The Journal of Supercomputing (2026) 82:513
https://doi.org/10.1007/s11227-026-08611-5

Distributed multi-GPU algorithm for accurate registration of UAV-based multispectral and multitemporal orthomosaics

Daniel Fuentes[1] · Álvaro Ordóñez[1,2] · Dora B. Heras[1,2] · Francisco Argüello[2]

Received: 15 October 2025 / Accepted: 18 May 2026

Abstract

Accurate registration of high-resolution multispectral UAV orthomosaics acquired on different dates and sensors is essential for a wide range of remote sensing applications. However, this task remains challenging due to variations in acquisition conditions, including seasonal changes, differences in illumination, weather, and sensor characteristics. This article presents a parallel multilevel registration method that combines and improves two existing algorithms: HSI-KAZE, a feature-based approach, and HYFM, an area-based method. The three-level approach first applies an optimized HSI-KAZE (OHSI-KAZE) for coarse estimation of scale, rotation, and translation, followed by HYFM for fine correction. The multi-node multi-GPU proposed implementation, leveraging MPI, OpenMP, and CUDA, enables efficient processing on GPU-accelerated HPC clusters. Experiments on six real multispectral orthomosaic pairs from river environments, compared against state-of-the-art classical and deep learning methods, achieve high registration accuracy with RMSE values below 1.57 pixels and a 30× speedup, confirming both the accuracy and scalability of the proposed method.

Keywords Image registration · Remote sensing · Multispectral · High performance computing · Parallel computing · GPU

✉ Daniel Fuentes
danielfuentes.rodriguez@usc.gal

✉ Álvaro Ordóñez
alvaro.ordonez@usc.gal

Dora B. Heras
dora.blanco@usc.gal

Francisco Argüello
francisco.arguello@usc.gal

[1] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain

[2] Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain

Published online: 20 June 2026

1 Introduction

Over the last few years, one of the most important advances in remote sensing has been the widespread adoption of multispectral sensors, driven by their growing affordability. As a result, multispectral sensors mounted on UAV-based systems have become essential tools in a wide range of applications as an alternative due to their low cost, including precision agriculture [1], environmental monitoring [2], forest management [3], and ecosystem change detection [4], among others. Generally, UAVs capture hundreds of multispectral images, known as frames, during a single flight. These frames are then aligned, orthorectified, and merged into a single image, called an orthomosaic. This process, particularly when multispectral sensors are used, results in high-resolution multispectral orthomosaics.

When working with images or orthomosaics captured on different dates for the same geographical area, multitemporal imagery, image registration is required to enable tasks such as change detection [5]. Registration is the process of obtaining the geometric transformations required to align two or more images so that the corresponding pixels in each image refer to the same physical location [6].

Ensuring accurate alignment is critical as registration errors propagate and compromise the reliability of subsequent processing stages, leading to incorrect change detection results, false anomalies, or misclassifications [5]. In many applications, high accuracy is particularly important. For example, even small pixel misalignment has been shown to introduce errors exceeding 50% in the calculation of indices such as the Normalized Difference Vegetation Index (NDVI) [7]. Achieving high accuracy becomes even more challenging when dealing with multitemporal images affected by seasonal variations, illumination differences, vegetation growth, weather conditions, or acquired by different sensors, all of which can introduce significant radiometric and geometric inconsistencies [8].

Moreover, residual registration errors often persist even after generating orthomosaics from orthorectified frames, which have been geometrically corrected to remove sensor and terrain distortions and adjusted to uniform scale and map projection [9]. These residual errors typically arise from limitations in the software used for orthomosaic construction. All these factors prevent a single method from achieving consistent and accurate alignment. This highlights the need for multilevel registration approaches, which combine complementary techniques to progressively refine alignment and handle complex variations in scale, rotation, and translation.

Several methods have been proposed in the literature for the registration of remote sensing images captured on different dates. For example, Puniach et al. presented [10] an automated pipeline to determine horizontal displacements using high-resolution RGB orthomosaics captured by UAVs. It is based on the computation of the weighted normalized cross correlation between two images, using LiDAR data to weight the reliability of the pixels. Recently, deep learning-based image registration methods have emerged as a prominent trend due to their ability to learn high-level semantic feature representations. To address appearance variations in multitemporal RGB images, Yang et al. [11] proposed a registration method that combines multi-scale feature descriptors extracted from a convolutional neural network (CNN) with a dynamic inlier selection strategy. Similarly, Zeng et al. [12] introduced a CNN-based image registration tech-

nique for mosaic generation. This method extracts hierarchical convolutional features and matches them using a correlation filter to evaluate the similarity between feature points.

This ability to learn robust semantic features has also made deep learning (DL) highly suitable for multimodal registration tasks. For example, Xiao et al. [13] proposed ADRNet, a CNN-based network for aligning visible-infrared and visible-SAR image pairs. ADRNet is based on an end-to-end network architecture that combines an affine registration module to correct large-scale deformations with a deformable registration module to capture and learn local disparities. Ye et al. [14] introduced an unsupervised method called MU-Net for the registration of optical-optical, optical-infrared, and optical-SAR datasets. The method is based on stacking several deep neural network (DNN) models to create a coarse-to-fine registration pipeline that directly learns transformation parameters by optimizing a structural similarity loss function. Furthermore, Shi et al. presented CHA-Net [15], an unsupervised network for the non-rigid registration of multispectral-panchromatic, visible-NIR, and RGB-map image pairs. CHA-Net is based on domain adaptation, utilizing a Siamese feature decoupling (SFD) structure to weaken modality-specific style differences and a transformer-based hierarchical refinement alignment (HRA) structure to progressively capture and correct both global and local geometric distortions. Despite their promising capabilities, these DL solutions face significant challenges in real-world scenarios. Due to the scarcity of labeled datasets, these methods rely on pre-registered images augmented with synthetic transformations for network training. Consequently, they often overfit to artificial conditions and struggle to real multitemporal orthomosaics characterized by significant variations in scale, rotation, and shift.

Scaling registration to large, multitemporal, multispectral datasets at high spatial resolution requires more computationally efficient methods. In this context, several studies have explored cost-reduction strategies or the use of parallelization on hardware accelerators such as GPUs. For instance, Zhang et al. [16] introduced a GPU-accelerated coarse-to-fine strategy for registering very high-resolution (VHR) multispectral images on a single GPU. Similarly, Huo et al. [17] developed a multilevel SIFT-based method that reduces computational demands while preserving accuracy; however, their approach is fully implemented on the CPU. Most of these techniques are tailored primarily to RGB or panchromatic imagery, suggesting a lack of methods specifically designed for large-scale multispectral datasets. Moreover, although a few parallel CPU and GPU implementations exist, the potential of multi-GPU and multi-node architectures for accelerating the registration of such data remains largely unexplored.

To address the limitations described above, this article presents a method for registering high-resolution multispectral orthomosaics. The registration technique performs multilevel registration by combining an optimized version of Hyperspectral KAZE (OHSI-KAZE) [18], a feature-based method, and Hyperspectral Fourier-Mellin (HYFM) [19], an area-based alignment algorithm, in a progressive refinement approach. Feature-based methods such as HSI-KAZE offer robust performance in scenarios that involve severe misalignments caused by significant variations in scale, rotation, illumination, or sensor-specific characteristics [6]. In particular, the proposed optimized HSI-KAZE algorithm improves the accuracy of rotation angle calculation.

In the case of area-based methods, they may be less effective in the presence of large-scale discrepancies between images, but they demonstrate excellent accuracy in refining minor residual misalignments, as is the case with the HYFM algorithm [8, 20]. Furthermore, the proposed method is computationally efficient thanks to image registration based on fixed-size regions and its parallel implementation across multi-node and multi-GPU architectures, enabling the efficient processing of large sets of orthomosaics. It also supports registration using multiple GPUs and concurrent alignment of different pairs across separate nodes in a high-performance computing (HPC) cluster.

The main contributions of this work are the following:

- A multilevel registration method that combines HSI-KAZE (feature-based) and HYFM (area-based) algorithms is proposed for accurate alignment of high-resolution, multitemporal, and multispectral UAV orthomosaics captured by different multispectral sensors.
- An optimized HSI-KAZE (OHSI-KAZE) algorithm is presented to increase accuracy in calculating rotation angle differences between multispectral orthomosaics.
- The multilevel method is designed to reduce computational cost by performing registration based on fixed-size regions, independent of the dataset size.
- The method is further optimized by its implementation for multi-GPU and multi-node HPC clusters using MPI, OpenMP, and CUDA, enabling the processing of large sets of orthomosaics with reduced execution time.
- A new dataset for registration is proposed. It comprises pairs of real high–spatial-resolution multispectral orthomosaics captured by different sensors across different years. For each pair, the dataset includes the reference registration parameters: rotation, scale, and horizontal and vertical shifts.

The article is structured as follows. Section 2 presents the proposed multilevel registration method, highlighting the advantages of the optimized HSI-KAZE and HYFM for progressive alignment, and detailing their CUDA implementations. Section 3 discusses the parallelization strategies followed for the multi-GPU and multi-node implementation. Section 4 presents the experimental setup and results on registration accuracy and computational performance. The implementation's scalability is also analyzed, concluding with a comprehensive comparison against state-of-the-art methods. Finally, the conclusions are outlined in Sect. 5.

2 Multilevel registration method

This section explains the multilevel registration method proposed in this paper. It is designed to efficiently handle high-resolution multitemporal, multisensor, and multispectral orthomosaics.

