Pattern Analysis and Applications (2025) 28:91
https://doi.org/10.1007/s10044-025-01462-5

ORIGINAL ARTICLE

GGNN: group-guided nearest neighbors for efficient image matching

Ersin Çine[1] · **Yalın Baştanlar**[1] · **Mustafa Özuysal**[1]

Received: 15 December 2024 / Accepted: 24 March 2025 / Published online: 29 April 2025

Abstract

The widely adopted image matching approach remains dependent on exhaustive matching of local features across images. Existing methods aiming to improve efficiency either approximate nearest neighbor (NN) search, compromising accuracy, or apply filtering only after establishing tentative matches, which restricts potential efficiency gains. We challenge the assumption that exhaustive NN search is necessary by proposing a more efficient hierarchical approach that maintains matching accuracy without relying on full-scale NN search. Our key insight is that efficiently identifying sufficiently similar, geometrically meaningful feature matches-rather than the most similar but geometrically random ones-can improve or maintain performance at a lower computational cost. We propose a novel method, Group-Guided Nearest Neighbors (GGNN), which matches groups of features first and then matches individual features only within these matched groups. This hierarchical pipeline reduces the computational complexity of feature matching from $\theta(n^2)$ to $\theta(n\sqrt{n})$, significantly improving efficiency. Experimental results on homography estimation demonstrate that GGNN outperforms standard NN search while achieving performance comparable to state-of-the-art methods. Additionally, we formulate GGNN as a general framework, where conventional NN search is a special case with a single global feature group. This formulation provides a continuum of feature matching methods with varying computational costs, enabling automatic selection based on a given time budget.

Keywords Image matching · Feature matching · Feature aggregation · Hyperdimensional computing · Group testing

1 Introduction

Matching two or more images of the same scene is a fundamental problem in computer vision and serves as a prerequisite for various applications, including image stitching [1–7], 3D reconstruction [8–16], and simultaneous localization and mapping [17–24]. Image matching typically involves extracting local features from all images and matching the most similar features across images. This process is followed by a robust estimation which searches the largest subset of matches that are geometrically consistent.

Many works aimed at enhancing the pipeline has been concentrated on either improving feature extraction or accelerating the geometric verification of feature matches. Nevertheless, the feature matching step has remained relatively unchanged, continuing to depend on either exact or approximate nearest neighbor (NN) search in descriptor space.

Efforts in feature matching tend to be categorized into two groups: either a comprehensive solution that is more accurate yet computationally more expensive than exact NN search with simple filters, or an approximate search that is more efficient but less accurate than exact NN search. A recent and successful exemplar of accurate solutions is AdaLAM [25], which establishes region matches between images and filters feature matches based on these region matches. However, methods like AdaLAM rely on nearest neighbors and add extra steps at the end of the exhaustive search rather than replacing it. This approach limits their efficiency. On the other hand, approximate NN methods such as FLANN-based solutions [26] result in a substantial degradation of performance [27]. To the best of our

✉ Ersin Çine
ersincine@iyte.edu.tr

Yalın Baştanlar
yalinbastanlar@iyte.edu.tr

Mustafa Özuysal
mustafaozuysal@iyte.edu.tr

[1] Department of Computer Engineering, İzmir Institute of Technology, İzmir, Türkiye

knowledge, exact NN search with heuristic filtering remains on the Pareto front for balancing image matching performance and computational efficiency.

We observe that the NN search process for feature matching represents a "leaky abstraction". This is to say that simply identifying the most similar features may not lead to geometrically meaningful results, especially when dealing with repetitive patterns in images. Even when geometric validation successfully identifies and eliminates these highly similar but incorrect feature matches, they continue to cause problems by slowing down the validation process and hindering the use of information from the correct matches of these features.

In our study, we ask: Is it possible to enhance the efficiency of feature matching without losing its accuracy, using a hierarchical approach that does not depend on exhaustive search among feature descriptors? Our thesis is that efficiently identifying sufficiently similar geometrically meaningful feature matches, rather than the most similar but geometrically random ones, can potentially improve or at least maintain matching performance.

We propose a hierarchical pipeline that employs hyperdimensional computing for efficient group testing of feature similarities. Figure 1 illustrates the main idea behind this approach, which first detects, describes and matches feature groups rather than directly matching individual features. Experimental evidence suggests that this group-based approach is not only efficient but also highly effective for image matching.

