



On-Board Geometric Rectification for Micro-Satellite Based on Lightweight Feature Database

Linhui Wang^{1,2,3}, Yuming Xiang^{1,2,3,*} , Zhenzhou Wang^{1,2}, Hongjian You^{1,2,3} and Yuxin Hu^{1,2}

¹ Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; wanglinhui21@mails.ucas.ac.cn (L.W.)

² Key Laboratory of Technology in Geo-Spatial Information Processing and Application Systems, Chinese Academy of Sciences, Beijing 100190, China

³ School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 101408, China

* Correspondence: xiangym@aircas.ac.cn; Tel.: +86-152-1056-0903

Abstract: On-board processing is increasingly prevalent due to its efficient utilization of satellite resources. Among these resources, geometric rectification can significantly enhance positioning accuracy for subsequent tasks, such as object detection. This approach mitigates the heavy burden on downlink bandwidth and minimizes time delays by transmitting targeted patches rather than raw data. However, existing rectification methods are often unsuitable due to the limitations and conditions imposed on satellites. Factors like hardware quality, heat dissipation, storage space, and geographic positioning are frequently constrained and prone to inaccuracies. This paper proposes a novel on-board rectification method. The method introduces a two-step matching framework to address substantial positioning errors and incorporates a feature-compression strategy to reduce the storage space of reference patches. Quantitative and practical experiments demonstrate the method's efficacy in terms of storage space, time efficiency, and geometric rectification accuracy.

Keywords: on-board processing; micro-satellite; rectification; lightweight database



Citation: Wang, L.; Xiang, Y.; Wang, Z.; You, H.; Hu, Y. On-Board Geometric Rectification for Micro-Satellite Based on Lightweight Feature Database. *Remote Sens.* **2023**, *15*, 5333. <https://doi.org/10.3390/rs15225333>

Academic Editors: Akira Iwasaki, Tianwen Zhang, Tianjiao Zeng, Jun Shi, Xiaoling Zhang, Shuaicheng Liu, Shunjun Wei and Qingze Zou

Received: 20 September 2023
Revised: 8 November 2023
Accepted: 11 November 2023
Published: 12 November 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

Micro-satellites are rapidly advancing and finding applications across various domains, including wildfire detection, technology demonstrations, scientific research, communication, and Earth observation [1–4]. These micro-satellites have gained popularity over traditional devices due to their reduced costs and development cycles, smaller sizes, lower power consumption, and adaptability for networking and constellation [1]. However, their extensive use has resulted in a substantial surge in the volume of raw image data [5]. Conventional data processing involves transmitting images to ground stations before further utilization, leading to significant delays caused by limited downlinking bandwidth. This places enormous strain on ground data processing systems [6]. The delay between data generation and information extraction not only impacts time efficiency and ground station maintenance but also diminishes the value of the data [7]. This delay can result in missed opportunities, such as overlooking a low-cloud-coverage observation window during ground analysis [1]. Consequently, satellites often generate a considerable amount of redundant data, squandering resources and hardware while failing to address emergencies [8]. This scenario has prompted researchers to consider shifting some of the processing flow onto satellites themselves.

On-board processing is gaining significant traction due to its ability to reduce the volume of data transmitted between satellites and ground stations, allowing for nearly real-time responses [8]. Newly launched micro-satellites now come equipped with on-board intelligent processing modules, enabling the automatic generation of standardized image products and real-time object detection [7]. Instead of transmitting raw data, these satellites

send targeted patches and their positional information to ground receivers for further analysis. This reduction in on-board processed data for downlink transmission significantly alleviates the strain on communication bandwidth [7]. Ground stations no longer have to contend with waiting for transmissions or handling vast amounts of raw data, thereby bypassing time delays, system intricacies, and maintenance costs [8]. On-board processing empowers satellites to swiftly respond to emergencies and seamlessly integrate information from multiple sources for continuous wide-area observation [9].

However, on-board processing encounters several new challenges. Reports indicate that micro-satellites exhibit lower geometric accuracy compared to traditional satellites [10,11], potentially limiting the reliability and effectiveness of subsequent applications. Achieving precise geometric accuracy is crucial before proceeding to object detection. This necessitates accurate calibration or geometric rectification. Calibration often demands prolonged time for repeated observations in calibration fields to update positioning parameters, proving time-consuming and labor-intensive for micro-satellites. Alternatively, on-board geometric rectification presents a rapid and cost-effective solution. However, the constraints of weight, volume, and power consumption in micro-satellites impose limitations on computational and storage resources [1,12]. Furthermore, the thermal conditions in space preclude the continuous operation of processing systems. Algorithms requiring extensive storage make traditional ground control point (GCP) databases unfeasible, while those with high computational complexity may not deliver practical real-time performance [6,7,9,12]. In essence, on-board rectification necessitates a lightweight database and improved computational efficiency.

On-board processing algorithms primarily focus on two aspects: simplifying models and reducing algorithmic complexity while maintaining accuracy. This is exemplified by the utilization of lightweight neural networks. Techniques such as Principal Component Analysis (PCA) [13] and Linear Discriminant Analysis (LDA) [14] aim to represent keypoint descriptors with fewer dimensions, reducing both storage consumption and computational complexity. Another method, quantization, transforms 32-bit floating-point data into 8-bit integers for the same purpose [15–17]. Complex computations can be replaced by less intensive approximations that maintain similar precision [18,19]. The trend of incorporating deep learning into on-board processing is growing, introducing lightweight network models [7,9,20]. These models enhance efficiency while preserving accuracy.

The other emphasis lies in shifting processing procedures onto specialized hardware resources. Embedded on-board processing systems generally fall into three categories: Digital Signal Processors (DSPs), Field-Programmable Gate Arrays (FPGA), and Graphics Processing Units (GPUs). Traditional on-board processing methods primarily revolve around FPGA utilization, involving the adaptation of algorithms to suit hardware characteristics [8,12,21] and designing specific FPGA implementations tailored to particular algorithms [22–25]. However, DSP and FPGA platforms have their drawbacks, including obsolescence, rigidity, and high costs [5]. These devices often lack sufficient computing capacity and struggle to handle diverse tasks [9].

Traditional embedded on-board processing systems such as DSPs and FPGAs are optimized for executing specific commands to enhance efficiency. However, these typical on-board processing platforms exhibit limitations such as inflexibility and high implementation costs with the exponential growth in data volume [5]. With recent advancements in software and hardware, there is a growing preference for employing embedded ARM and GPU architectures in on-board processing systems due to their enhanced portability and developmental flexibility [20]. The ability of GPUs to process data in parallel has proven successful in various on-board processing tasks, including sensor correction [5], ship detection [7], and image registration [20,26]. Notably, the NVIDIA Jetson Series exhibits characteristics such as low power consumption, high performance, and compact integration, rendering it suitable for real-time on-board processing systems. Given the constraints and costs associated with DSP and FPGA platforms, it is advisable to consider embedded GPUs for on-board processing systems. Products including TX1, TX2, and Nano are designed for

edge computing, offering high performance with low power consumption, a compact size, and a lower cost, making them suitable candidates for on-board processing systems.