Figure 1 presents a pair of multispectral orthorectified orthomosaics captured onboard UAVs to illustrate the registration process performed by the method. The orthomosaics contain some pixels with no information (black regions in the figure), as data are only available for the areas actually overflown by the UAV. As shown in the figure, each pair of orthomosaics is registered based on only one region per orthomo-

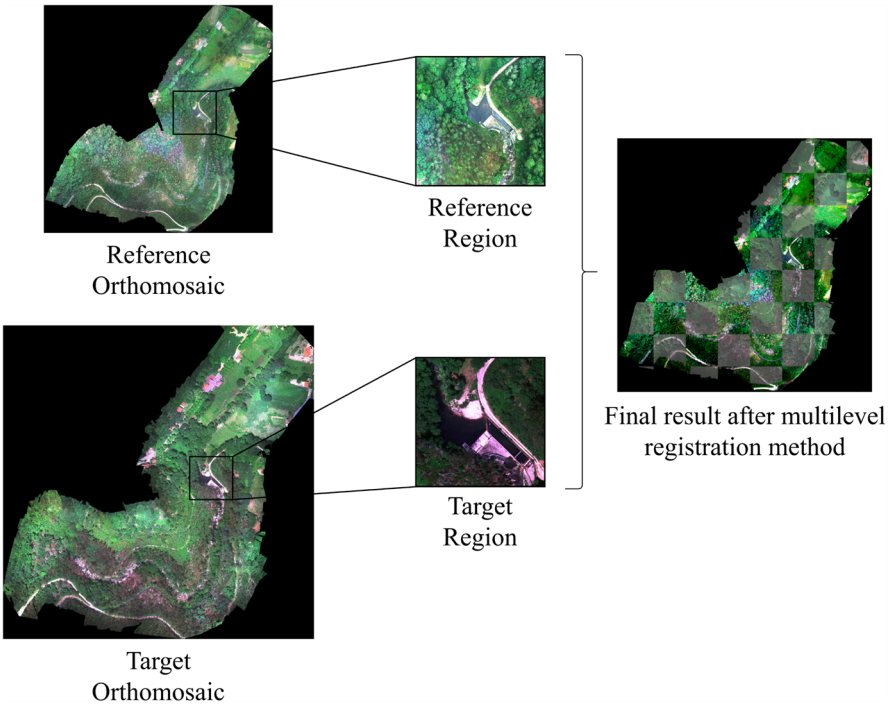

Fig. 1 Overview of the registration process for accurate alignment of two multispectral orthomosaics. The results are shown as a chessboard visualization

saic. Then, the multilevel registration method is applied to these regions. The resulting transformation is then applied to the entire image.

To decide the best methods for the proposed registration levels, a review has been conducted to identify the most accurate registration methods for multi and hyperspectral remote sensing images. Starting with feature-based methods, the Scale-Invariant Feature Transform (SIFT) [21] is one of the most widely used methods, composed of four stages: scale-space extrema detection, keypoint detection, orientation assignment, and keypoint description. The method constructs a scale-space pyramid using Gaussian filters and computes the difference-of-Gaussians (DoG) to detect keypoints. These keypoints are refined by discarding those with low contrast or unstable location and scale. Once refined, a dominant orientation is assigned to them to ensure rotation invariance, and descriptors of length 128 are generated. These descriptors are then used to match keypoints between the two images to be registered. For this reason, SIFT includes a descriptor that is robust to noise, illumination changes, and minor geometric misalignments. Despite its robustness and accuracy, SIFT is computationally intensive, which has motivated the development of alternative methods to improve its efficiency [22].

Following the development of SIFT, several methods were proposed to enhance its performance. One such method is Speeded Up Robust Features (SURF) [23], which was designed to reduce the computational cost. SURF achieves this by using an approx-

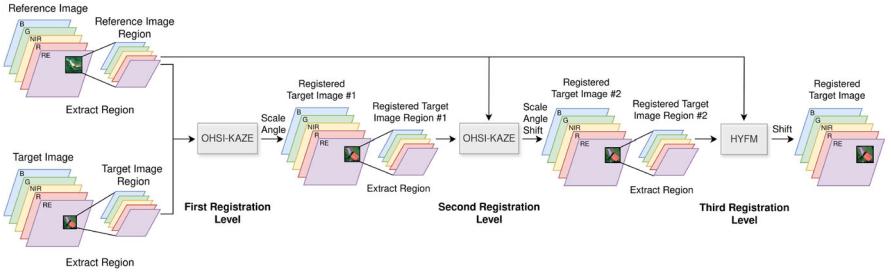

Fig. 2 Proposed multilevel registration method consisting of three levels to progressively improve the registration accuracy. For each pair of orthomosaics, only a region for each image is extracted, and the registration parameters are calculated over that region

imation of the Gaussian scale-space that can be efficiently computed, along with other algorithmic optimizations. However, this computational improvement comes at the cost of reduced keypoint detection accuracy and descriptor distinctiveness, which negatively affects registration accuracy.

Another relevant feature-based method is HSI-KAZE [18], which is based on KAZE features [24]. The method was specifically designed to exploit spectral information in multi and hyperspectral images by performing keypoint detection across multiple bands and integrating spectral information with the M-SURF spatial descriptor. Compared to SIFT, the HSI-KAZE method builds the scale-space using nonlinear diffusion filters, achieving similar accuracy at lower computational cost. Additionally, this method was designed to operate under challenging conditions, including large variations in scale and rotation, as well as illumination changes and noise.

Unlike feature-based methods, area-based methods do not have the capability to correct large differences in scale, rotation, or displacement [25], but rather focus on refining small misalignments between images at reduced computational cost. HYFM [19] stands out as an area-based method that uses the fractional multilayer Fourier transform, a combination of log-polar maps, and peak processing, to achieve accurate alignments in hyperspectral remote-sensing images.

Even though the described methods are very efficient in terms of registration accuracy, they cannot achieve satisfactory results when used individually to register images acquired with different sensors and in different years, as these factors introduce more variability in image characteristics. The significant differences among the images, including high variations in scale, rotation, and shifts, require an iterative refinement process that combines multiple methods to effectively address these challenges.

The proposed registration method is outlined in Fig. 2. It comprises three sequential levels that refine the transformation obtained in the preceding level. OHSI-KAZE, an optimized version of HSI-KAZE for finer adjustments, was selected for the first two levels. For the final level, HYFM was chosen for its GPU performance and effectiveness in registration refinement.

As illustrated in Fig. 2, the process starts by extracting the regions. The user selects a reference point in each image, and a region centered on that point is extracted. These two regions are then registered by OHSI-KAZE at the first registration level,

which provides a scale and rotation transformation that is applied to the entire target image. After the initial registration process, as shown in Fig. 2, a new target region centered at the same reference point is extracted from the registered target image, which has been scaled and rotated according to the transformation obtained in the first registration level. Thanks to this correction of scale and rotation, the new target region exhibits greater overlap with the reference region, enabling OHSI-KAZE at the second registration level to detect additional features and thereby improving the overall accuracy of the registration process. In this level, shift differences are corrected, while the scale and rotation parameters are further refined.

After applying the second registration level, some residual shift registration errors may remain. To correct them, a third level of low-computational-cost refinement will be performed. Again, a new target region is extracted from the new registered target image, which has been scaled, rotated, and shifted according to the transformations obtained in the previous levels. At this third level, the HYFM method is applied to the pair of regions, yielding the final registered target image.

Algorithm 1 Multilevel registration method

Input: Reference orthomosaic $\mathbf{O}_\mathrm{r}$ (N_B bands), target orthomosaic $\mathbf{O}_\mathrm{t}$ (N_B bands), region center coordinates (x_r, y_r) and (x_t, y_t)

Output: Registered target orthomosaic $\mathbf{O}_\mathrm{t}^\mathrm{reg}$, scale factor ρ_3, rotation angle θ_3, and shift (x_3, y_3)

$\mathbf{R}_\mathrm{r}, \mathbf{R}_\mathrm{t} \leftarrow$ Extract corresponding regions from $\mathbf{O}_\mathrm{r}$ at (x_r, y_r) and $\mathbf{O}_\mathrm{t}$ at (x_t, y_t) ▷ CPU

Level 1: Coarse Scale + Rotation

$(\rho_1, \theta_1) \leftarrow \mathbf{OHSI\text{-}KAZE}(\mathbf{R}_\mathrm{r}, \mathbf{R}_\mathrm{t})$ ▷ *see Alg. 2*

$\mathbf{O}_\mathrm{t}^1 \leftarrow$ Apply transformation (ρ_1, θ_1) to $\mathbf{O}_\mathrm{t}$ ▷ GPU

Level 2: Coarse Shift

$\mathbf{R}_\mathrm{t}^1 \leftarrow$ Extract region from $\mathbf{O}_\mathrm{t}^1$ at (x_r, y_r) ▷ CPU

$(\rho_2, \theta_2, x_2, y_2) \leftarrow \mathbf{OHSI\text{-}KAZE}(\mathbf{R}_\mathrm{r}, \mathbf{R}_\mathrm{t}^1)$ ▷ *see Alg. 2*

$\mathbf{O}_\mathrm{t}^2 \leftarrow$ Apply transformation $(\rho_2, \theta_2, x_2, y_2)$ to $\mathbf{O}_\mathrm{t}^1$ ▷ GPU

Level 3: Fine Adjustment

$\mathbf{R}_\mathrm{t}^2 \leftarrow$ Extract region from $\mathbf{O}_\mathrm{t}^2$ at (x_r, y_r) ▷ CPU

$(\rho_3, \theta_3, x_3, y_3) \leftarrow \mathbf{HYFM}(\mathbf{R}_\mathrm{r}, \mathbf{R}_\mathrm{t}^2)$ ▷ *see Alg. 3*

$\mathbf{O}_\mathrm{t}^\mathrm{reg} \leftarrow$ Apply transformation $(\rho_3, \theta_3, x_3, y_3)$ to $\mathbf{O}_\mathrm{t}^2$ ▷ GPU

The rest of this section describes the single-GPU implementation of the multilevel registration method presented in Algorithm 1. The implementation relies on custom CUDA kernels developed by the authors for the core registration stages, complemented by optimized libraries (cuFFT, cuBLAS, cuSOLVER, NPP, and Thrust) for established operations such as FFT computation, matrix multiplication, and image interpolation. In particular, Sect. 2.1 describes the details of extracting corresponding regions from the reference and target orthomosaics. The following subsections detail each of the three registration levels: Sect. 2.2 explains the coarse correction of scale and rotation parameters using the proposed OHSI-KAZE; Sect. 2.3 details the subsequent use of OHSI-KAZE to correct shift misalignments while refining the initial scale and rotation estimates; and Sect. 2.4 describes the final fine adjustment level performed by HYFM.

2.1 Region extraction

The registration process begins by extracting a region from the reference orthomosaic $\mathbf{R}_r$ and a corresponding region of similar geographical location from the target orthomosaic $\mathbf{R}_t$ (line 1 of Algorithm 1). These region locations could be determined by the user during preprocessing through visual inspection of the orthomosaics. Using only one region per image focuses the registration process on informative and stable areas, reducing computational complexity and improving robustness against noise and irrelevant background.