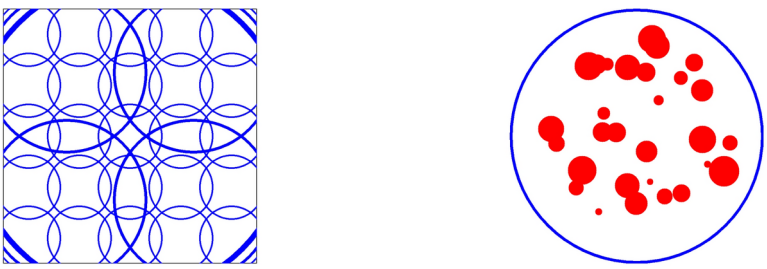
2 Related work

In image matching systems handcrafted feature extractors such as SIFT [28, 29] along with its variants [30–34] and alternative real-valued descriptors [35–41] have been widely adopted due to their robustness against photometric and geometric transformations. With the advent of deep learning, convolutional neural networks have been employed to learn feature descriptors directly from data, which has shown superior performance over handcrafted methods for most tasks. Binary descriptors have also gained popularity due to their computational efficiency and lower memory requirements. Techniques such as BRIEF [42, 43] and ORB [44], which are handcrafted descriptors, along with LATCH [45] and BEBLID [46], which are learned descriptors, provide a faster alternative to real-valued descriptors.

Traditional methods like Nearest Neighbors (NN) and Mutual Nearest Neighbors (MNN) have been foundational for matching sparse features. MNN works by applying the nearest neighbors method bidirectionally and then calculating the intersection, retaining only the mutual matches. This ensures fewer mismatches by confirming that a feature in one image is the nearest neighbor of a feature in the second image and vice versa. However, it often reduces the number of correct matches as well.

To enhance robustness, the Nearest Neighbors with Ratio Test (SNN) [29] was introduced. SNN compares the distance of the closest neighbor to that of the second-closest, filtering out matches where the ratio exceeds a predefined threshold. This ratio test effectively reduces false matches by ensuring that the best match is significantly closer than the second-best match. SNN has been highly influential in the development of newer algorithms and continues to remain relevant today. A more recent approach, SMNN [27], combines the principles of MNN and SNN to further refine the matching process by ensuring mutual consistency and applying a ratio test, thereby reducing false matches.

Beyond descriptor-only methods, some techniques are geometry-aware, meaning they consider the spatial relationships between keypoints. First Geometrically Inconsistent Nearest Neighbors (FGINN) [47] extends SNN by searching for second nearest neighbors only among keypoints that are spatially distant from the first nearest neighbor, producing a superset of SNN. Common Visual Pattern (CVP) discovery [48] formulates feature matching as a clustering



(a) Image with a pyramid of circular regions (b) Region as a group of local features

Fig. 1 Hierarchical approach for feature matching: **a** illustrates the coarse feature matching where the images are divided into regions of varying sizes, resulting in $\sqrt{n}$ regions for n features. Each region is then compactly described by aggregating the group of local features detected within it, as shown in (**b**), instead of using a descriptor extraction method directly. This aggregation helps to discard the vast, uninformative space within the region. Subsequently, groups of features are matched across images, followed by the matching of individual features within the corresponding groups

problem by mapping keypoint correspondences into transformation spaces modeled as matrix Lie groups, which are then refined using nonlinear mean shift clustering to detect repeating patterns across images. Grid-based Motion Statistics (GMS) [49] enhances feature matching by enforcing local motion smoothness constraints, identifying clusters of consistent matches across a grid structure while efficiently rejecting outliers. Locality Preserving Matching (LPM) [50] strengthens feature matching by enforcing local neighborhood consistency through a geometric model, allowing it to handle non-rigid transformations and effectively remove outliers. Geometry Consistency Aware Confidence Evaluation (GCCE) [51] improves feature matching by iteratively refining correspondences through local geometric consistency analysis, employing a shrinking-expanding strategy for robust match pruning and expansion. Finally, Adaptive Locally-Affine Matching (AdaLAM) [25] filters NN matches with local geometric verification around SNN matches using Random Sample Consensus (RANSAC) [52]. All these methods operate by filtering feature matches after establishing putative correspondences, which inherently limits efficiency.

