PROSPECTIVE COMPARISON OF SURF AND BINARY KEYPOINT DESCRIPTORS FOR FAST HYPERSPECTRAL REMOTE SENSING REGISTRATION

Adrián Rodríguez-Molina[1], *Álvaro Ordóñez*[2], *Dora B. Heras*[3,4], *Francisco Argüello*[4], *José F. López*[1]

[1] Instituto Universitario de Microelectrónica Aplicada, Universidad de Las Palmas de Gran Canaria, Spain
[2] Universidade da Coruña, Integrated Group for Engineering Research, CITIC, Elviña, 15071 A Coruña, Spain
[3] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Spain
[4] Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain

ABSTRACT

Image registration is a crucial process that involves determining the geometric transformation required to align multiple images. It plays a vital role in various remote sensing image processing tasks that involve analyzing changes among images. To enable real-time response, it is essential to have computationally efficient registration algorithms, especially when dealing with large datasets as is the case of hyperspectral images. This article presents a comparative analysis of two descriptors used to characterize local features of images prior to their matching and registration. The objective is to analyze whether the LATCH binary keypoint descriptor, which produces compact descriptors, provides similar results to the gradient-based SURF descriptor in terms of execution time and registration precision. To obtain the best computational performance, multithreaded implementations using OpenMP have been proposed. LATCH has proven to be 7$\times$ faster and as reliable as SURF in terms of accuracy on scale differences of up to 1.2$\times$.

***Index Terms*—** Binary descriptor, hyperspectral, multispectral, registration, OpenMP.

1. INTRODUCTION

Image registration consists of calculating the geometric transformation that allows aligning two or more images. This is a necessary step in any remote sensing image processing, in particular for tracking changes between images of the same area taken at different dates. A computationally efficient registration process is necessary for real-time applications and to handle large datasets. For this purpose, both, adequate registration algorithms and efficient parallel implementations are required.

Feature-based algorithms are the most widely used image registration techniques. They are based on the detection of features such as corners, lines, contours, or edges in the images to be registered to find a correspondence between the images, the reference and the target one if we define the problem between couples of images [1]. The description of these features could be done by two groups of techniques: complex but very efficient vector structures, or low-cost approaches.

The first group comprises gradient-based methods, in which the Scale-Invariant Feature Transform (SIFT) and the Speeded Up Robust Features (SURF) [2] are the most representative. These methods construct descriptors by extracting spatial information around the feature at different scales and after applying different gradients to obtain invariance to transformations and changes. Their descriptors use integers or floating-point numbers for better accuracy.

Methods of the second group, called binary descriptors, construct simpler descriptors that encode most of the information of a feature as binary values using only comparison of intensity values in the images rather than gradients. In this group, Oriented FAST and Rotated BRIEF (ORB), and Learned Arrangements of Three Patch Codes (LATCH) [3] stand out. These techniques are more adequate for low execution time but present more restrictions in the conditions under which they can perform a correct registration.

In this article, efficient implementations of SURF [2], as gradient-based algorithm, and LATCH [3], as low-cost binary keypoint descriptor are compared in terms of execution time and registration efficiency for hyperspectral remote sensing images.

This work was supported in part by grants PID2019–104834GB–I00, PID2020-116417RB-C42 (TALENT-HExPERIA project), TED2021–130367B–I00, and FJC2021-046760-I funded by MCIN/AEI/10.13039/501100011033 and by "European Union NextGenerationEU/PRTR". It was also supported by Xunta de Galicia - Consellería de Cultura, Educación, Formación Profesional e Universidades [grant numbers 2019-2022 ED431G-2019/04 and ED431C-2022/16], by APOGEO project [grant number MAC/1.1.b/226] - Interreg Program (MAC 2014-2020), and by European Regional Development Fund (ERDF). The work of Adrián Rodríguez-Molina was supported by the 2020-2 PhD Training Program for Research Staff of the University of Las Palmas de Gran Canaria.

https://doi.org/10.1109/IGARSS52108.2023.10281734

2. SURF AND LATCH DESCRIPTORS FOR HYPERSPECTRAL REGISTRATION

In this section, we present the integration of SURF and LATCH descriptors into an image registration algorithm to provide a prospective comparative analysis for fast hyperspectral remote sensing registration as well as their OpenMP implementations.