2.2 Level 1: coarse scale and rotation (OHSI-KAZE)

Algorithm 2 Optimized HSI-KAZE (OHSI-KAZE)

Input: Reference region $\mathbf{R}_\mathrm{r}$, target region $\mathbf{R}_\mathrm{t}$
Output: Scale factor ρ, rotation angle θ, and shift (x, y)

$(\mathbf{P}_\mathrm{r},\mathbf{P}_\mathrm{t}) \leftarrow$ Preprocessing $(\mathbf{R}_\mathrm{r},\mathbf{R}_\mathrm{t})$ ▷ GPU
for each band b in $(\mathbf{P}_\mathrm{r},\mathbf{P}_\mathrm{t})$ **do** ▷ OpenMP + multi-GPU
  $(\mathbf{K}_b^r, \mathbf{K}_b^t) \leftarrow$ Detect keypoints in band b in the preprocessed regions $(\mathbf{P}_b^r, \mathbf{P}_b^t)$ ▷ GPU
    $<$ Nonlinear scale-space construction $>$
    $<$ Feature detection and extraction $>$
  $(\mathbf{D}_b^r, \mathbf{D}_b^t) \leftarrow$ Compute M-SURF and spectral descriptors for $(\mathbf{K}_b^r, \mathbf{K}_b^t)$ ▷ GPU
    $<$ Dominant orientation $>$
    $<$ M-SURF descriptor computation $>$
  $\mathbf{M}_b \leftarrow$ Match keypoints $(\mathbf{K}_b^r, \mathbf{K}_b^t)$ using their descriptors $(\mathbf{D}_b^r, \mathbf{D}_b^t)$ ▷ GPU
    $<$ Euclidean distance computation $>$
    $<$ 2-NN matching $>$
end for
$\mathbf{M} \leftarrow$ Combine the matched keypoints $\{\mathbf{M}_0, \mathbf{M}_1, \dots\}$ ▷ CPU

Optimized registration stage:
$n \leftarrow 0$
for $i \leftarrow 1$, $|\mathbf{M}|$ **do** ▷ OpenMP
  for $j \leftarrow i + 1$, $|\mathbf{M}|$ **do** ▷ OpenMP
    Calculate scale factor ρ_n, rotation angle θ_n, and translation (x_n, y_n) for the pairs $M[i]$ and $M[j]$
    $n \leftarrow n + 1$
  end for
end for
Compute the histogram of the set of n angles ▷ CPU
Find the bin with maximum frequency (modal bin) of the histogram ▷ CPU

Rotation angle parameter:
Sort the elements of the modal bin by their angle value ▷ OpenMP
$\theta \leftarrow$ Rotation parameter from the median of the sorted elements ▷ CPU

Scale and shift parameters:
Sort the elements of the modal bin by their scale value ▷ OpenMP
$(\rho, x, y) \leftarrow$ Scale and shift parameters from the median of the sorted elements ▷ CPU
return (ρ, θ, x, y)

In the first registration level, an optimized version of the HSI-KAZE method, referred to as OHSI-KAZE, is used to estimate a coarse transformation, specifically

focusing on scale (ρ_1) and rotation (θ_1) parameters (line 2 in Algorithm 1). Because of the significant differences in scale and rotation between the orthomosaics, which may result from differing sensor resolutions, flight altitudes, or acquisition conditions, no shift correction is performed at this level. The pseudocode of the proposed OHSI-KAZE method is provided in Algorithm 2. Each line is annotated with a tag (GPU, CPU, OpenMP, or OpenMP + multi-GPU) to denote its specific execution model. The lines executed on the GPU involve several kernels or calls to the official NVIDIA CUDA Toolkit libraries. The most relevant groups of kernels are indicated between <> symbols.

The main stages of the HSI-KAZE method, along with the optimizations introduced in the proposed OHSI-KAZE, are described as follows:

1. **Preprocessing**. Both input regions $\mathbf{R}_\mathrm{r}$ and $\mathbf{R}_\mathrm{t}$ are padded on the GPU to dimensions that are divisible by the number of octaves required for the scale-space construction. The padded images are then upsampled using cubic interpolation via NPP (line 1 of Algorithm 2) to minimize aliasing artifacts and enable the extraction of more keypoints.
2. **Keypoint Detection**. Keypoints are detected within each spectral band of the preprocessed regions $\mathbf{P}_\mathrm{r}$ and $\mathbf{P}_\mathrm{t}$ (line 3). A nonlinear scale-space is constructed through an anisotropic diffusion pyramid that preserves edges and fine details while suppressing noise. At each pyramid level, the GPU computes the multiscale spatial derivatives and the Hessian determinant, from which keypoint candidates are identified as local extrema. The position and scale are then refined at sub-pixel accuracy. Both the detection and refinement kernels utilize CUDA atomic operations and shared memory to efficiently manage concurrent writes to the array storing the extracted keypoints.
3. **Keypoint Description**. For each detected keypoint, a descriptor is calculated that combines spatial and spectral information. The spectral component is derived from the pixel's spectral signature at the keypoint location. For the spatial component, a dominant orientation is first assigned, followed by the computation of a 64-dimensional M-SURF descriptor. These two processes are implemented as separate GPU kernels, where each thread processes a single keypoint in parallel (line 4).
4. **Keypoint Matching**. Keypoint matching is performed by evaluating both the Euclidean distance of the M-SURF descriptors and the cosine similarity of the spectral components (line 5). Two keypoints are considered a valid match only if they satisfy a twofold condition. First, their M-SURF Euclidean distance must fall below a predefined fraction of the distance to the second-nearest neighbor. Second, their spectral cosine similarity must exceed a specified threshold. Both of these thresholds act as tunable hyperparameters. To ensure computational efficiency, the calculation of pairwise Euclidean distances between all keypoints in $\mathbf{K}_b^r$ and $\mathbf{K}_b^t$ is formulated as a matrix operation and accelerated via cuBLAS. The final neighbor selection, which extracts the two closest matches for each keypoint in $\mathbf{K}_b^r$, is performed using custom CUDA kernels that implement a modified insertion sort. Finally, the matched keypoints extracted from the different bands are aggregated in $\mathbf{M}$ (line 7).

5. **Optimized Registration**. The original HSI-KAZE focuses on registering low-resolution satellite images with large scale differences [18]. For this reason, the final selection of the registration parameters was based exclusively on the scale factor. In contrast, OHSI-KAZE improves accuracy by separating rotation estimation from scale and shift estimation into two separate stages. The process is as follows:
For each combination of two matched pairs (4 keypoints), a scale factor ρ_n, rotation angle θ_n, and translation (x_n, y_n) are calculated (lines 8–14). This doubly nested for loop was parallelized using OpenMP. To compute the final registration parameters, a histogram is created using the angle values with a bin size of 0.5° and 0.25° overlap between bins (5° and 2.5°, respectively, in the original HSI-KAZE) (line 15). To further refine the rotation calculation, the elements of the bin with the highest frequency (modal bin) that is computed in line 16, are sorted by their rotation angle, and the rotation parameter θ is recovered from the median (lines 17–18). Then, the elements of this bin are sorted again based on their scale factor, as in the original HSI-KAZE (line 19). To maximize computational efficiency, both of these sort operations are performed using a parallel quicksort algorithm implemented in OpenMP. The scale ρ and shift (x, y) parameters are extracted from the median element (line 20). This optimized process yields more robust and accurate estimates of the scale and rotation parameters.

Finally, the obtained scale factor ρ_1 and rotation angle θ_1 are applied to the entire target orthomosaic $\mathbf{O}_\mathrm{t}$ on the GPU using the NPP library. This step performs the geometric transformation using bicubic interpolation across all spectral bands, producing an initial corrected version denoted as $\mathbf{O}_\mathrm{t}^1$ (line 3 of Algorithm 1).

2.3 Level 2: coarse shift (OHSI-KAZE)

From the registered orthomosaic $\mathbf{O}_\mathrm{t}^1$, a new region $\mathbf{R}_\mathrm{t}^1$ is extracted using the center of the reference region (line 4 in Algorithm 1). At this point, both the reference region and this new target region share approximately the same geographical center, since the scale and rotation angle were nearly corrected at the previous level. The shift parameters can now be reliably computed.

For the same reason, the newly extracted region contains significantly more overlapping information with the corresponding region in the reference image. This increase in mutual information allows the extraction of a larger number of keypoints, which, combined with the optimizations introduced by OHSI-KAZE, leads to improved accuracy in refining the registration parameters.

Therefore, in this level, OHSI-KAZE registers $\mathbf{R}_\mathrm{t}^1$ with respect $\mathbf{R}_\mathrm{r}$, obtaining a refined scale factor and rotation angle, along with a first approximation of the shift $(\rho_2, \theta_2, x_2, y_2)$ (line 5 in Algorithm 1). The three parameters are applied to $\mathbf{O}_\mathrm{t}^1$ to obtain a more accurate registered orthomosaic $\mathbf{O}_\mathrm{t}^2$ (line 6 in Algorithm 1).

2.4 Level 3: fine adjustment (HYFM)

A new region $\mathbf{R}_{\mathrm{t}}^2$ is extracted from the orthomosaic $\mathbf{O}_{\mathrm{t}}^2$ obtained in the previous level (line 7 in Algorithm 1). This final registration level aims to achieve higher accuracy by addressing the small misalignments that remain after the previous levels, which are mainly due to the lack of fine-grained adjustment of the shift parameters. In this case, the area-based HYFM method [19] is used to register the region $\mathbf{R}_{\mathrm{t}}^2$ with respect to $\mathbf{R}_{\mathrm{r}}$ (line 8), as described in detail below.