Efficient alternatives to exact NN methods have been extensively explored in the literature. Notably, the Fast Library for Approximate Nearest Neighbors (FLANN) [26] employs multiple randomized kd-trees [53] and hierarchical k-means trees [26] to approximate NN. Although these methods offer improved efficiency, they perform worse than exhaustive nearest neighbor searches in terms of matching accuracy [27].

Additionally, hash-based methods such as LDAHash [54] and CasHash [55] provide efficient indexing of descriptors. While these methods are more efficient, they are typically less accurate than tree-based methods and are often implemented in a feature-specific manner, which is another disadvantage.

Another category of approximate NN algorithms includes graph-based matchers [56], which scale well for searching. Examples include the Navigating Spreading-out Graphs (NSG) [57] and Hierarchical Navigable Small Worlds (HNSW) [58]. These algorithms are utilized for general searches in large vector databases and are not limited to local feature matching across two images.

All these tree-based, hash-based, and graph-based approximate nearest neighbor methods are purely descriptor-based and do not incorporate geometric awareness. Consequently, even under optimal conditions, these algorithms can only perform as well as exact NN methods and not better, for sufficiently large datasets.

The matching algorithm proposed in this work neither filters nor approximates NN. It is designed to be more efficient than exact NN and more accurate than approximate NN.

3 Approach

We propose Group-Guided Nearest Neighbors (GGNN), a hierarchical approach to matching image pairs. Initially, we spatially group keypoints and represent each group with a **single** descriptor vector. This is achieved by describing individual features with vectors that are probably approximately orthogonal to each other and then through aggregation of these vectors by summation as explained below. Then, GGNN matches these groups efficiently across images, and we carry out feature matching and match filtering within these matched groups. By performing feature matching across group matches instead of conducting a global search, we enhance efficiency. Consequently, the proposed method showcases a lower time complexity than NN search. Figure 2 shows an overview of the proposed system.

3.1 Probably approximately orthogonal feature description

The simple summation operation effectively represents a set of vectors when the involved vectors are pairwise orthogonal. The expected value for the angle between two random, zero-centered descriptor vectors in $k > 1$ dimensions is $90°$. However, the variance is significant in relatively lower-dimensional spaces, particularly when the descriptors are not statistically random. Achieving perfect orthogonality among descriptors without compromising other desirable characteristics such as matchability is infeasible. To achieve "probably approximately orthogonal" vectors, we employ higher-dimensional descriptor vectors than usual, as vectors in higher dimensions are more likely to exhibit orthogonality. This approach is inspired by the principles of hyperdimensional computing [59], where vector symbolic architectures enable symbolic computation using very high-dimensional random vectors. These vectors, when subjected to algebraic operations, function akin to distinct symbols.

We explored various publicly available feature extractor algorithms and their combinations through concatenation, focusing on binary descriptors since they are faster to compute and compare, and usually higher-dimensional than the real-valued alternatives. The concatenation of bipolar representations of two binary descriptors, 512-dimensional LATCH [45] and 512-dimensional BEBLID [46], computed on Oriented FAST [44] keypoints, emerged as an effective solution in terms of orthogonality and matchability. To enhance orthogonality, we identified better configurations of these algorithms than their default settings in OpenCV [60]. Additionally, we implemented non-maximum suppression on keypoints based on their response scores to enhance orthogonality further, in response to our observation that