In response to these developments, we propose an on-board rectification method that leverages a lightweight feature database to achieve precise and rapid geometric rectification based on embedded GPU technology. Initially, we construct a lightweight reference database using edge features and compression techniques. Subsequently, fast template matching is employed to mitigate significant initial geometric offsets. Finally, a refined local matching process is implemented to enhance the accuracy of the sensed image relocation. Both quantitative and practical on-board experiments confirm the viability of the proposed method. The major contributions of this paper are listed as follows:

1. A robust and easy-to-implement feature establishment method for a lightweight reference database, which focuses on distinctive runways and coastlines;
2. A feature-compressing strategy based on Run-Length Encoding is applied to the aforementioned reference database, significantly reducing the storage space while maintaining sufficient structural features;
3. Adaptation for on-board implementation, including a two-step matching framework that overcomes large and unknown positioning errors of micro-satellites, along with a GPU migration to fulfill the computation demand of on-board processing.

2. Method

The overall flow is depicted in Figure 1. Initially, we elaborate on constructing a lightweight feature database, which is created on the ground and then transferred and stored on the micro-satellites. During on-board processing, the sensed images are geocoded using initially inaccurate positioning parameters. Subsequently, the corresponding feature is searched for and decompressed from the database. Extracting the feature response of the sensed image, a coarse matching process is conducted to mitigate large initial offsets. Finally, a refined matching process is executed to precisely identify correspondences and correct the sensed images.

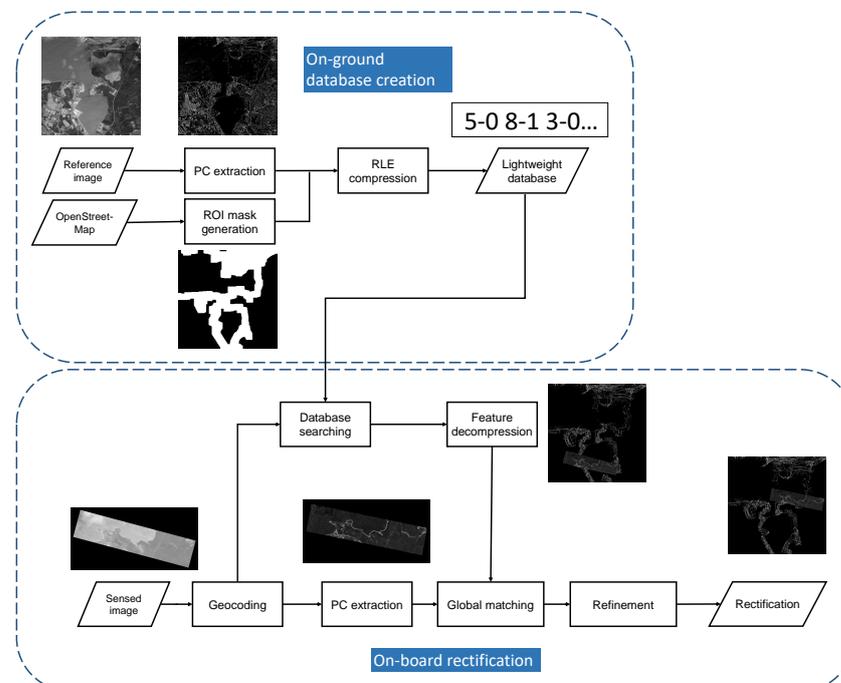


Figure 1. The entire process flow involves constructing the lightweight feature database on the ground and executing rectification on board.

2.1. Lightweight Feature Database Construction

Conventional geometric rectification methods relying on reference images demand substantial storage space. For instance, a 16-bit reference image, fixed at 5000×5000 resolution with a 1 m scale, consumes over 20 MB of storage, rendering it impractical for on-board devices. We aim to establish a universal lightweight reference database for on-board rectification, independent of micro-satellite payloads or imaging modes. We consider micro-optical satellites like the Jilin-1 series; Qilu-3; and micro-SAR satellites including Qilu-1, Hisea-1, and the Nuwa constellation. This database must be intrinsic and invariant across various imaging mechanisms and conditions.

To begin, we select 1-meter-resolution Google Earth images as reference images and employ the phase congruency method [27] to detect robust binary edge features. Subsequently, we generate a Region of Interest (ROI) mask using OpenStreetMap, preserving vital information on runways and coastlines for template matching. Additionally, the ROI mask helps eliminate redundant areas and reduces storage requirements using the Run-Length Encoding (RLE) compression technique [28]. Finally, this lightweight database is stored in satellite hardware before launching.

2.1.1. Feature Extraction

Image matching technology can be broadly classified into two primary frameworks: template matching and feature matching. Template matching identifies correspondences using phase correlation or cross-correlation, while feature matching seeks correspondences based on differences measured by feature descriptors from two sets of keypoints. Since our focus is on constructing a universal lightweight reference database, we must account for significant differences in radiometric and geometric characteristics across diverse image modalities. Real-world optical and SAR images, illustrated in Figure 2, vary considerably. For instance, what appears as flatland in optical images may show fluctuations in SAR images. These differences, due to imaging mechanisms, can lead to low keypoint repeatability in feature matching methods [29]. In such cases, the feature database may store numerous redundant keypoints to facilitate reliable feature matching, although only a fraction of them are eventually matched. This demands delicately designed feature descriptors, resulting in impractical storage requirements. Hence, we opt for template matching as the basis for our rectification process.

When performing image matching using powerful on-ground servers, it is logical to extract features comprehensively for better accuracy. However, on-board devices face constraints in storage space and computing power. Therefore, we must meticulously design the reference database to cover distinct regions and extract consistent features within limited memory. First, the features should be intrinsic and easily detectable regardless of imaging conditions such as angle, timing, weather, noise, or minor ground changes. Second, the features must exhibit invariance across diverse image modalities. Lastly, the features should possess enough uniqueness to preserve only a select few without the risk of mismatches.