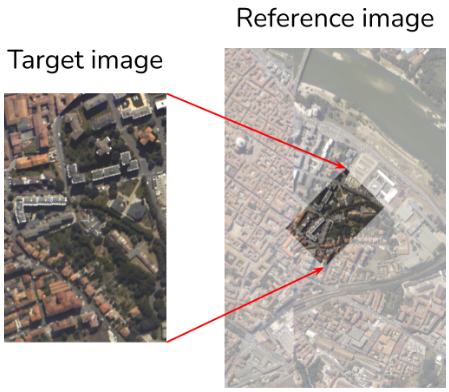


Fig. 1. Example of remote sensing registration. On the left, the target image. On the right, the registration result.

The registration problem addressed in this article consists in obtaining a geometric transformation (scale, angle of rotation, and translation) that allows aligning two hyperspectral images. Figure 1 presents an example. The image to be aligned is called the target image, while the other is called the reference image.

Figure 2 shows the conventional feature-based image registration workflow. It consists of feature detection, feature description, feature matching, and registration.

After detection of features over both images, the descriptor constructs a vector associated with each feature. The descriptor has to be able to produce similar vectors for the same feature in both images even if one of the images is scaled, rotated, shifted or has illumination changes. Then, features detected in the target image are matched with the corresponding ones in the reference image in the matching stage. This process is based on the computation of distances between the descriptors. Each feature of the target image is compared to all features of the reference image. Therefore, the type of data used in the descriptor affects the computation time. Finally, a geometrical transformation is calculated from the matched features.

This article focuses on feature description and feature matching stages (in blue in Figure 2).

2.1. SURF and LATCH descriptors

Figure 3 shows the main differences in the calculation of a keypoint descriptor using SURF and LATCH. The left part of the figure shows the SURF operation for describing a keypoint. The SURF descriptor consists of a vector of 64 floating point numbers that represent different wavelet responses extracted from a region of size $20s \times 20s$ centred on the keypoint and divided into 4×4 subregions, where s is the scale of the keypoint given by the keypoint detector. In short, in each subregion, different Haar wavelet responses in the horizontal and vertical directions, dx and dy, are calculated and summed. In addition, to provide information about the polarity of the intensity changes, the absolute values of these summed responses are also used as part of the descriptor. Thus, each subregion contributes four values to the descriptor: dx, $|dx|$, dy, and $|dy|$. To achieve rotational invariance, the region is previously rotated.

On the right part of Figure 3, the process followed by LATCH is represented. For each keypoint, a window of 48×48 pixels centered on it is selected. LATCH proposes a set of 512 precomputed triplets of small patches inside the window. The figure shows two of these triplets. Each triplet consists of three 7×7-pixel patches. For each triplet, one of the patches is considered the anchor patch. Two distances calculated by the squared Frobenius norm are calculated between the anchor patch and each one of the other two. If the first distance is higher than the second one, a 1 is stored in the descriptor [3]. The calculation is repeated for the different triplets. In our case, 256 triplets were considered.

2.2. Parallel implementations

SURF and especially LATCH methods were designed to have low computational resource requirements. In our case, the algorithms are designed to be executed using low cost hardware requirements, a laptop or a desktop computer. For this reason, we have opted for multithreaded implementations using OpenMP.

The proposed algorithm detects keypoints using SURF as a feature detector in one band of each hyperspectral image. Once the keypoints have been detected, SURF or LATCH are used to describe them. This stage, known as feature description, is performed in parallel using OpenMP. The number of threads supported by the CPU determine the number of parallel keypoints that can be described at the same time.

After the description process outlined in Section 2.1, a matching of the keypoints described in both images is necessary. Each feature of the reference image is compared to all the keypoints of the target image. In the case of SURF, the Euclidean distance is used to decide whether a match is correct, while in the case of LATCH, the Hamming distance is used. The lower the distance, the higher the possibility that these two keypoints correspond to the same area captured in both images. One of the advantages of binary descriptors such as LATCH is that distance computation is reduced to counting the bits that differ between descriptors. In our implementation, it is done by doing a XOR operation between