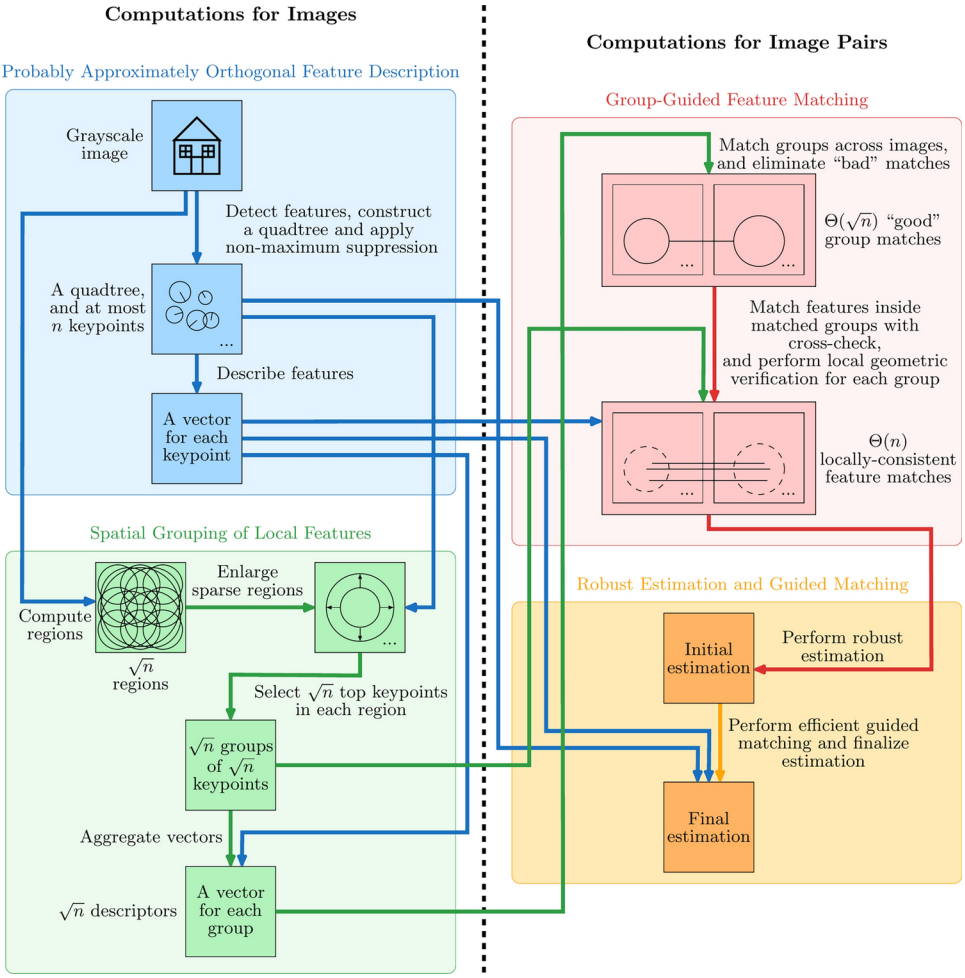


Fig. 2 Comprehensive system overview: the diagram illustrates the computational processes applied to individual images (left) and to pairs of images (right). It is important to note that for a dataset of m images, the number of potential image pairs escalates to $\Theta(m^2)$. This exponential increase underscores the significance of optimizing the efficiency of computations that are performed for each image pair. For optimal visual clarity, viewing in color is recommended

keypoints in close spatial proximity are less likely to yield orthogonal vectors.

3.2 Spatial grouping of local features

The hierarchical approach necessitates the grouping of local features. In our experiments, it was observed that grouping features based on keypoint positions consistently outperforms random grouping or grouping based on descriptor vectors, such as maximizing intra-group pairwise orthogonality.

Various spatial grouping strategies were explored, including the use of top-scale keypoints as regions, clustering of keypoints, and random sampling of large circles on images. Remarkably, the most effective method was also the simplest: employing fixed-size, overlapping circles that are systematically sampled, with their centers forming a grid layout. The total number of these circular regions correlates with the time allocated for solving the matching problem. We suggest using $\sqrt{n}$ regions where n is the number of features. Other hyperparameters, such as circle sizes and distances between them, were empirically determined. These parameters could potentially be further tuned in the future using larger, more diverse datasets or those specific to certain domains. The sampling algorithm was improved by introducing a pyramid-like pattern of variable-size circles, which better accommodates scale differences between images.

To aggregate descriptors, we employed element-wise addition of the bipolar representation of binary descriptors. This operation, referred to as "bundling" in hyperdimensional computing literature, efficiently represents a superposition of orthogonal vectors. There is no need for L_2 normalization of the vectors before aggregating since they all possess the same L_2 norm, with components being either 1 or -1. After aggregation, there is still no need for such normalization because we always add the same number of such vectors, and thus the resulting vectors have the same norm.

3.3 Group-guided feature matching

Once individual features are computed and groups are formed and described, we match these $\sqrt{n}$ groups of features

across images (right side of Fig. 2). We compare the group descriptors using cosine similarity, as our experiments demonstrated that it performs better than other, more complex set-theoretic formulations such as the Jaccard Index. This correspondence search is performed bidirectionally, considering the union of the resulting match sets. We retain only the top 50% of the group matches based on their vector similarities to eliminate low-quality matches, resulting in a maximum of $\sqrt{n}$ group matches.