Algorithm 3 HYFM

Input: Reference region $\mathbf{R}_{\mathrm{r}}$, target region $\mathbf{R}_{\mathrm{t}}$
Output: Scale factor ρ, rotation angle θ, and shift (x, y)

1: $(\mathbf{PC}_r, \mathbf{PC}_t) \leftarrow$ Preprocessing $(\mathbf{R}_{\mathrm{r}}, \mathbf{R}_{\mathrm{t}})$ ▷ GPU
< Blackman >
< PCA >
2: **for** each **PC pair** $(\mathbf{PC}_i^r, \mathbf{PC}_i^t)$ from $(\mathbf{PC}_r, \mathbf{PC}_t)$ **do**
3: $(\mathbf{G}_{\mathrm{MLFFT}}{}_i^r, \mathbf{G}_{\mathrm{MLFFT}}{}_i^t) \leftarrow$ 4-level MLFFT grid $(\mathbf{PC}_i^r, \mathbf{PC}_i^t)$ ▷ GPU
< High-pass filtering >
< Fractional Fourier Transform (FRFT) >
4: $(\mathbf{G}_{\mathrm{LP}}{}_i^r, \mathbf{G}_{\mathrm{LP}}{}_i^t) \leftarrow$ Log-polar grid for each $(\mathbf{G}_{\mathrm{MLFFT}}{}_i^r, \mathbf{G}_{\mathrm{MLFFT}}{}_i^t)$ ▷ GPU
< Log-polar interpolation >
5: $\mathbf{C}_{\mathrm{LP}i} \leftarrow$ Phase correlation for each $(\mathbf{G}_{\mathrm{LP}}{}_i^r, \mathbf{G}_{\mathrm{LP}}{}_i^t)$ ▷ GPU
< FFT >
< Correlation >
< Inverse FFT >
6: **end for**
7: $\mathbf{M}_{\mathrm{LP}} \leftarrow$ Aggregate $\mathbf{C}_{\mathrm{LP}i}$ maps from each PC pair $(\mathbf{PC}_i^r, \mathbf{PC}_i^t)$ ▷ GPU
8: $(\rho, \theta, x, y) \leftarrow$ Process peaks in $\mathbf{M}_{\mathrm{LP}}$ and determine transformation ▷ GPU
< Select the highest peak to obtain scale and rotation >
< Correct scale and rotation in $\mathbf{PC}_1^t$ >
< Phase correlation >
< Select the highest peak to obtain shift parameters >
9: **return** (ρ, θ, x, y)

The pseudocode for the GPU implementation of the HYFM method is outlined in Algorithm 3. As previously established, the most relevant group of kernels is denoted by <>. The algorithm begins with a preprocessing stage (line 1 of Algorithm 3) that incorporates a Blackman window and Principal Component Analysis (PCA). First, the Blackman window is applied to both input regions to remove high frequencies that degrade registration accuracy. Next, PCA is performed to condense the variance across all spectral bands into the first principal components (PCs). PCA is implemented using cuBLAS and cuSOLVER functions to compute the covariance matrix and perform the

eigenvalue decomposition. The resulting output consists of pairs ($\mathbf{PC}_r$, $\mathbf{PC}_t$) that retain the most discriminative spectral information.

Following these steps, HYFM iteratively processes the first N_B corresponding pairs of PCs from the reference and target regions (line 2). For each PC, a 4-level multilayer fractional Fourier transform (MLFFT) grid $\mathbf{G}_{\text{MLFFT}}$ is computed (line 3). Prior to this, a high-pass filter is applied to attenuate low-frequency components. Each level of the MLFFT is then computed using the fractional Fourier transform (FRFT). Both operations rely on the cuFFT library alongside dedicated CUDA kernels.

The MLFFT approach uses a multilevel grid structure that enables the subsequent computation of a log-polar grid $\mathbf{G}_{\text{LP}}$ with significantly reduced interpolation errors (line 4) [26]. To accelerate this operation, GPU texture memory is utilized, taking advantage of its ability to perform linear floating-point interpolation automatically during memory reads at no extra computational cost.

Phase correlation is then applied for each pair of log-polar grids $\mathbf{G}_{\text{LP}}{}^r$ and $\mathbf{G}_{\text{LP}}{}^t$ (line 5). This involves several forward and inverse FFTs computed using cuFFT, while the correlation is performed by a dedicated CUDA kernel. After all PC pairs have been processed, the resulting correlation maps $\mathbf{C}_{\text{LP}}$ are aggregated (line 7), producing the combined map $\mathbf{M}_{\text{LP}}$.

Finally, the values of the log-polar correlation map $\mathbf{M}_{\text{LP}}$ are sorted by magnitude using the Thrust library (line 8). The highest peak of this map is processed to recover the scale and rotation parameters. After that, both parameters are applied to the first PC of the target region $\mathbf{PC}_1^t$ using the NPP library. To recover the remaining shift parameters, phase correlation is computed in the Cartesian domain between the reference $\mathbf{PC}_1^r$ and the scaled and rotated $\mathbf{PC}_1^t$. The peaks of the resulting Cartesian correlation map are once again sorted via Thrust to identify the global maximum, yielding the final shift parameters.

The resulting parameters, scale (ρ_3), rotation (θ_3), and shift (x_3, y_3), are subsequently passed back to the main registration pipeline, where they are applied to $\mathbf{O}_{\text{t}}^2$ (line 9 of Algorithm 1), yielding the final registered orthomosaic $\mathbf{O}_{\text{t}}^{\text{reg}}$.

3 Distributed implementation for multi-GPU clusters

Registering high-resolution, multispectral UAV orthomosaics is computationally demanding due to the need to process all spectral bands and the complexity of the registration task. HPC resources are essential for handling the sequence of orthomosaic pairs generated by UAV flights [27]. We propose a distributed version of the multilevel registration method, optimized for multi-node, multi-GPU HPC clusters.

As illustrated in Fig. 3, the proposed multilevel registration method implements a hierarchical parallelism strategy that integrates distributed-memory parallelism (MPI), shared-memory multithreading (OpenMP), and fine-grained GPU parallelism (CUDA). The set of orthomosaic pairs to be registered is partitioned among the available MPI processes so that each compute node operates fully independently, without requiring inter-process communication during the registration. Since all registration regions share the same fixed dimensions, the computational load per pair is approxi-

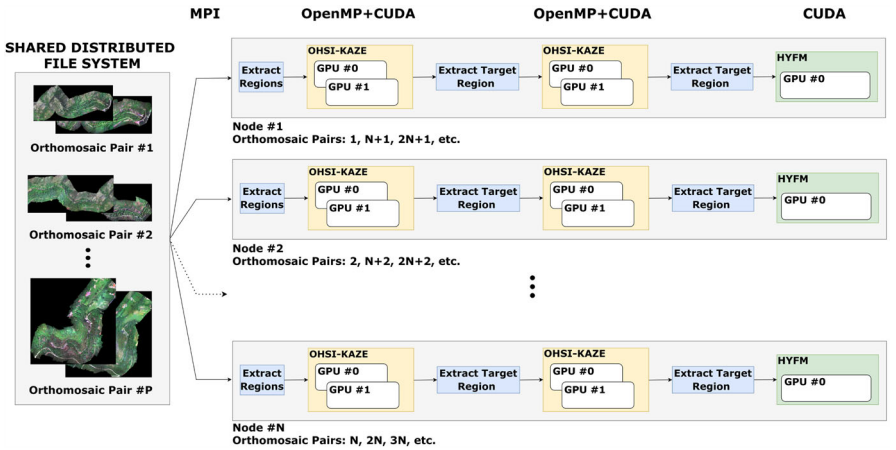

Fig. 3 Multi-GPU multi-node implementation of the multilevel registration method. Pairs of orthomosaics are distributed among the different nodes

mately uniform, making this static partitioning sufficient to achieve balanced workload distribution across nodes.

Each OHSI-KAZE execution shown in Fig. 3 distributes spectral band pairs across the available GPUs (two per node in our setup) using a round-robin strategy. A detailed view of the OHSI-KAZE implementation is provided in Fig. 4. In this configuration, each GPU is managed by a dedicated OpenMP thread responsible for device initialization, including cuBLAS setup and a GPU warm-up phase [28]. The GPUs then independently perform keypoint detection, descriptor computation, and keypoint matching on their assigned band pairs.

During the matching stage, the Euclidean distance between each keypoint in one image and each keypoint in the other must be computed, resulting in a computational cost proportional to the product of the keypoint counts. Since this volume of data exceeds the available GPU memory, the matching is performed iteratively in batches. The batch sizes are aligned with the GPU block dimensions to ensure full thread occupancy and efficient resource utilization. This strategy is particularly important for orthomosaics with a large number of detected keypoints, where multiple iterations are required to complete the matching process within device memory constraints.

Once all bands have been processed, each GPU transfers its matches to the host memory, where they are merged into a single array for the registration stage. This merge constitutes the only synchronization point required throughout the OHSI-KAZE computation. Finally, the registration parameters are computed and applied to the full image.

In contrast to the OHSI-KAZE registration levels of the proposed multilevel method shown in Fig. 3, the HYFM level is executed entirely on a single GPU, as detailed in Fig. 5. Given that HYFM accounts for only a small fraction of the total execution time, the additional complexity of a multi-GPU implementation would not justify the limited performance gains it might provide. Fourier-based registration methods such as HYFM are inherently efficient because they rely on FFT-based cross-correlation,

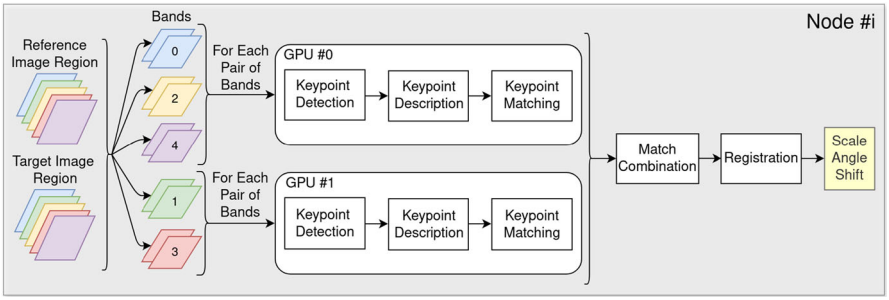

Fig. 4 Multi-GPU implementation of OHSI-KAZE. Detailed view of the processing stages executed within a single node equipped with two GPUs

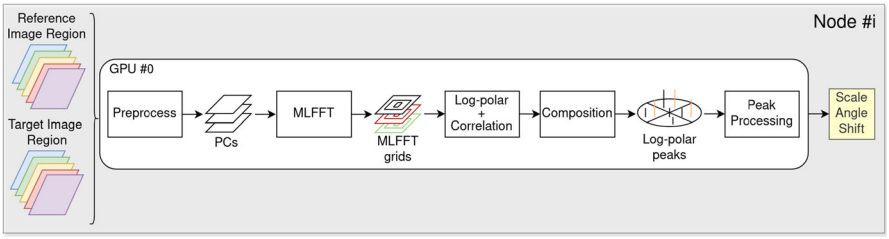

Fig. 5 GPU implementation of the HYFM registration algorithm. A single GPU per node is used to execute all processing stages

an operation that is highly optimized on NVIDIA GPUs via the cuFFT library. Therefore, the implementation focuses on maximizing single-GPU utilization by exploiting texture memory for hardware-accelerated interpolation and minimizing host–device data transfers by reusing memory allocations throughout the computation.