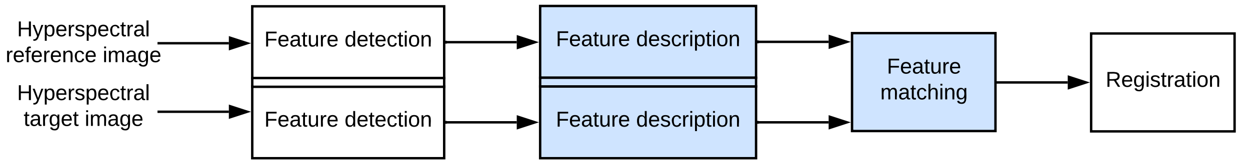
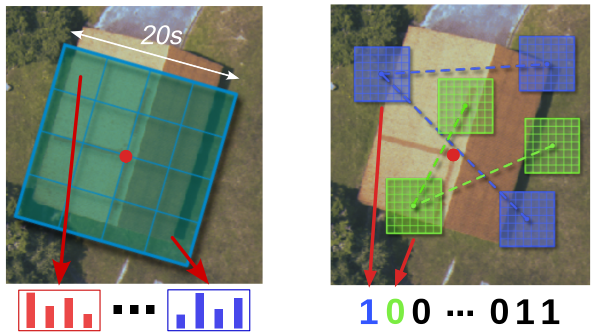


Fig. 2. Feature-based image registration workflow. In blue, the stages this work focuses on: feature description and matching.



(a) SURF (b) LATCH

Fig. 3. Computation of a keypoint descriptor using SURF and LATCH. The keypoint is marked as a red dot.

the descriptors and then counting the bits set as 1. The bit counting is done by using the GCC compiler built-in function *__builtin_popcount()*. This function allows the counting to be performed in a single CPU instruction. Therefore, the comparison between descriptors is performed using two instructions for each 32 bits of the descriptor (8 times in our case, as a 256 bit descriptor is being used). This optimization significantly reduces the number of calculations.

The matching stage is also carried out in parallel. Each thread looks for a match of a different keypoint of the reference image. To avoid critical regions when a match is detected, a local vector to store matches is used for each thread. Once all matches have been obtained, the local vectors are merged into one. Finally, if two or more matches are achieved, the transformation to register the target image with the reference one is computed.

3. RESULTS

The experiments were carried out on a PC with an eight-core Intel i7-10700K CPU at 3.80 GHz and 32 GB of RAM. The code was written in C and C++, and compiled using the gcc and the g++ 10.3.0 versions with the O3 optimization level enabled under Ubuntu 20.04.6 LTS. The results of computation times and speedup provided correspond to the average of ten independent executions for each experiment. The proposed parallel implementations take advantage of the hyper-threading available in the processor by using 16 threads.

Four datasets of remote sensing images obtained by the AVIRIS and ROSIS sensors were considered in the tests. The images contained in the dataset were taken at four different locations: Pavia University (one image of 610×340 pixels and 103 bands), Santa Barbara Line (two images of 1024×769 pixels and 224 bands), Santa Barbara Front (two images of 900×470 pixels and 224 bands), and Jasper Ridge (two images of 1286×588 pixels and 224 bands) [4]. The images of Santa Barbara and Jasper Ridge were also taken on different dates, from different viewpoints, and show changes due to the lapse of time. As a reference for comparison, a random band was selected from each image, in this case, band 28.

Table 1 presents the execution times and speedups achieved in the description and matching stages using SURF or LATCH as feature descriptors to register each pair of images in the dataset. The table presents results for a sequential version (third and fourth columns) as well as a 16-thread OpenMP implementation (fifth and sixth columns). The speedup between implementations (sequential vs. OpenMP) is shown in the last two columns. As can be seen, SURF requires less execution time than LATCH in the description stage because most of the computations required for SURF have been performed in the previous stage of registration, the detection stage, and they are reused in the description stage. LATCH completes the matching stage $10\times$ faster than SURF. One of the advantages of binary descriptors such as LATCH is that the match process is very efficient since it consists of counting the 1s resulting from an XOR operation between descriptors, as explained in Section 2.1. This makes SURF take on average 26.237s for both stages, while LATCH needs 2.907s. Thanks to the exploitation of the 16 CPU threads using OpenMP, the average execution times decrease significantly: 2.587s for SURF and 0.369s for LATCH. This results in speedups of $10.14\times$ for SURF and $7.88\times$ for LATCH.