Most micro-satellites are primarily dedicated to observing activities in major cities worldwide. Hence, for a preliminary on-board experiment, we select airports and harbors as the primary targets. Both airports, with their artificial runways, and harbors, characterized by coastlines, exhibit strong regularity; unique characteristics; and, crucially, consistency and robustness across different image modalities and conditions. Artificial runways at airports display a distinct, regular pattern, while harbors feature distinctive coastlines. Given the substantial radiometric and geometric differences among multi-modal images, edges serve as simple and robust features invariant across various image modalities. Hence, in this study, we choose edge features from these significant structures—runways and coastlines in the reference images—as the basis for the on-board rectification feature database. We use phase congruency [27] to extract edge features, considering its robustness against intensity variations and noise. Notably, for SAR images, we employ the enhanced version of phase congruency, SAR phase congruency [30], to extract edges. By applying thresholding, the binary edge occupies considerably less space than the original image

while retaining high distinctiveness for template matching. Additionally, our proposed method can be extended to establish a reference database for various scenes. For instance, we might focus on road networks derived from OpenStreetMap in urban scenarios.

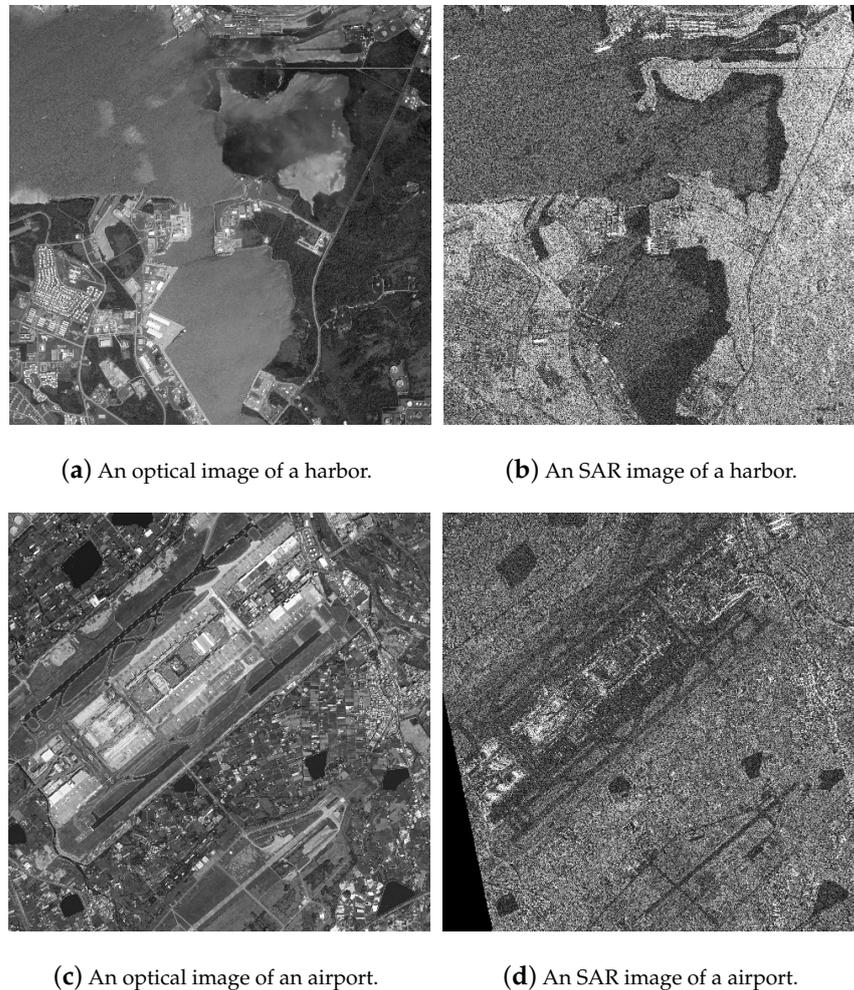
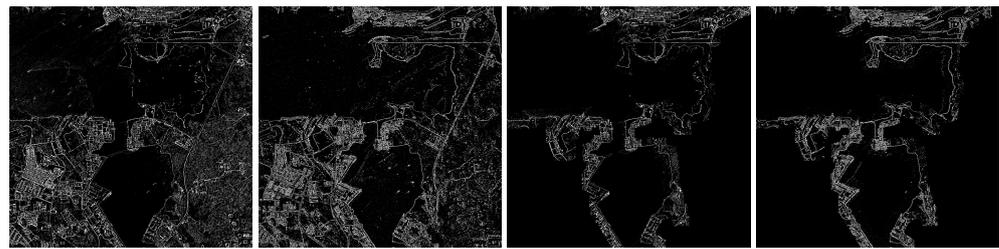


Figure 2. Two examples of real-world optical and SAR images.

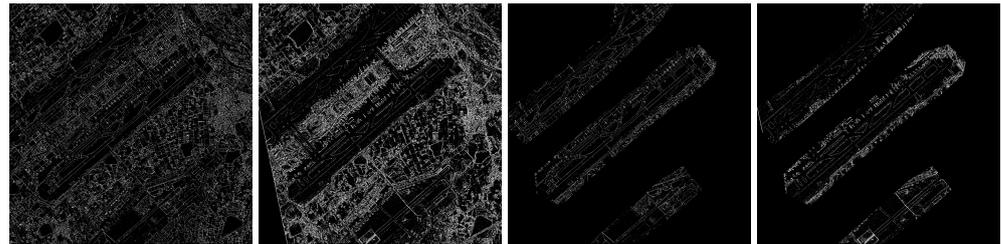
2.1.2. Feature Compression

To reduce redundant storage capacity, we employ ROI through Run-Length Encoding (RLE). RLE is a well-established coding technique that compresses continuous regions of the same value. Following edge extraction and thresholding, the binary edge features and the masking of uninteresting areas generate extensive continuous zero-value regions, making them highly amenable to RLE for redundancy removal. The exact location of runways and coastlines can be pre-determined. By implementing the masking strategy, the reference database size is reduced to between half and one-third of its original size, effectively alleviating storage space constraints.

As depicted in Figure 3, the edge features of coastlines and runways in the airport and harbor images exhibit remarkable similarity, regardless of the imaging modality. However, other areas display inconsistency across multi-modal images, increasing the risk of mismatching. Thus, storing only certain regions of interest in the database maintains performance while consuming less space. For instance, taking Honolulu harbor as an example, the storage space of the reference image amounts to 14.63 MB, approximately 32.4 times larger than that of the proposed feature database (463 KB). Comparatively, our feature database occupies significantly less storage space than the reference images.



(a) Results of the harbor example.



(b) Results of the airport example.

Figure 3. Results of the proposed feature masking strategy. From left to right: edge feature of the optical image, edge feature of the SAR image, masked feature of the optical image, masked feature of the SAR image.

2.2. Two-Step Matching Framework

The current template matching algorithm is unsuitable for on-board geometric rectification due to significant geometric errors that demand an expanded searching window. However, this approach breaches the constraints of timely processing. To overcome this, we propose a two-step matching framework to address the substantial offsets between micro-satellite images and the feature database.