From a practical standpoint, the distributed implementation naturally accommodates both batch and stream processing. As new orthomosaic pairs arrive, they are continuously dispatched to the available nodes and GPUs within the cluster. By doing so, the method exploits maximum parallelism at every level of the hardware architecture, guaranteeing efficient execution for both massive dataset processing and real-time data streams.

4 Results

This section describes the experimental setup and datasets used, followed by a presentation and discussion of the results. The analysis focuses on evaluating registration accuracy and computational performance across CPU, GPU, multi-GPU, and multi-node configurations, including a comprehensive comparison with state-of-the-art methods.

4.1 Experimental setup

All experiments were conducted on the FinisTerrae III supercomputer, hosted at the Center of Supercomputing of Galicia (CESGA). This heterogeneous system comprises 357 nodes, of which 64 are equipped with two NVIDIA A100 GPUs (Ampere architecture), each with 40 GB of HBM2 memory, 40 MB of L2 cache, and 192 KB of shared memory per streaming multiprocessor. In addition, two specialized nodes are available: one equipped with five NVIDIA A100 GPUs and another with eight NVIDIA A100 GPUs. All of these GPU-enabled nodes are powered by two Intel Xeon Ice Lake 8352Y with 32 cores each and a base frequency of 2.20 GHz, optimized for multi-threaded workloads. Inter-node communication is handled by an InfiniBand HDR network, providing low latency and high bandwidth. All nodes share access to a high-throughput Lustre parallel file system for data storage and retrieval.

The input data consist of six pairs of multi-temporal multispectral orthomosaics captured onboard UAVs over riparian environments in Galicia, specifically from different locations along the Oitavén and Xesta rivers (Pontevedra, Spain). Each orthomosaic comprises five multispectral bands stored as 16-bit unsigned integers (uint16), with calibrated reflectance values ranging from 0 to 65,535. These datasets, referred to as Oitavén (2018 and 2024), Eiras (2018), Eiras F1–F3 (2024), and Liñares F1–F2 (2018 and 2024), were captured using UAV-mounted MicaSense sensors as part of an environmental monitoring initiative focused on detecting invasive plant species and assessing the long-term impact of human infrastructure [29].

The 2018 imagery was captured with a MicaSense RedEdge-MX sensor flown at an altitude of 120 m, resulting in a spatial resolution of 8.2 cm/pixel. Each orthomosaic includes five spectral bands: blue (475 nm), green (560 nm), red (668 nm), red edge (717 nm), and near-infrared (842 nm). In contrast, the 2024 images were acquired using a MicaSense Altum sensor, which also operated at a flight altitude of 120 m but delivered higher-resolution imagery (5.2 cm/pixel) and included an additional thermal band at 11μm. This thermal band was excluded from processing to ensure compatibility with the 2018 dataset.

While the 2018 campaign includes a single orthomosaic for the Eiras area, the 2024 dataset consists of three separate orthomosaics. Accordingly, the 2018 orthomosaic of Eiras was spatially divided into three subareas (Fig. 6c, e, g) to correspond with the three distinct UAV flights conducted over the same region in 2024 (Fig. 6d, f, h).

Similarly, the 2024 Liñares campaign comprises two separate flights. Therefore, the 2018 Liñares orthomosaic was divided into two corresponding subareas (Fig. 6i, k) to match the 2024 data (Fig. 6j, l).

In contrast, a single orthomosaic was acquired for the Oitavén River in both 2018 (Fig. 6a) and 2024 (Fig. 6b). This results in a total of six orthomosaic pairs, with the 2018 images serving as the reference and the 2024 images as the targets to be registered.

The multilevel registration method is primarily evaluated using two key metrics: alignment accuracy and execution time. Alignment accuracy is assessed by comparing the estimated transformation parameters (scale, rotation, and shift) against the manually annotated reference values for each orthomosaic pair that are detailed in Table 1, and by computing the root mean square error (RMSE). RMSE quantifies the average

Table 1 Reference transformation parameters (scale, rotation, and x/y shifts) obtained through manual registration for each orthomosaic pair

Orthomosaic pair	Size (px)	Size (MiB)	Scale (×)	Rotation (°)	Tx (px)	Ty (px)
Oitavén	6523 × 6560	816.171	1.960	−0.020	119	−116
Eiras F1	7494 × 2919	417.232	2.233	0.080	141	71
Eiras F2	7811 × 3415	508.777	2.218	0.035	−60	−21
Eiras F3	6725 × 5003	641.731	2.229	0.030	−33	12
Liñares F1	8222 × 5701	937.472	2.061	−0.050	162	−122
Liñares F2	4900 × 3812	373.576	2.196	−0.200	−50	−314

These parameters serve as reference information for evaluating alignment accuracy. All orthomosaics include five spectral bands

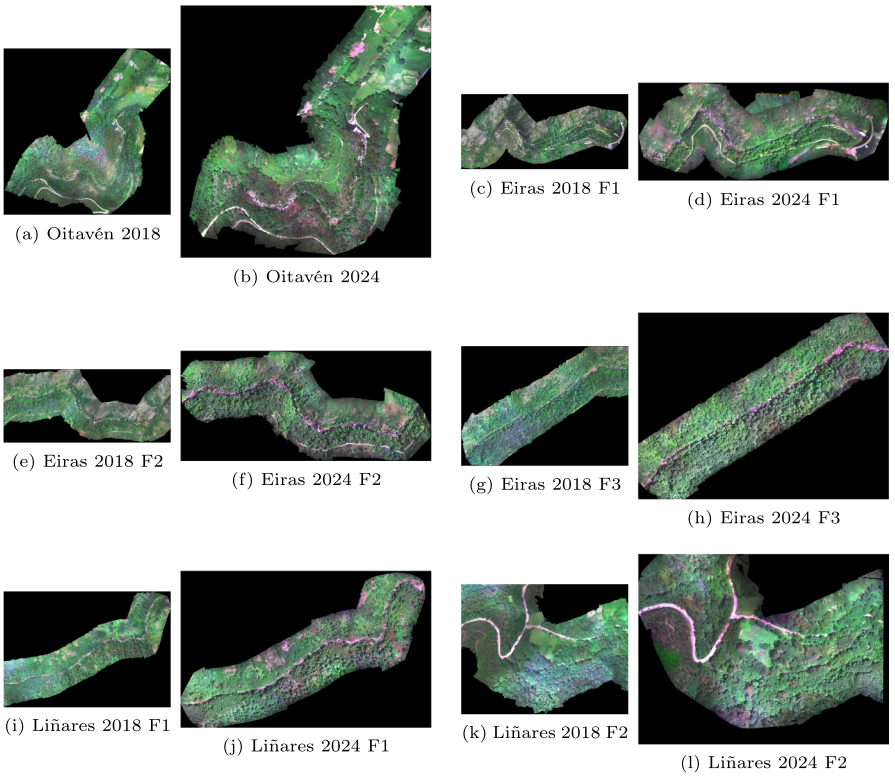

Fig. 6 False-color composites of the pairs of multispectral orthomosaics used in the experiments, illustrating the riparian environments under study

pixel-wise error between the image registered using the proposed multilevel method and the reference, providing a comprehensive measure of geometric alignment quality.

Execution performance is measured as the wall-clock time required to complete the full registration process, with detailed breakdowns reported for each of the three registration levels. To ensure statistical robustness, each experiment is repeated eleven times. The first run is discarded to mitigate caching effects and cold-start variability, and the reported results are the average of the remaining 10 executions. For parallel execution, GPU kernels are launched with 16×16 thread blocks. CPU-based stages are executed with 64 OpenMP threads to fully utilize the physical cores available on each compute node.

The regions extracted at each level of the method are 1000×1000 pixels. The OHSI-KAZE method involves different configurable parameters that influence its behavior across different processing stages. In the feature detection stage, an initial resizing factor of $2 \times$ is applied to the input image, following the configuration proposed in the original method [18], in order to increase the number of detected keypoints. During the feature matching stage, a similarity threshold of 0.65 is used to discard inconsistent matches, and the matching process is executed in batches of 7168 feature pairs (see Sect. 3). This batch size was selected as a multiple of the GPU thread-block

configuration and the warp size (32 threads) to enhance GPU utilization and ensure efficient execution by minimizing idle threads within warps. In the registration stage, the bin size is set to 0.5° and the bin overlap to 0.25°, providing a finer estimate of the rotation parameter.

The performance of the proposed method was evaluated across different implementations. The first is a CPU approach that leverages OpenMP for multithreading with 64 threads, corresponding to all available CPU cores on each node. The second, a single-GPU implementation, uses a single NVIDIA A100 GPU for acceleration. The multi-GPU version employs both A100 GPUs available within a single compute node, operating them concurrently. Finally, a multi-node implementation distributes the workload across multiple GPU-equipped nodes, using MPI for inter-node communication and OpenMP to coordinate the GPU execution.

The code of the proposed method and the datasets used in the experiments are available at https://gitlab.citius.gal/HPC4RS/distributed-multigpu-uav-reg.

4.2 Registration accuracy

The accuracy of the proposed method was evaluated using six datasets. To assess the accuracy of each registration parameter (scale, rotation, and shifts in x and y), the computed transformations were compared against the manually annotated reference data provided in Table 1.

Table 2 presents the differences between the estimated transformation parameters and the corresponding reference values, reported after each of the three levels of the multilevel registration method. Additionally, the root-mean-square error (RMSE) is included to evaluate the overall geometric alignment quality. These accuracy results are independent of the parallel implementation used.

The initial levels of OHSI-KAZE, which combine spatial feature detection with spectral analysis, provide robust and accurate registration, even in scenarios with significant scale, rotation, and geometric variations. These capabilities enable the method to estimate scale and rotation parameters with high precision from the early stages. On average, scale errors are reduced to zero and rotation errors to as little as 0.002 degrees by the final level.

The final HYFM level significantly enhances the accuracy of the shift (translation) parameters, consistently reducing displacement errors to below one pixel. For instance, in the Oitavén dataset, shift errors decrease from 9.240 and 5.922 pixels after the second OHSI-KAZE level to 0.331 and 0.013 pixels, respectively, after applying HYFM. Comparable improvements are observed across all other datasets. The average overall shift error at the final level is 0.284 pixels, confirming the effectiveness of the full refinement process. The RMSE values obtained at the final level are 1.571 pixels on average, and are also shown in Table 2, further supporting the high accuracy of the method, reflecting minimal residual geometric misalignment. The accuracy of the method is evident in the checkerboards of Fig. 7, which show details from different orthomosaics by overlapping the reference image with the registered target image. Overall, the multilevel registration strategy demonstrates strong accuracy across all evaluated metrics.

