H., Singh, S., Gupta, A., Goyal, S., Kakkar, R.: A novel approach of object detection using point feature matching technique for colored images. In: Singh, P.K., Kar, A.K., Singh, Y., Kolekar, M.H., Tanwar, S. (eds.) Proceedings of ICRIC 2019. pp. 561–576. Springer International Publishing, Cham (2020)
2. Rashid, M., Khan, M.A., Sharif, M., Raza, M., Sarfraz, M.M., Afza, F.: Object detection and classification: a joint selection and fusion strategy of deep convolutional neural network and sift point features. Multimedia Tools and Applications **78**(12), 15751–15777 (2019)
3. Ma, J., Zhou, H., Zhao, J., Gao, Y., Jiang, J., Tian, J.: Robust feature matching for remote sensing image registration via locally linear transforming. IEEE Transactions on Geoscience and Remote Sensing **53**(12), 6469–6481 (2015)
4. Ravi, C., Gowda, R.M.: Development of image stitching using feature detection and feature matching techniques. In: 2020 IEEE international conference for innovation in technology (INOCON). pp. 1–7. IEEE (2020)
5. Bochkovskiy, A., Wang, C., Liao, H.M.: Yolov4: Optimal speed and accuracy of object detection. CoRR **abs/2004.10934** (2020)

6. O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J.: Deep learning vs. traditional computer vision. In: Science and information conference. pp. 128–144. Springer (2019)
7. Tian, Y., Barroso Laguna, A., Ng, T., Balntas, V., Mikolajczyk, K.: Hynet: Learning local descriptor with hybrid similarity measure and triplet loss. Advances in Neural Information Processing Systems **33**, 7401–7412 (2020)
8. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 2911–2918. IEEE (2012)
9. Pérez-Lorenzo, J., Vázquez-Martín, R., Marfil, R., Bandera, A., Sandoval, F.: Image Matching Based on Curvilinear Regions. na (2007)
10. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 1150–1157. Ieee (1999)
11. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: European conference on computer vision. pp. 404–417. Springer (2006)
12. Calonder, M., Lepetit, V., Strecha, C., Brief, F.: Binary robust independent elementary features. In: Proceedings of the European Conference on Computer Vision. pp. 778–792
13. Rublee, E., Rabaud, V., Konolige, K., Orb, G.: An efficient alternative to sift or surf. In: Proceedings of International Conference on Computer Vision. pp. 2564–2571
14. Alcantarilla, P.F., Bartoli, A., Davison, A.J.: Kaze features. In: European conference on computer vision. pp. 214–227. Springer (2012)
15. Efe, U., Ince, K.G., Alatan, A.A.: Effect of parameter optimization on classical and learning-based image matching methods. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2506–2513 (2021)
16. Verdie, Y., Yi, K., Fua, P., Lepetit, V.: Tilde: A temporally invariant learned detector. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5279–5288 (2015)
17. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European conference on computer vision. pp. 467–483. Springer (2016)
18. 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. pp. 661–669 (2017)
19. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. Advances in neural information processing systems **30** (2017)
20. Luo, Z., Shen, T., Zhou, L., Zhu, S., Zhang, R., Yao, Y., Fang, T., Quan, L.: Geodesc: Learning local descriptors by integrating geometry constraints. In: Proceedings of the European conference on computer vision (ECCV). pp. 168–183 (2018)
21. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11016–11025 (2019)
22. Zheng, L., Yang, Y., Tian, Q.: Sift meets cnn: A decade survey of instance retrieval. IEEE transactions on pattern analysis and machine intelligence **40**(5), 1224–1244 (2017)

23. Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key. net: Keypoint detection by handcrafted and learned cnn filters. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5836–5844 (2019)
24. Tianyu, Z., Zhenjiang, M., Jianhu, Z.: Combining cnn with hand-crafted features for image classification. In: 2018 14th ieee international conference on signal processing (icsp). pp. 554–557. IEEE (2018)
25. Rodríguez, M., Facciolo, G., von Gioi, R.G., Musé, P., Morel, J.M., Delon, J.: Sift-aid: boosting sift with an affine invariant descriptor based on convolutional neural networks. In: 2019 IEEE international conference on image processing (ICIP). pp. 4225–4229. IEEE (2019)
26. Song, Y., Xie, Z., Wang, X., Zou, Y.: Ms-yolo: Object detection based on yolov5 optimized fusion millimeter-wave radar and machine vision. IEEE Sensors Journal **22**(15), 15435–15447 (2022)
27. Yu, G., Morel, J.M.: ASIFT: An Algorithm for Fully Affine Invariant Comparison. Image Processing On Line **1**, 11–38 (2011)
28. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM J. Img. Sci. **2**(2), 438–469 (apr 2009)
29. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13713–13722 (2021)
30. Zhou, D., Hou, Q., Chen, Y., Feng, J., Yan, S.: Rethinking bottleneck structure for efficient mobile network design. In: European Conference on Computer Vision. pp. 680–697. Springer (2020)
31. Winder, S.A., Brown, M.: Learning local image descriptors. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
32. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
33. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Bmvc. vol. 1, p. 3 (2016)
34. He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 596–605 (2018)
35. Kim, J., Jung, W., Kim, H., Lee, J.: Cycnn: A rotation invariant cnn using polar mapping and cylindrical convolution layers. arXiv preprint arXiv:2007.10588 (2020)
36. Gunatilaka, A.H., Baertlein, B.A.: Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. IEEE transactions on pattern analysis and machine intelligence **23**(6), 577–589 (2001)
37. Chum, O., Werner, T., Matas, J.: Two-view geometry estimation unaffected by a dominant plane. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 772–779. IEEE (2005)
38. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR (2017)