First, we extract binary edge features from the micro-satellite images and identify corresponding features in the database based on geographical scope, defining them as reference and sensed features. Both reference and sensed features are downsampled by a factor of k . This allows the template, at the same size, to manage an initial offset k times larger than the original size. This downsampling process captures additional structural and contextual information from a broader geographic region, enhancing the chance of accurate matching. This initial phase, known as coarse matching, aims to achieve a rough alignment between reference and sensed images. If coarse matching fails, we perform the process once more on roughly matched features and use the result to validate the accuracy of the initial matching. As coarse matching may not guarantee matching accuracy, the subsequent step, termed refined matching, focuses on improving precision based on the roughly matched image pairs. In this phase, we divide the sensed feature into smaller patches and match each patch with the corresponding reference feature. To ensure evenly distributed matching points, we detect Harris corner points on the sensed image to define the center of these small patches and then crop the reference patches based on the coarse matching results. The Random Sample Consensus (RANSAC) method is subsequently employed to eliminate outliers in the matching process.

2.3. On-Board Implementation

Given the conditions on-board satellites, it is advisable to consider embedded GPUs for processing systems, as mentioned in Section 1. In on-board applications that demand real-time processing, we accelerate the proposed matching method using CUDA. Feature detection using phase congruency involves convolving image intensities with a LogGabor kernel in the spatial domain. NCC primarily involves simple summation and multiplication, which are suitable for multi-thread processing using CUDA, providing better efficiency

than phase correlation on this platform. The definition of template matching using NCC is as follows:

$$C(x_c, y_c, dx, dy) = \sum_{(x,y) \in W} \frac{1}{\sigma_R \sigma_S} (I_R(x, y) - \mu_R) * (I_S(x - dx, y - dy) - \mu_S), \quad (1)$$

where I represents feature intensity, and subscripts R and S represent reference and sensed features, respectively. In the reference image, several corner points (x_c, y_c) are selected to calculate the NCC coefficient. (x, y) denotes pixels in template W , and the size of W is denoted as $l \times l$. μ and σ represent the mean and variance of pixels in W , respectively.

$$\mu = \frac{1}{l^2} \sum_{(x,y) \in W} I(x, y), \sigma = \frac{1}{l^2} \sum_{(x,y) \in W} (I(x, y) - \mu)^2. \quad (2)$$

The translation (dx, dy) shifts the template W in the sensed image. For a fixed (x_c, y_c) , we change (dx, dy) to find the maximum NCC as an accurate translation, denoted as

$$(tx, ty)_{(x_c, y_c)} = \arg \max_{(dx, dy)} C(x_c, y_c, dx, dy), |dx| < \text{maxoff}, |dy| < \text{maxoff}. \quad (3)$$

To calculate the NCC value for each (x_c, y_c, dx, dy) , l^2 summations, $2l^2$ submissions, and l^2 multiplications are required. To reduce computational complexity, the calculation can be rewritten as

$$C(x_c, y_c, dx, dy) = \frac{1}{\sigma_R \cdot \sigma_S} * \left(\sum_{(x,y) \in W} I_R(x, y) \cdot I_S(x - dx, y - dy) - l^2 \cdot \mu_R \mu_S \right). \quad (4)$$

The calculation, which only requires l^2 summations and l^2 multiplications, can be further optimized using the Kogge–Stone Adder method to compute all summations in Equation (4). By executing on distinct threads in parallel, the Kogge–Stone Adder can determine the summation of N elements within $\log(N)$ periods.

2.4. Complexity Analysis

In this section, we provide a complexity analysis of the proposed method. Compared to the two-step matching process, the complexity of feature decompression can be disregarded. Assuming the fixed sizes of the reference and sensed images are $n \times n$, and the template W is $l \times l$, the coarse matching step involves the initial and check matching of downsampled images of size $\frac{n^*n}{k^*k}$. One matching process, implemented by phase correlation, consists of two Fast Fourier Transforms (FFTs), one matrix multiplication, and one inverse FFT. Consequently, the time complexity of the coarse matching step is $O\left(12 \frac{n^2}{k^2} \log \frac{n}{k} + 2 \frac{n^2}{k^2}\right)$, where k is the downsampled factor.

The primary time complexity lies in the refined matching step. We select K evenly distributed keypoints and compute the correspondences using NCC. Each NCC of (x_c, y_c, dx, dy) requires l^2 summations and l^2 multiplications, as previously explained. The shifting range (dx, dy) is set as $|dx| < \text{maxoff}, |dy| < \text{maxoff}$. Therefore, the overall complexity of the proposed method is

$$O\left(K * \left(l^2 + l^2\right) * \left(2 * \text{maxoff} + 1\right)^2\right) + O\left(12 \frac{n^2}{k^2} \log \frac{n}{k} + 2 \frac{n^2}{k^2}\right), \quad (5)$$

which can be simplified as

$$O\left(K * l^2 * \text{maxoff}^2\right) + O\left(\frac{n^2}{k^2} \log \frac{n}{k}\right). \quad (6)$$

In practice, we allocate one thread on the GPU for each translation (dx, dy) , each pixel in template W , and each feature point (x_c, y_c) to perform the overall calculation in parallel, enhancing efficiency. Parallel computation minimizes the continuous working period, and the Jetson GPU's advantage of low power consumption contributes to on-board energy efficiency and thermodynamic conditions. Importantly, the GPU optimization does not compromise the accuracy of the proposed method, maintaining identical algorithmic logic to CPU processing. In summary, on-board rectification demands low storage consumption, reduced complexity, and enhanced efficiency while addressing substantial initial offsets. To adapt our method for on-board processing, we undertake the following steps:

1. We develop a lightweight database that concentrates on the most representative area and employs RLE encoding to significantly reduce on-board storage requirements.
2. We introduce a two-step matching framework, which effectively addresses large positioning errors and rectifies the sensed images. Our implementation of the NCC matching method using CUDA for parallel computation aligns with on-board efficiency needs.

3. Experimental Results

The proposed method was evaluated through both on-ground and on-board experiments. Notably, the sensor of the micro-satellites is not restricted, indicating the applicability of the proposed method to both micro-SAR and micro-optical satellites. Consequently, the experiment was conducted in two parts. Firstly, we assessed the rectification performance for micro-SAR images in on-ground experiments. Subsequently, we conducted on-board experiments using the Qilu-3 wide-range micro-optical satellites.

3.1. Results of On-Ground Experiments for Micro-SAR Satellite

The test dataset for on-ground experiments included ten Hisea-1 micro-SAR images captured at Honolulu harbor and thirty-one natural corner reflectors as the ground truth (GT) to assess the rectification accuracy. Seven regions with GTs were cropped from the SAR images, and corresponding reference images were obtained from Google Earth, resulting in thirty-five image pairs used to evaluate positioning accuracy after rectification with matching points. Hisea-1, the initial commercial micro-satellite in China, weighs less than 185 kg and captures images in C-band Spotlight mode with a 1 m resolution. Each patch pair in the test dataset contained multiple GTs with their precise locations manually measured beforehand [31]. Further details of the testing data are outlined in Table 1. The average positioning errors served as the primary metric, supplemented by the percentage of patch pairs with positioning errors smaller than 1, 3, 5, and 10 m to evaluate rectification accuracy. The ROI mask of the test dataset preserved features of roads and streets similar to runways, providing artificial regularity and uniqueness while economizing on storage space.