To analyze the performance of LATCH in terms of registration, besides the scale factor, rotation, and translation changes already present in the images, additional scales and angles are applied. Specifically, the images are scaled down $0.5\times$ and $0.8\times$ and up $1.2\times$, $1.4\times$ and $1.5\times$, and for each scale rotated from 0º to 360º in increments of 5º (72 angles). In total, 432 cases are tested for each image.

Table 2 presents the ratio of successfully registered angles per each scale obtained for the four datasets. In most cases, LATCH can register images which differ on a ± 0.2 scaling factor. However, for larger scales, LATCH is not able to per-

Scene	Stage	Sequential (s)		OpenMP (s)		Speedup	
		SURF	LATCH	SURF	LATCH	SURF	LATCH
Pavia	Description	0.182	0.381	0.025	0.047	7.28×	8.11×
	Matching	3.276	0.177	0.305	0.022	10.74×	8.05×
SB Line	Description	0.689	1.549	0.107	0.196	6.44×	7.90×
	Matching	42.654	2.966	4.171	0.377	10.23×	7.87×
SB Front	Description	0.352	0.782	0.051	0.095	6.90×	8.23×
	Matching	10.965	0.913	1.041	0.111	10.53×	8.23×
Jasper	Description	0.710	1.549	0.107	0.198	6.64×	7.82×
	Matching	46.119	3.294	4.541	0.428	10.16×	7.70×
Average	Desc+Match	26.237	2.907	2.587	0.369	10.14×	7.88×

Table 1. Sequential and 16-thread OpenMP execution times (in seconds) and speedups of the description and matching stages for SURF and LATCH methods.

	Ratio of angles (%)			
Scale	**Jasper**	**Pavia**	**SB Line**	**SB Front**
0.5×	0.0	0.0	0.00	0.0
0.8×	68.1	100.0	100.0	94.4
1.0×	**100.0**	**100.0**	**100.0**	**100.0**
1.2×	97.2	100.0	100.0	100.0
1.4×	0.0	8.3	5.6	0.0
1.5×	0.0	0.0	4.2	0.0

Table 2. Ratio of angles correctly registered for each scale factor using LATCH as descriptor expressed in percentage.

form any correct registration. The LATCH descriptor performs a pixel-by-pixel calculation that compares three 7×7 patches around each keypoint. As the size of the patches is constant, for images with a large scale difference, LATCH will be describing regions corresponding to spatial regions of different sizes. This will result in descriptors of the same keypoint that encode different information and will therefore be difficult to match.

On the other hand, SURF is able to register 100% of the cases [5]. SURF builds a scale-space to achieve scale invariance. As explained in the previous section, the scale at which the keypoint is detected is used to adjust the size of the region around the keypoint to calculate the descriptor.

4. CONCLUSIONS

In this work, a preliminary comparison of LATCH and SURF descriptors of local features for the registration of hyperspectral remote sensing images is presented. The comparison is performed in terms of execution time and precision. Several datasets consisting in pairs of images for the same location captured by the AVIRIS and ROSIS sensors were considered.

The results obtained show that both LATCH and SURF were efficiently implemented in OpenMP, achieving execution time reductions of 7 and 10 times with respect to the sequential implementations, respectively. The less costly algorithm is LATCH, as it is based on simpler binary descriptors. Although both algorithms were able to register the four datasets considered in this paper, the registration capability for different scales and angles is lower in LATCH than in SURF.

Future work should focus on testing extensively the performance of LATCH for images with different spatial and spectral resolutions, on modifying the LATCH descriptor to achieve better scale invariance, and on extending the parallel implementations of the algorithms in order to be capable of efficiently processing time series of hyperspectral images.

5. REFERENCES

[1] Mohamed Ihmeida and Hong Wei, "Image registration techniques and applications: comparative study on remote sensing imagery," in *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE, 2021, pp. 142–148.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] Gil Levi and Tal Hassner, "LATCH: learned arrangements of three patch codes," in *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–9.

[4] Álvaro Ordóñez, Álvaro Acción, Francisco Argüello, and Dora B Heras, "HSI-MSER: Hyperspectral Image Registration Algorithm Based on MSER and SIFT," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 12061–12072, 2021.

[5] Álvaro Ordóñez, Dora B. Heras, and Francisco Argüello, "SURF-based registration for hyperspectral images," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 63–66.