Table 2 Scale, rotation, and shifts (translation) obtained for each orthomosaic and registration level. For each case, the table includes the error calculated as the difference between the estimated and the reference transformation values, along with the RMSE

Orthomosaic	Level	Scale (×)	Error (×)	Rot (°)	Error (°)	Tx (px)	Error (px)	Ty (px)	Error (px)	RMSE
Oitavén	1° - OHSI-KAZE	1.963	0.003	0.000	0.020	–	–	–	–	–
	2° - OHSI-KAZE	1.960	0.000	−0.018	0.002	128.240	9.240	−110.078	5.922	–
	3° - HYFM	–	–	–	–	119.331	0.331	−115.987	0.013	1.789
Eiras F1	1° - OHSI-KAZE	2.230	0.003	0.050	0.030	–	–	–	–	–
	2° - OHSI-KAZE	2.234	0.001	0.080	0.000	137.718	3.282	71.538	0.538	–
	3° - HYFM	–	–	–	–	140.445	0.555	71.356	0.356	1.528
Eiras F2	1° - OHSI-KAZE	2.218	0.000	0.016	0.019	–	–	–	–	–
	2° - OHSI-KAZE	2.219	0.000	0.035	0.000	−60.998	0.998	−21.429	0.429	–
	3° - HYFM	–	–	–	–	−59.998	0.002	−21.338	0.338	1.119
Eiras F3	1° - OHSI-KAZE	2.231	0.002	−0.022	0.008	–	–	–	–	–
	2° - OHSI-KAZE	2.229	0.000	−0.029	0.001	−29.184	3.816	12.820	0.820	–
	3° - HYFM	–	–	–	–	−33.093	0.093	11.820	0.180	0.398
Liñares F1	1° - OHSI-KAZE	2.059	0.002	0.020	0.040	–	–	–	–	–
	2° - OHSI-KAZE	2.061	0.000	−0.014	0.006	159.898	2.102	−120.225	1.775	–
	3° - HYFM	–	–	–	–	161.498	0.502	−122.225	0.225	2.107
Liñares F2	1° - OHSI-KAZE	2.236	0.040	−0.216	0.016	–	–	–	–	–
	2° - OHSI-KAZE	2.195	0.001	−0.202	0.002	−26.125	23.875	−274.489	39.511	–
	3° - HYFM	–	–	–	–	−50.425	0.425	−314.389	0.389	2.483
Average registration error		–	0.000	–	0.002	–	0.318	–	0.250	1.571

(a) Eiras F1 (b) Eiras F2 (c) Oitavén

Fig. 7 Checkerboard view of a detail of the Eiras and Oitavén orthomosaics after the multilevel registration process

4.3 Computational performance

This section presents the execution times and speedup results for the four implementations of the proposed method: CPU with OpenMP, GPU, multi-GPU, and multi-node. Table 3 reports the execution times for the Eiras F2 dataset across these implementations. The execution times are organized into five main categories: I/O, Extract and Transform, Data Movement, OHSI-KAZE, and HYFM. The I/O category includes the time required to read and write orthomosaics between disk and host memory. The Extract and Transform category corresponds to preprocessing and postprocessing operations performed before and after each registration stage. This includes extracting image regions to be registered at each level and applying geometric transformations estimated in the previous stages. For GPU-based implementations, a separate Data Movement category is introduced to account for all transfers between host (CPU) and device (GPU) memory, as well as GPU memory allocations. The OHSI-KAZE category captures the execution times of the first two registration levels. Finally, the HYFM category corresponds to the execution time of the third registration level. For both OHSI-KAZE and HYFM, execution times are further broken down by stage to provide a detailed view of the computational workload and performance across implementations.

As shown in Table 3, I/O and Extract and Transform stages exhibit minimal and insignificant variation across implementations. In GPU-based configurations, the overhead introduced by CPU–GPU data transfers slightly increases total execution time by 0.62 s in the single-GPU implementation and 0.67 s in the multi-GPU setup. The most computationally expensive stages are in OHSI-KAZE, particularly keypoint detection and keypoint matching. These stages benefit significantly from GPU acceleration due to their high operational intensity (number of operations per byte of data) and fully parallelizable computations. As a result, the multi-GPU implementation achieves speedups of 7.35× and 12.93× for detection and matching, respectively, yielding an overall 13.74× speedup for the full OHSI-KAZE pipeline. The HYFM method, which uses FFT-based operations, also shows substantial performance gains with the optimized cuFFT library. The MLFFT stage, in particular, achieves the highest speedup,

Table 3 Execution times (in seconds) for the Eiras F2 dataset for the CPU with OpenMP, GPU, and multi-GPU (two GPUs) implementations

	CPU (s)	GPU (s)	Multi-GPU (s)
I/O	2.18	2.18	1.94
Extract & Transform	0.43	0.58	0.51
Data movement	–	0.62	0.67
Keypoint detection	48.14	3.63	1.76
Keypoint description	3.93	0.96	0.52
Keypoint matching	175.19	22.24	13.55
Registration	0.74	0.74	0.75
OHSI-KAZE	228.00	27.57	16.59
Preproccesing	0.36	0.03	0.03
MLFFT	5.66	0.03	0.03
Logpolar	0.11	0.04	0.04
Correlation	1.35	0.01	0.01
Composition	0.01	0.01	0.01
Peak processing	0.03	0.01	0.01
HYFM	7.51	0.13	0.12
Total	238.12	31.08	19.82
Speedup	–	7.66×	12.00×

Speedups are calculated over the OpenMP implementation in the CPU column

reducing execution time from 5.66 s on the CPU to 0.03 s on the multi-GPU configuration. As a result, the overall HYFM stage achieves a 62.58× speedup on multi-GPU systems, demonstrating the high efficiency of area-based methods on modern GPU architectures. These results demonstrate strong scalability across implementations, with total speedups of 7.66× for the single-GPU and 12.00× for the multi-GPU configuration, relative to the OpenMP CPU baseline.

To analyze the performance at scale, Table 4 compares execution times and speedups for the registration of all six orthomosaic pairs using four configurations: CPU with OpenMP, single-GPU, multi-GPU (dual-GPU on a single node), and multi-node (distributing orthomosaic pairs across two dual-GPU nodes, for a total of four GPUs). The results show effective parallel scalability across the different implementations. The multi-node configuration achieves 3.56× the speedup relative to a single-GPU configuration and 30.07× the speedup relative to the CPU baseline. At the OHSI-KAZE level, execution time is reduced from 155.04 s (single-GPU) to 90.76 s (multi-GPU), and further to 42.53 s in the multi-node implementation. The speedup between single-GPU and multi-GPU configurations remains below 2× due to the uneven distribution of the five spectral bands across two GPUs, which limits parallel efficiency. In contrast, the multi-node configuration distributes the six orthomosaic pairs evenly across two nodes, enabling balanced workload allocation and maximizing GPU utilization. These results highlight the importance of workload distribution strategies in achieving optimal scalability on heterogeneous high-performance computing systems.

Table 4 Execution times (in seconds) for registering the six orthomosaic pairs using the CPU with OpenMP, GPU, multi-GPU, and multi-GPU multi-node implementations

	CPU (s)	GPU (s)	Multi-GPU (s)	Multi-Node (s)
I/O	14.83	13.96	12.88	4.53
Extract & transform	2.10	5.42	5.34	1.38
Data movement	–	3.37	3.33	1.50
Keypoint detection	293.77	20.44	8.99	4.38
Keypoint description	25.65	5.91	3.01	0.87
Keypoint matching	1119.36	127.13	77.00	36.25
Registration	1.45	1.56	1.56	1.03
OHSI-KAZE	1440.23	155.04	90.76	42.53
Preprocessing	1.99	0.18	0.18	0.05
MLFFT	33.91	0.15	0.16	0.05
Logpolar	0.61	0.22	0.22	0.07
Correlation	8.27	0.02	0.02	0.01
Composition	0.01	0.01	0.01	0.01
Peak processing	0.16	0.01	0.01	0.01
HYFM	44.94	0.59	0.59	0.20
Total	1507.82	178.38	112.90	50.14
Speedup	–	8.45×	13.36×	30.07×

Speedups are calculated over the OpenMP implementation

As a summary of results, Table 5 presents the total execution times across the four implementations for each orthomosaic pair. In the multi-node implementation, the registration of the six orthomosaic pairs is distributed across two nodes, as explained before. GPU speedups range from 7.66× (Eiras F2) to 9.61× (Liñares F2), while multi-GPU configurations achieve speedups between 12.00× and 15.51×. These performance differences are primarily due to dataset-specific characteristics, such as image dimensions and the number of detected keypoints, which are in turn influenced by factors like vegetation density, texture complexity, and lighting conditions. The multi-node configuration processes all six orthomosaic pairs concurrently in 50.14 s, achieving an aggregate speed up of 30.07× over the cumulative CPU execution time (1507.83 s) for the same task.

Complementing these observations, Table 6 provides additional information on resource utilization and dataset characteristics. For instance, Liñares F1 stands out as the largest dataset (937.47 MiB and 3882.74 MiB, respectively) and requires the highest peak RAM consumption (8702.96 MiB). However, it is not the dataset with the highest VRAM consumption or execution time. In contrast, Eiras F1 exhibits the largest number of detected keypoints (154,001), resulting in increased computational load and the highest VRAM usage, making it the most demanding dataset in both respects. These results highlight that orthomosaic size alone does not directly define processing time. Instead, the number of extracted keypoints emerges as the dominant factor, as it correlates strongly with GPU memory consumption and GPU workload.