Table 2 displays the quantitative results of the proposed method and two state-of-the-art comparison methods, including the Robust Oriented Filter-Based Matching Method (OFM) [32] and OSMNet [33]. All methods were implemented on a personal computer equipped with an Intel Core i9 12th generation CPU and an RTX 3060 GPU. The proposed method and OFM were tested in a Windows environment, while OSMNet was tested in the Windows Subsystem for Linux (WSL2).

In the quantitative experiments, the proposed method exhibited lower accuracy than OFM and OSMNet due to its simple algorithm and feature compression, yet it remained acceptable for subsequent applications. Across all 35 image pairs, the maximum rectification error using the proposed matching method was less than 6 m, with an average accuracy of approximately 3 m. Moreover, the proposed matching method required the least storage space and running time. The design of the lightweight feature database and the two-step matching framework effectively addressed the large initial offsets, showcasing favorable performance in terms of storage space, time efficiency, and geometric accuracy. The proposed method's storage space requirement stood at 463 KB, covering reference features and ROI masks. In comparison, OFM necessitated 14.6 MB of original image

storage, while OSMNet required an additional 2.55 MB for pre-training weight storage. Notably, OSMNet needed nearly ten times more storage compared to the proposed method, while OFM consumed significantly more time than the other two methods.

Table 1. Characteristics of test data.

Type	Region	Number	Sensor	Pixel Spacing
On-ground	Honolulu	35	SAR	1 m
On-board	Yokosuka	1	Optical	4 m
On-board	Seattle	1	Optical	4 m
On-board	San Diego	1	Optical	4 m

Table 2. Quantitative comparison of on-ground experiments. Ape denotes the average positioning error.

	<1 m	<3 m	<5 m	<10 m	Ape (m)	Storage	Time (s)
Proposed	8%	56%	92%	100%	2.99	463 KB	6.46
OFM	8%	80%	100%	100%	2.28	14.6 MB	221.18
OsmNet	36%	100%	100%	100%	1.25	17.15 MB	68.53

3.2. Results of On-Board Experiments for Micro-Optical Satellite

3.2.1. Details of On-Board Platform

To introduce the on-board platform, the Qilu-3 wide-range optical payload was developed by the Aerospace Information Research Institute, Chinese Academy of Sciences. It employs four sets of lenses to merge multiple fields of view in the direction of orbit, achieving extensive wide-angle imaging with a sub-satellite point resolution of 4 m. A single camera spans over 30 km in width, contributing to a total stitching width surpassing 120 km. Each frame measures 7920×1200 pixels with 12-bit quantization. Weighing less than 196 kg, the Qilu-3 satellite successfully entered orbit from the Taiyuan Satellite Launch Center in China on 15 January 2023.

The Qilu-3 satellites are equipped with push-broom, gaze-broom, and staring imaging modes, operating under both front-facing and side-swing conditions. Additionally, they feature intelligent information processing and service loads. The intelligent load integrates a dual FPGA (Virtex7+Zynq)+GPU+SATA disk array architecture, enabling on-board high-performance computing, storage, and transmission functions. The GPU handles real-time calculations of the original data. The processing module employs two NVIDIA Tegra X1 (TX1) units for fast parallel data processing according to the main control module's processing requirements. It receives the original data via the PCIe bus and transmits the processed results back to the main control module via Ethernet. Each TX1 unit is equipped with 256 CUDA cores, 4 CPU threads, and 4 GB memory and boasts a single-precision floating-point computing capability of 1 TFLOPS.

3.2.2. Results of On-Board Experiments

Before conducting the on-board experiments, we selected nine distinct areas to establish the database, which was then stored on the platform's disk. Utilizing Google Earth images with a 1 m resolution as the reference images, we expanded the ROI masks several times to ensure reliability. In this scenario, the original images occupied 199 MB, almost 70 times larger than the proposed lightweight database (2.87 MB).

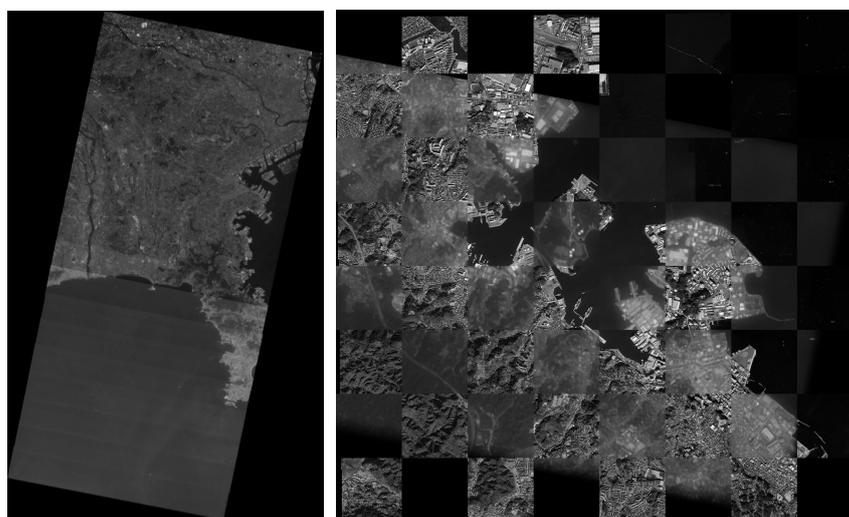
To thoroughly evaluate the efficacy of the proposed method, three test areas—Seattle, San Diego, and Yokosuka—were chosen for the on-board experiments. Matching the adjacent frames of testing data was simplified as they were captured in the same imaging modes and parameters. Each frame was then matched to the database using the proposed two-step framework. If one frame failed, we seamlessly moved on to the next frame. Once

a particular frame was rectified with the reference database, the same transformation model was applied to the remaining frames to prevent redundant computation. Ultimately, we merged all frames to generate a single wide-range product with a precise location. The results from different transformation models were then evaluated.

The results of the on-board rectification are detailed in Table 3. For an accurate assessment, we manually selected 10 checkpoints in each image pair to calculate rectification accuracy. Initially, the positioning error could be several kilometers, but after rectification, it was notably reduced to approximately 10 m. This reduction equaled nearly three pixels in Qilu-3 wide-range images. Our findings indicated that the affine model offered better accuracy, while the translation model provided higher efficiency. In comparison to the on-ground results listed in Table 2, the proposed method using the affine transformation model exhibited an average error of about three pixels. The translation model, while less accurate, could swiftly apply the matching result of one image to adjacent frames, ensuring higher efficiency. Furthermore, the size of the reference images was roughly 20 times larger than the proposed database, aligning with the results from the ground experiment. In Figure 4, the comparison between the target regions' chessboards and the reference images validates the accuracy of our rectification results.

Table 3. Rectification results of on-board experiments on three test areas. Image refers to the storage of raw images, and Database refers to the storage of the on-board database. CPU-m and GPU-m denote the time consumption of the proposed two-step matching framework using only CPU or GPU on the Nvidia TX1 platform. It should be noted that the transformation time of the overall image was not considered in the calculation.

	Initial	Translation	Affine	Image	Database	CPU-m	GPU-m
Yokosuka	11 km	22.85 m (5.71 pixels)	8.17 m (2.04 pixels)	5008 KB	183 KB	31.10 s	7.26 s
Seattle	10 km	22.31 m (5.58 pixels)	7.41 m (1.85 pixels)	583 KB	28 KB	15.22s	3.84 s
San Diego	2 km	12.34 m (3.09 pixels)	6.95 m (1.74 pixels)	1046 KB	39 KB	23.23 s	4.75 s



(a) Yokosuka

(b) Chessboard of Yokosuka after rectification

Figure 4. Cont.

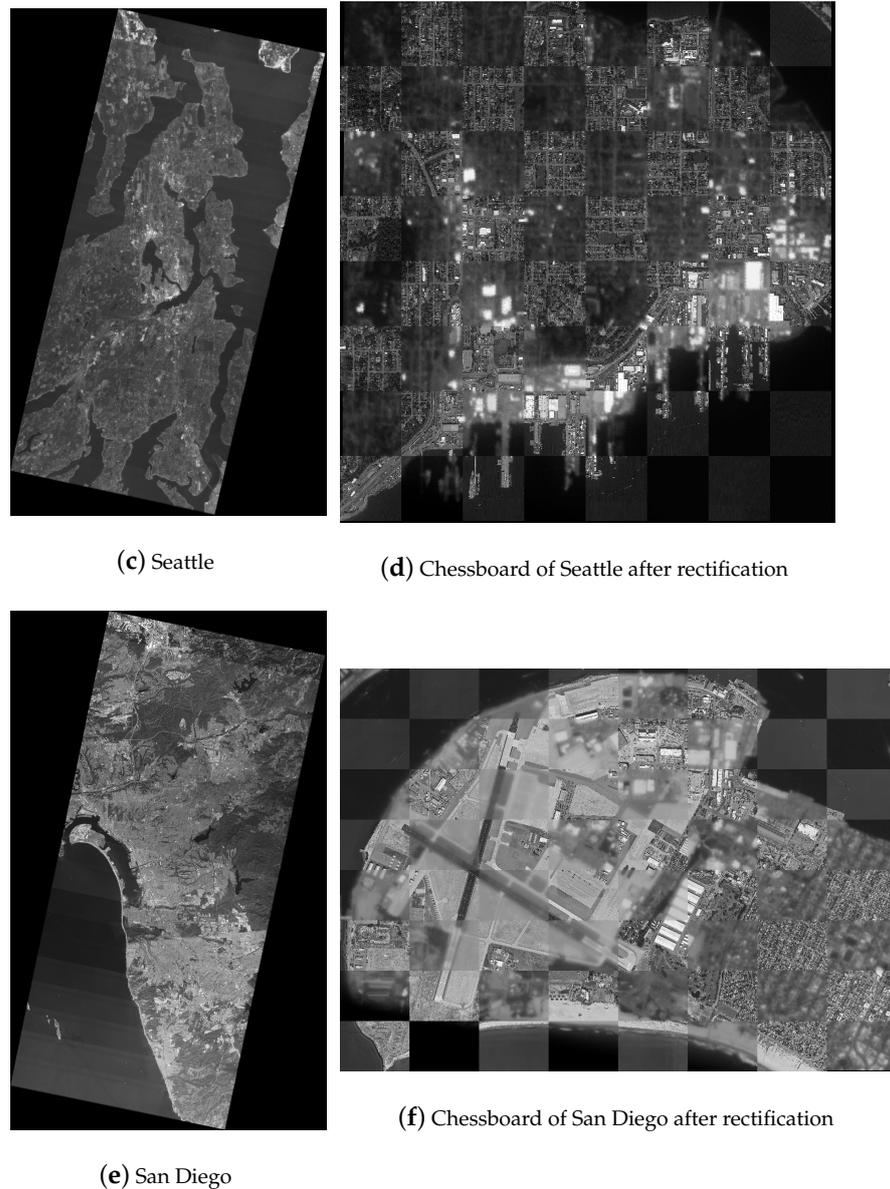


Figure 4. The rectification results of on-board wide-range experiment.

To emphasize the efficiency improvement, we present the time consumption for the CPU and GPU of the TX1 platform. With 256 CUDA cores and 4 CPU threads, the GPU architecture was more conducive to parallel computing. As a result, the matching process accelerated by the GPU was nearly five times faster than that of the CPU. This significant reduction in time demonstrates the effectiveness of the GPU implementation.

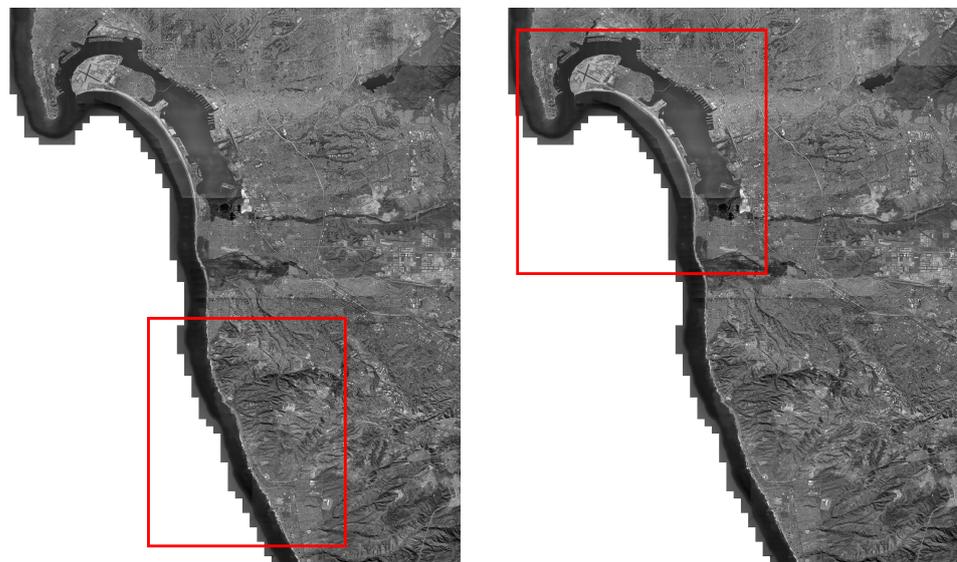
4. Discussion

Since the on-board rectification on the Qilu-3 micro-satellite was a preliminary experiment, we constructed a feature database for only nine target regions. For other regions, we present a discussion on the selection of appropriate features and ROI masks, along with an analysis of the boundary conditions of the proposed method.