Table 5 Performance comparison across multiple orthomosaics

Images	CPU	GPU	GPU speedup	Multi-GPU	Multi-GPU speedup	Multi-node	Multi-node speedup
Oitavén	264.99	34.38	7.71×	17.97	14.74×	50.14	30.07×
Eiras F1	242.77	26.37	9.20×	19.93	12.18×		
Eiras F2	238.12	31.08	7.66×	19.81	12.00×		
Eiras F3	246.23	31.02	7.93×	19.84	12.41×		
Liñares F1	256.58	28.62	8.96×	18.73	13.69×		
Liñares F2	259.14	26.94	9.61×	16.71	15.51×		

Execution time (in seconds) and relative speedup for each configuration. The multi-node columns represent the multi-GPU, multi-node implementation that registers all image pairs. Speedups are calculated over the baseline OpenMP CPU implementation shown in the CPU column

Table 6 Performance metrics for each dataset: size of the dataset, maximum number of detected keypoints, peak RAM and VRAM consumptions, and processing time

Dataset		Size (MiB)	Kpts	Peak RAM (MiB)	Peak VRAM (MiB)	Time (s)
Oitavén	2018	857.61	149,668	7,455.38	6,067	17.96
	2024	3126.20				
Eiras F1	2018	468.22	154,001	4,847.46	6,301	19.91
	2024	2077.34				
Eiras F2	2018	499.34	150,491	5,771.00	6,025	19.82
	2024	2502.15				
Eiras F3	2018	610.34	152,183	7,319.86	6,135	19.80
	2024	3185.21				
Liñares F1	2018	937.47	149,559	8,702.96	6,075	18.73
	2024	3882.74				
Liñares F2	2018	373.58	152,017	4,939.34	6,139	16.71
	2024	1815.89				

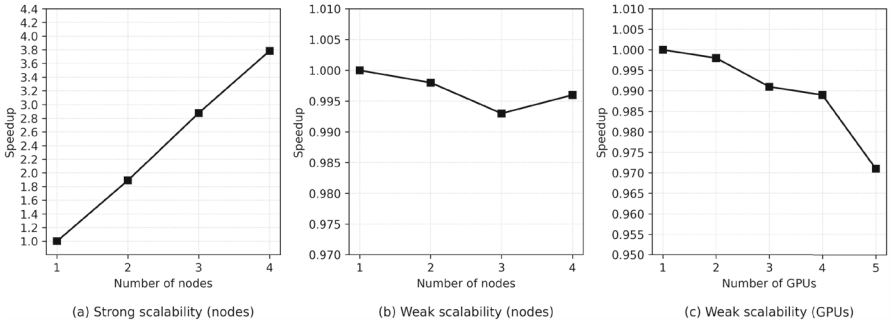

Fig. 8 Scalability analysis of the algorithm. **a** Speedup relative to single-node execution for 12 orthomosaic pairs as the number of computing nodes increases. **b** Speedup relative to a single node when both the number of computing nodes and the number of images increase proportionally. **c** Speedup relative to a single GPU for the Oitavén dataset, where the number of spectral bands scales with the number of GPUs

Overall, the results presented in this section demonstrate the effectiveness of an implementation that exploits multiple levels of parallelism available in HPC clusters, including GPU-level parallelism, multiple GPUs per node, and distributed execution across multiple nodes.

4.3.1 Scalability analysis

To further analyze the scalability of the algorithm, Fig. 8a illustrates its strong scalability, which measures performance as the number of computing nodes increases while keeping the problem size constant, in this case, 12 orthomosaic pairs. The results show the efficiency of the parallel implementation, with the observed speedups confirming strong scalability and indicating low parallel overhead across the tested configurations.

Weak scalability was also analyzed at both the node and GPU levels. In the first case, scalability across nodes was evaluated by increasing the number of images proportionally to the number of nodes. The results, shown in Fig. 8b, indicate that the implementation achieves near-ideal weak scalability, with speedup values close to 1 across different node counts. Minor variations are observed, primarily due to differences in I/O performance between executions.

Figure 8c presents the results of weak scalability when increasing the number of GPUs within a single computing node. This experiment evaluates band-level parallelization across the GPUs available within a single node. A single orthomosaic pair (Oitavén) was used, with its spectral bands replicated as many times as the number of GPUs used for each experiment. As shown in the figure, a slight decline in speedup is observed as the number of GPUs increases. This reduction is mainly attributed to the Extract and Transform stages, which are executed on the CPU using OpenMP, and therefore do not scale as efficiently as the GPU-accelerated stages. Despite this, the implementation maintains a scalability factor of $0.97\times$ with five GPUs and 25-band images, indicating that the overall performance impact is minimal.

4.4 Comparison with other methods

In this section, the proposed method is evaluated against several state-of-the-art approaches in terms of registration accuracy and execution time. Furthermore, a detailed discussion of the results is provided.

As explained in Sect. 2, feature-based methods have been broadly used on image registration thanks to their robustness to noise, rotation, scaling, and illumination changes. Specifically, we select SIFT as a baseline [21], given its widespread adoption as a standard reference in the literature, and the original HSI-KAZE method [18], as it demonstrated good performance on hyperspectral remote sensing images with large scale and rotation differences. Moreover, we compare our approach against HSI-MSER (Hyperspectral Maximally Stable Extremal Regions) [30], another feature-based method specifically designed to align remote sensing images by exploiting spectral information. This method extracts regions through image thresholding at different grey levels, employing MSER [31] for region detection and SIFT for feature description. In previous comparative studies, HSI-MSER outperformed both SIFT and SURF, achieving higher registration accuracy at a lower computational cost [30].

To provide a comprehensive evaluation, we also compare our proposed method against state-of-the-art deep learning models. Following a preliminary assessment of various DL-based techniques, we selected DeepIRDI [11] and GLU-Net [32] as they demonstrated the best registration results on the evaluated datasets. The DeepIRDI method was specifically proposed to register multi-temporal remote sensing images under severe appearance variations. The idea behind this method is similar to classical feature-based methods such as SIFT, but it uses a deep neural network to extract and describe the keypoints (deep features). The method also proposes a strategy for selecting confident keypoint matches. Finally, instead of estimating a homography, it computes a non-rigid transformation by modeling the points as a Gaussian Mixture Model and using Expectation-Maximization to find the optimal parameters.

On the other hand, GLU-Net [32] is a universal network architecture designed to solve pixel-to-pixel correspondence problems, such as image registration. This approach promises high robustness to large viewpoint changes and appearance transformation, as well as high accuracy in estimating small displacements. To compute the dense displacement field, GLU-Net employs a coarse-to-fine strategy, conceptually similar to our proposed multilevel method. It employs a VGG-16 network as its feature extractor backbone and processes the data through four pyramid levels. First, it applies a global correlation layer on downscaled images (L-Net) to capture long-range matches and large displacements. Then, it employs local correlation layers on high-resolution images (H-Net) to iteratively refine this flow and accurately predict small displacements.

The original implementations provided in the references cited above are used, except for SIFT, for which the OpenCV implementation is used [33]. All experiments are conducted on regions of 1000×1000 pixels, since GLU-Net cannot handle the original orthomosaic sizes. As a result, the registration process is restricted exclusively to these cropped areas rather than the full orthomosaics.

Table 7 Scale, rotation, RMSE, and execution times achieved by each method across the six orthomosaic pairs

Orthomosaic	Method	Scale (×)	Rot (°)	RMSE (px)	Time (s)
Oitavén	SIFT	1.960	−0.160	0.138	6987.670
	HSI-MSER	1.869	3.369	16.198	3.024
	HSI-KAZE	1.963	0.002	0.625	17.048
	GLU-Net	–	–	27.554	0.356
	DeepIRDI	–	–	63.374	52.560
	Multilevel (proposed)	1.960	−0.018	0.092	17.970
Eiras F1	SIFT	2.230	−0.210	1.114	8809.200
	HSI-MSER	2.231	−172.931	365.130	3.870
	HSI-KAZE	2.237	0.069	0.634	17.970
	GLU-Net	–	–	232.590	0.525
	DeepIRDI	–	–	195.228	55.780
	Multilevel (proposed)	2.234	0.080	0.316	19.930
Eiras F2	SIFT	2.210	−0.640	2.344	9769.740
	HSI-MSER	1.887	4.693	36.130	3.059
	HSI-KAZE	2.218	−0.029	0.503	15.512
	GLU-Net	–	–	352.184	0.403
	DeepIRDI	–	–	437.849	53.270

Table 7 continued

Orthomosaic	Method	Scale (×)	Rot (°)	RMSE (px)	Time (s)
	Multilevel (proposed)	2.219	0.035	0.320	19.810
Eiras F3	SIFT	2.260	−0.770	3.487	10358.670
	HSI-MSER	2.360	−36.637	112.263	2.840
	HSI-KAZE	2.231	0.139	0.826	15.775
	GLU-Net	–	–	184.901	0.410
	DeepIRDI	–	–	170.713	53.780
	Multilevel (proposed)	2.229	−0.029	0.063	19.840
Liñares F1	SIFT	2.030	0.670	3.970	10457.900
	HSI-MSER	2.142	0.101	7.442	3.135
	HSI-KAZE	2.059	0.020	0.597	16.227
	GLU-Net	–	–	580.900	0.309
	DeepIRDI	–	–	481.126	53.830
	Multilevel (proposed)	2.061	−0.014	0.389	18.730
Liñares F2	SIFT	1.290	−40.590	227.960	6740.980
	HSI-MSER	2.405	−63.981	188.391	3.050
	HSI-KAZE	2.223	−0.173	2.335	17.417
	GLU-Net	–	–	211.403	0.300
	DeepIRDI	–	–	237.901	54.080
	Multilevel (proposed)	2.195	−0.202	0.322	16.710

Scale and rotation parameters are not provided for the DL methods (GLU-Net and DeepIRDI), as they do not compute a homography

Table 7 presents the estimated scale, rotation, registration RMSE, and execution time obtained by each method for all orthomosaic pairs. Because the DL methods compute a dense displacement field rather than a global homography, their scale and rotation parameters are not reported. When comparing the estimated scales and rotation angles against the reference values in Table 1, it can be observed that the proposed registration method achieves the closest results. The feature-based methods generally estimate both parameters accurately, except for SIFT and HSI-MSER, which fail on the Liñares F2 pair, and HSI-MSER, which also fails on the Eiras F2 pair. Moreover, HSI-MSER is highly unreliable for rotation estimation, as it yields incorrect values in 4 of the 6 orthomosaics.

RMSE provides a common metric to compare all the methods. As shown in Table 7, the proposed method achieves the lowest errors, demonstrating the high precision enabled by its multilevel approach. HSI-KAZE is the second-best method, which justifies its selection for our coarse registration levels. SIFT achieves acceptable results except for Liñares F2, where it fails to align correctly. HSI-MSER only successfully registers Liñares F1, but with an RMSE of 7.442 pixels versus 0.389 pixels for the proposed method. Finally, the DL-based approaches obtain significantly higher RMSE values, indicating their inability to correctly register these image pairs.

Despite their promising capabilities, DL-based solutions face significant challenges when applied to real-world images captured by different sensors with varying resolutions. Their primary limitation is the need for large training datasets. The lack of labeled big datasets for image registration forces the researchers to either generate artificial pairs by applying small transformations (e.g., rotations of $[-45°, 45°]$, scale variations of $[0.8, 1.2]$, and minor shifts of $[-20, 20]$ pixels) [13–15], or to restrict their experiments to very small datasets comprising highly similar image pairs [11]. Consequently, models trained on such synthetically augmented data struggle to generalize and to handle images with real multitemporal and scale differences.