The Qilu-3 micro-satellite is primarily intended for observing activities in major cities worldwide. Hence, for this on-board experiment, we selected runways and coastlines as primary features for our reference database. The proposed feature detection and compression method can be applied to establish a reference database for various scenes. For instance, in urban scenarios, focusing on the road network obtained from OpenStreetMap

or using irrigation canals to build reference features for farmland could be viable options. The criteria for selecting features, outlined in Section 2.1, encompass attributes such as intrinsicality, invariance, robustness, and uniqueness. However, regions like mountains or deserts, which possess intricate textures and lack artificial facilities, might pose challenges in extracting suitable features to meet these requirements, potentially causing rectification failure. Further analysis for these areas is on our agenda.

The database size is also related to the limits of our approach. Complete runway structures at airports typically cover a range of 3 to 4 km. Therefore, the proposed database size of 5000×5000 pixels suffices for airport scenarios. Notably, airport structures, apart from minor changes, remain relatively constant for prolonged periods. Hence, under typical circumstances, on-board airport databases do not require frequent updates. On the other hand, the applicability of coastlines is more complex. The database size hinges on coastline shapes and the geographical scope of sensed images to ensure adequate information for rectification. In our experiment, the 5000×5000 pixels encompassed distinct coastline structures (depicted in Figure 5b) to support the proposed method. However, if the coastlines are excessively simple, such as featuring nearly linear structures (as shown in Figure 5a), the proposed method may not accurately rectify the sensed images.



(a) A section of coastline without uniqueness. (b) A section of coastline with adequate uniqueness.

Figure 5. Feature uniqueness of different geographical scopes.

When coastlines change, the necessity for updates depends on the scale of the changes. The rectification method primarily emphasizes unchanged regions. Therefore, if only a small part of the coastline is changed or the general structure remains only slightly altered, the existing information in the database remains valid. However, in the event of a significant change due to a natural disaster or a complete alteration in the coastline's structure, we recommend re-uploading the database. Similar advice applies to particularly uninteresting terrains that may have valuable new information. In our future research, we will explore an on-board database updating mode to address changes and new targets.

The effectiveness of the proposed feature database also relies on the ROI mask to reduce storage and maintenance. Choosing an appropriate ROI is a critical aspect of the proposed method. The accurate location of airport runways and coastlines can be acquired in advance from OpenStreetMap. As discussed in Section 3, the vicinity of runways and coastlines should be partially covered by ROI masks to ensure reliability for template matching. However, this balance is crucial. Preserving too little surrounding information might result in severe mismatching while retaining too much redundant information

consumes unnecessary storage. For airports, dilating the runway regions a few times ensures sufficient structural information. Regarding coastlines, precise land coverage maps offer a one-pixel coastline for extraction using edge detection methods. Dilating the one-pixel coastline a few times (ideally between 10 to 20 pixels) provides a reliable mask. For intricate or fragmented coastlines, an increased dilation count accommodates more information for template matching, while simpler coastlines may reduce dilation counts to save storage.

For the on-board platform of Qilu-3, we stored feature databases of nine test areas, as this was merely a preliminary on-board experiment. However, for a worldwide database, which is crucial for practical applications, the expected size corresponds to the number of target regions and coverage.

5. Conclusions

This paper introduced a lightweight feature database construction method and a two-step matching framework for on-board rectification. Results from both on-ground and on-board experiments affirmed the method's advantages in terms of low storage requirements and high efficiency, making it an ideal choice for on-board rectification. The primary advantages are summarized as follows:

1. Robust and easy-to-implement feature database. Focused on runways and coastlines, the phase congruency feature significantly reduces storage requirements (to approximately 1/20) while retaining essential data for accurate rectification.
2. Feature compression strategy. Using ROI and RLE techniques, storage for the lightweight database decreases by over 50%. The compressed database occupies only 1/70 of the original storage, ensuring rectification accuracy within 10 m.
3. Two-step matching framework. The method employs a coarse matching step, utilizing a robust global FFT to handle large initial positioning errors. In the refined matching, evenly distributed keypoints ensure precise rectification.
4. The selection of NCC for template matching. Leveraging its suitability for parallel computing, the NCC method is implemented using the Kogge–Stone Adder and migrated to the TX1 GPU for accelerated processing.

Author Contributions: Conceptualization, L.W. and Y.X.; methodology, L.W.; software, L.W. and Y.X.; validation, L.W. and Z.W.; formal analysis, L.W.; investigation, L.W.; resources, Z.W. and Y.H.; data curation, Z.W.; writing—original draft preparation, L.W.; writing—review and editing, Y.X. and H.Y.; visualization, Z.W.; supervision, H.Y. and Y.H.; project administration, Y.H.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research Program of Frontier Sciences, Chinese Academy of Science, under grant number ZDBS-LY-JSC036.

Data Availability Statement: Restrictions apply to the availability of these data. Hisea-1 images were accessed on 1 December 2022 from <https://data.spacety.com/home>. OpenStreetMap data are available under the open database license. Map data are copyrighted by OpenStreetMap contributors and available from <https://planet.osm.org/> (accessed on 1 December 2022). The Google Earth images were downloaded from the Google Earth platform.

Acknowledgments: The authors would like to thank the editors and anonymous reviewers for their constructive comments and suggestions. The authors would also like to thank Spacety for providing the Hisea-1 SAR images.