Moreover, DL architectures for image registration commonly rely on analyzing fixed-size image patches, which are particularly vulnerable to large-scale variations. This limitation affects both optical flow-based networks, such as GLU-Net, which estimate dense displacement fields, and deep-feature-based methods, such as DeepIRDI, which extract deep descriptors by imposing a fixed spatial grid over resized images. If the scale difference is significant, the effective receptive fields of the patches no longer correspond to the same physical area. As a result, the network attempts to match semantic features across different geographic locations, which results in failure. In summary, current DL-based methods for image registration remain largely limited to scenarios with minor geometric and appearance differences.

Finally, the execution times presented in Table 7 demonstrated the need for parallel and distributed implementations. SIFT exhibits the longest execution times because the OpenCV implementation is restricted to the CPU and lacks GPU acceleration. In contrast, HSI-MSER and HSI-KAZE benefit from highly efficient CUDA-based GPU implementations. Regarding our proposed multilevel method, when registering a single image pair, its acceleration relies on distributing the different spectral bands across the available GPUs within a single node (two GPUs in our experimental setup). This multi-GPU implementation enables the proposed method to achieve sub-pixel accuracy while maintaining execution times similar to those of HSI-KAZE running

on a single GPU, e.g., 17.048 s versus 17.970 s for Oitavén, respectively. Finally, both DL-based approaches leverage a single GPU for inference, but with clearly different time profiles. GLU-Net achieves the fastest processing times, and DeepIRDI proves computationally demanding, recording the second-longest times. However, these times are irrelevant since both networks completely fail to register any of the datasets.

To conclude, the results in Table 7 demonstrate the efficiency of the parallel implementation and the higher accuracy achieved by the proposed method. This exceptional performance is driven by its coarse-to-fine multilevel architecture, which strategically combines the Optimized HSI-KAZE (OHSI-KAZE) to correct severe scale and rotation variations, with HYFM to refine the alignment to sub-pixel accuracy in these experiments.

5 Conclusions

This work presents a scalable, multilevel registration framework for high-resolution, multitemporal, multispectral UAV orthomosaics, targeting the demanding requirements of remote sensing applications such as change detection and environmental monitoring. The method integrates an optimized version of the HSI-KAZE feature-based method, OHSI-KAZE, which focuses on coarse initial registration at the first two levels, followed by an area-based Fourier–Mellin refinement stage (HYFM) at the third level. Experiments were conducted on six pairs of multi-temporal, multispectral orthomosaics acquired in Galician riparian environments. From a registration perspective, each level in the multistage pipeline contributes to progressive refinement, achieving an average overall RMSE of 1.57 pixels. A comparative evaluation confirmed that the proposed framework achieves the highest accuracy among state-of-the-art methods.

From an HPC perspective, the proposed implementation for execution on a heterogeneous cluster exploits different levels of parallelism, using MPI for multi-node distribution, OpenMP for intra-node multi-GPU management, and CUDA for GPU acceleration. Four different implementations, CPU, GPU, multi-GPU, and multi-node, were evaluated. The multi-node implementation achieved the highest performance, with an average speedup of 30× compared to the OpenMP CPU version. Scalability tests confirmed the method's efficient performance as the number of nodes and GPUs increased.

The combination of algorithmic robustness and scalable parallel execution makes this approach a practical solution for large-scale remote sensing workflows. While the current work focuses on UAV imagery, the proposed methodology and its HPC-centric design can be extended to satellite or airborne imagery with minimal adaptation.

Supplementary information

The underlying research materials for this article can be accessed at https://gitlab.citius.gal/HPC4RS/distributed-multigpu-uav-reg

Acknowledgements This work was supported in part by Grant PID2022-141623NB-I00 and TED2021–130367B–I00 funded by MCIN/AEI/10.13039/501100011033 and by "European Union NextGener-

ationEU/PRTR". It was also supported by Xunta de Galicia - Consellería de Educación, Ciencia, Universidades e Formación Profesional [Centro de investigación de Galicia accreditation 2024-2027 ED431G-2023/04 and Reference Competitive Group accreditation, ED431C-2022/16], and by "ERDF/EU".

Author Contributions Author Contributions: Conceptualization, A.O., F.A. and D.B.H.; methodology, A.O, and D.B.H.; experiments D.F.; validation and investigation, A.O., F.A., D.B.H. and D.F.; data curation, A.O., and D.F.; writing, D.F., A.O., D.B.H., and F.A.; visualisation, D.F.; supervision, A.O, D.B.H., and F.A.; All authors have read and agreed to the published version of the manuscript.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Zhang C, Kovacs JM (2012) The application of small unmanned aerial systems for precision agriculture: a review. Precision Agric 13:693–712
2. Asadzadeh S, Oliveira WJ, Souza Filho CR (2022) UAV-based remote sensing for the petroleum industry and environmental monitoring: state-of-the-art and perspectives. J Petrol Sci Eng 208:109633
3. Ecke S, Dempewolf J, Frey J, Schwaller A, Endres E, Klemmt H-J, Tiede D, Seifert T (2022) UAV-based forest health monitoring: a systematic review. Remote Sensing 14(13):3205
4. Lyu X, Li X, Dang D, Dou H, Wang K, Lou A (2022) Unmanned aerial vehicle (UAV) remote sensing in grassland ecosystem monitoring: a systematic review. Remote Sensing 14(5):1096
5. Dai X, Khorram S (1998) The effects of image misregistration on the accuracy of remotely sensed change detection. IEEE Trans Geosci Remote Sens 36(5):1566–1577
6. Zitova B, Flusser J (2003) Image registration methods: a survey. Image Vis Comput 21(11):977–1000
7. Townshend JR, Justice CO, Gurney C, McManus J (1992) The impact of misregistration on change detection. IEEE Trans Geosci Remote Sens 30(5):1054–1060
8. Le Moigne J, Netanyahu NS, Eastman RD (2011) Image registration for remote sensing. Cambridge University Press, Cambridge
9. Leprince S, Barbot S, Ayoub F, Avouac J-P (2007) Automatic and precise orthorectification, coregistration, and subpixel correlation of satellite images, application to ground deformation measurements. IEEE Trans Geosci Remote Sens 45(6):1529–1558
10. Puniach E, Gruszczynski W, Cwikala P, Matwij W (2021) Application of UAV-based orthomosaics for determination of horizontal displacement caused by underground mining. ISPRS J Photogramm Remote Sens 174:282–303
11. Yang Z, Dan T, Yang Y (2018) Multi-temporal remote sensing image registration using deep convolutional features. IEEE Access 6:38544–38555
12. Zeng Y, Ning Z, Liu P, Luo P, Zhang Y, He G (2020) A mosaic method for multi-temporal data registration by using convolutional neural networks for forestry remote sensing applications. Computing 102:795–811

13. Xiao Y, Zhang C, Chen Y, Jiang B, Tang J (2024) ADRNet: affine and deformable registration networks for multimodal remote sensing images. IEEE Trans Geosci Remote Sens 62:1–13
14. Ye Y, Tang T, Zhu B, Yang C, Li B, Hao S (2022) A multiscale framework with unsupervised learning for remote sensing image registration. IEEE Trans Geosci Remote Sens 60:1–15
15. Shi L, Zhao R, Pan B, Zou Z, Shi Z (2023) Unsupervised multimodal remote sensing image registration via domain adaptation. IEEE Trans Geosci Remote Sens 61:1–11
16. Zhang Y, Zhou P, Ren Y, Zou Z (2014) GPU-accelerated large-size VHR images registration via coarse-to-fine matching. Comput Geosci 66:54–65
17. Huo C, Pan C, Huo L, Zhou Z (2011) Multilevel SIFT matching for large-size VHR image registration. IEEE Geosci Remote Sens Lett 9(2):171–175
18. Ordóñez Á, Argüello F, Heras DB (2018) Alignment of hyperspectral images using KAZE features. Remote Sensing 10(5):756
19. Ordóñez Á, Argüello F, Heras DB (2017) Fourier–Mellin registration of two hyperspectral images. Int J Remote Sensing 38(11):3253–3273
20. Foroosh H, Zerubia JB, Berthod M (2002) Extension of phase correlation to subpixel registration. IEEE Trans Image Process 11(3):188–200
21. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
22. Tareen SAK, Saleem Z (2018) A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In: 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE, pp 1–10
23. Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I 9. Springer, pp 404–417
24. Alcantarilla PF, Bartoli A, Davison AJ (2012) KAZE features. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, Oct 7–13, 2012, Proceedings, Part VI 12. Springer, pp 214–227
25. Feng R, Du Q, Li X, Shen H (2019) Robust registration for remote sensing images by combining and localizing feature-and area-based methods. ISPRS J Photogramm Remote Sens 151:15–26
26. Pan W, Qin K, Chen Y (2008) An adaptable-multilayer fractional Fourier transform approach for image registration. IEEE Trans Pattern Anal Mach Intell 31(3):400–414
27. Cavallaro G, Heras DB, Wu Z, Maskey M, Lopez S, Gawron P, Coca M, Datcu M (2022) High-performance and disruptive computing in remote sensing: HDCRS—a new working group of the GRSS earth science informatics technical committee [technical committees]. IEEE Geosci Remote Sensing Mag 10(2):329–345
28. Ordóñez Á, Argüello F, Heras DB, Demir B (2020) GPU-accelerated registration of hyperspectral images using KAZE features. J Supercomput 76(12):9478–9492
29. Argüello F, Heras B, Garea D, Quesada-Barriuso ASP (2021) Watershed monitoring in Galicia from UAV multispectral imagery using advanced texture methods. Remote Sensing 13(14):2687
30. Ordóñez Á, Acción Á, Argüello F, Heras DB (2021) HSI-MSER: hyperspectral image registration algorithm based on MSER and sift. IEEE J Sel Top Appl Earth Observ Remote Sensing 14:12061–12072
31. Matas J, Chum O, Urban M, Pajdla T (2004) Robust wide-baseline stereo from maximally stable extremal regions. Image Vis Comput 22(10):761–767
32. Truong P, Danelljan M, Timofte R (2020) GLU-Net: global-local universal network for dense flow and correspondences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6258–6268
33. Bradski G (2000) The OpenCV Library. Dr Dobb's Journal of Software Tools