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

References

1. Salazar, C.; Gonzalez-Llorente, J.; Cardenas, L.; Mendez, J.; Rincon, S.; Rodriguez-Ferreira, J.; Acero, I.F. Cloud Detection Autonomous System Based on Machine Learning and COTS Components On-Board Small Satellites. *Remote Sens.* **2022**, *14*, 5597. [[CrossRef](#)]
2. Sweeting, M.N. Modern small satellites-changing the economics of space. *Proc. IEEE* **2018**, *106*, 343–361. [[CrossRef](#)]

3. Crusan, J.; Galica, C. NASA's CubeSat Launch Initiative: Enabling broad access to space. *Acta Astronaut.* **2019**, *157*, 51–60. [[CrossRef](#)]
4. Villela, T.; Costa, C.A.; Brandao, A.M.; Bueno, F.T.; Leonardi, R. Towards the Thousandth CubeSat: A Statistical Overview. *Int. J. Aerosp. Eng.* **2019**, *2019*, 5063145. [[CrossRef](#)]
5. Wang, M.; Zhang, Z.; Zhu, Y.; Dong, Z.; Li, Y. Embedded GPU implementation of sensor correction for on-board real-time stream computing of high-resolution optical satellite imagery. *J. Real-Time Image Process.* **2018**, *15*, 565–581. [[CrossRef](#)]
6. Huo, C.; Zhou, Z.; Ding, K.; Pan, C. Online Target Recognition for Time-Sensitive Space Information Networks. *IEEE Trans. Comput. Imaging* **2017**, *3*, 254–263. [[CrossRef](#)]
7. Xu, P.; Li, Q.; Zhang, B.; Wu, F.; Zhao, K.; Du, X.; Yang, C.; Zhong, R. On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning. *Remote Sens.* **2021**, *13*, 1995. [[CrossRef](#)]
8. Qi, B.; Shi, H.; Zhuang, Y.; Chen, H.; Chen, L. On-Board, Real-Time Preprocessing System for Optical Remote-Sensing Imagery. *Sensors* **2018**, *18*, 1328. [[CrossRef](#)] [[PubMed](#)]
9. Yao, Y.; Jiang, Z.; Zhang, H.; Zhou, Y. On-Board Ship Detection in Micro-Nano Satellite Based on Deep Learning and COTS Component. *Remote Sens.* **2019**, *11*, 762. [[CrossRef](#)]
10. Zhang, Y.; Chi, Z.; Hui, F.; Li, T.; Liu, X.; Zhang, B.; Cheng, X.; Chen, Z. Accuracy Evaluation on Geolocation of the Chinese First Polar Microsatellite (Ice Pathfinder) Imagery. *Remote Sens.* **2021**, *13*, 4278. [[CrossRef](#)]
11. Zheng, J.; Chen, Q.; Yan, X.; Ren, W. Challenges for next generation micro-SAR: Lessons learned from China's first light and small commercial SAR satellite-Hisea-1. In Proceedings of the EUSAR 2022; 14th European Conference on Synthetic Aperture Radar, VDE, Leipzig, Germany, 25–27 July 2022; pp. 1–4.
12. Liu, D.; Zhou, G.; Zhang, D.; Zhou, X.; Li, C. Ground Control Point Automatic Extraction for Spaceborne Georeferencing Based on FPGA. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3350–3366. [[CrossRef](#)]
13. Ke, N.Y.; Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004.
14. Gang Hua, E.A. Discriminant Embedding for Local Image Descriptors. In Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007.
15. Calonder, M.; Lepetit, V.; Fua, P.; Konolige, K.; Bowman, J.; Mihelich, P. Compact signatures for high-speed interest point description and matching. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009. [[CrossRef](#)]
16. Jégou, H.; Douze, M.; Schmid, C. Product Quantization for Nearest Neighbor Search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 117–128. [[CrossRef](#)] [[PubMed](#)]
17. Jingjin, H.; Guoqing, Z.; Xiang, Z.; Rongting, Z. A New FPGA Architecture of FAST and BRIEF Algorithm for On-Board Corner Detection and Matching. *Sensors* **2018**, *18*, 1014.
18. Low, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
19. Bay, H.; Tuytelaars, T.; Gool, L.V. SURF: Speeded up robust features. In Proceedings of the 9th European conference on Computer Vision-Volume Part I, Graz, Austria, 7–13 May 2006.
20. Xie, G.; Zhang, Z.; Zhu, Y.; Xiang, S.; Wang, M. On-Board GCPs Matching With Improved Triplet Loss Function. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *V-2-2020*, 105–112. [[CrossRef](#)]
21. Liu, D.; Zhou, G.; Huang, J.; Zhang, R.; Shu, L.; Zhou, X.; Xin, C. On-Board Georeferencing Using FPGA-Based Optimized Second-Order Polynomial Equation. *Remote Sens.* **2019**, *11*, 124. [[CrossRef](#)]
22. Wang, G.; Chen, H.; Xie, Y. An Efficient Dual-Channel Data Storage and Access Method for Spaceborne Synthetic Aperture Radar Real-Time Processing. *Electronics* **2021**, *10*, 662. [[CrossRef](#)]
23. Jingjin, H.; Guoqing, Z.; Dianjun, Z.; Guangyun, Z.; Rongting, Z.; Oktay, B. An FPGA-based implementation of corner detection and matching with outlier rejection. *Int. J. Remote Sens.* **2018**, *39*, 8905–8933.
24. Xu, M.; Chen, L.; Shi, H.; Yang, Z.; Li, J.; Long, T. FPGA-Based Implementation of Ship Detection for Satellite On-Board Processing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 9733–9745. [[CrossRef](#)]
25. Zhou, G.; Zhang, R.; Zhang, D.; Huang, J.; Baysal, O. Real-time ortho-rectification for remote-sensing images. *Int. J. Remote Sens.* **2018**, *40*, 2451–2465. [[CrossRef](#)]
26. Xie, G.; Wang, M.; Zhang, Z.; Xiang, S.; He, L. Near Real-Time Automatic Sub-Pixel Registration of Panchromatic and Multispectral Images for Pan-Sharpener. *Remote Sens.* **2021**, *13*, 3674. [[CrossRef](#)]
27. Kovese, P. Image features from phase congruency. *Videre J. Comput. Vis. Res.* **1999**, *1*, 1–26.
28. Golomb, S. Run-length encodings (corresp.). *IEEE Trans. Inf. Theory* **1966**, *12*, 399–401. [[CrossRef](#)]
29. Li, L.; Han, L.; Ye, Y. Self-supervised keypoint detection and cross-fusion matching networks for multimodal remote sensing image registration. *Remote Sens.* **2022**, *14*, 3599. [[CrossRef](#)]
30. Wang, L.; Xiang, Y.; You, H.; Qiu, X.; Fu, K. A Robust Multi-scale Edge Detection Method for Accurate SAR Image Registration. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 4006305.
31. Xiang, Y.; Wang, X.; Wang, F.; You, H.; Qiu, X.; Fu, K. A Global-to-local Algorithm for High-resolution Optical and SAR Image Registration. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5215320. [[CrossRef](#)]

32. Fan, Z.; Wang, M.; Pi, Y.; Liu, Y.; Jiang, H. A Robust Oriented Filter-Based Matching Method for Multisource, Multitemporal Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4703316. [[CrossRef](#)]
33. Zhang, H.; Lei, L.; Ni, W.; Tang, T.; Wu, J.; Xiang, D.; Kuang, G. Explore Better Network Framework for High-Resolution Optical and SAR Image Matching. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4704418. [[CrossRef](#)]

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.