Next, we conduct feature matching for each of the matched groups across images. To limit the impact of incorrectly matched groups and remove false feature matches, we first filter the nearest neighbors using the mutuality constraint. We then perform local geometric verification by running multiple RANSACs [52] in parallel, similar to the method described in [25]. We aggregate all feature matches obtained from all group matches into a single pool, which contains at most n feature matches, though typically fewer in practice.

This group-guided feature matching concept parallels the principles of group testing [61]. In group testing, individual tests are simultaneously conducted on multiple items. However, our approach involves not only grouping the items (features of the first image) but also the tests (features of the second image). Adopting this two-way grouping strategy enhances the efficiency of the testing process.

3.4 Robust estimation and guided matching

When features are matched and pre-filtered, they often still contain outliers-matches that are geometrically inconsistent with the largest consistent subset. To address this, we follow the standard procedure of performing a robust estimation of the geometric transform, typically using RANSAC [52] or its variants [62–70]. We maintain a high threshold for robust estimation to find a coarse estimation.

Next, we conduct estimation-guided feature matching using all keypoints. In this process, we compare features with other features within a small neighborhood centered around the estimated location of the keypoint. This is done using the ratio test for descriptors as described in [29] and scale-based filtering for keypoints as in [25]. Guided matching is highly efficient as it leverages the previously constructed quick keypoint search structure. We finalize the process with robust estimation over the new feature matches, this time using a small error threshold.

3.5 Computational complexity and generalization

The time complexity of NN search is $\Theta(kn^2)$, where k represents the descriptor dimensionality, which is typically a constant, and n represents the feature count, a variable whose optimal value depends on factors such as image resolution. In our approach, we use $\sqrt{n}$ groups containing $\sqrt{n}$ features. Our algorithm requires $\Theta(kn)$ time for exact group matching. Subsequently, matching $\sqrt{n}$ features to other $\sqrt{n}$ features takes $\Theta(kn)$ time for each the $\sqrt{n}$ group matches. The total complexity for group-guided feature matching thus becomes $\Theta(kn\sqrt{n})$. Since k is normally a constant, this complexity can be simplified to $\Theta(n\sqrt{n})$.

The proposed feature matching method can be formulated as a general framework in which g groups are matched across images and then g times c members are matched against c members of the matched group. This framework requires $g^2 + gc^2$ descriptor comparisons in total. It is sensible to constraint the relationship between these variables to satisfy the equation $c = n/g$. This way g groups of c features will have n features in total. (Note that this does not mean all features are covered as there are overlapping features between groups.) Applying this constraint the number of comparisons becomes $g^2 + n^2/g$. For simulating the conventional NN search, there must be only $g = 1$ group and thus $1 + n^2$ comparisons (or simply n^2 by avoiding the unnecessary group matching) must be performed. Whereas, the proposed method constructs $g = \sqrt{n}$ groups and requires only $n + n\sqrt{n}$ comparisons. Within this framework it is possible to minimize the computational cost beyond the proposed setting: the minimum number of comparisons is $3\sqrt[3]{2}n^2/(2\sqrt[3]{n^2})$, which occurs when there are $g = \sqrt[3]{4}\sqrt[3]{n^2}/2$ groups. Instead of using the proposed setting, one can select an appropriate integer g automatically from the range $[1, \sqrt[3]{4}\sqrt[3]{n^2}/2]$ depending on the time budget. The number of total comparisons decreases monotonically within this range.

4 Experiments

We evaluate the performance of the proposed method on the task of homography estimation using the Oxford classic image matching dataset [71]. This dataset consists of six images for each of the eight scenes, resulting in a total of 48 images. For each scene, there is one reference image, which is paired with each of the other five images in that scene, resulting in 40 image pairs with corresponding ground truth homography matrices. By calculating homography matrices for all possible image pairs within each scene, we extend this to a total of 288 image pairs, which we refer to as Oxford$^+$. The "Bikes" and "Trees" scenes exhibit varying degrees of blur. The "Leuven" scene involves variations in light conditions, while the "UBC" scene is characterized by JPEG compression. The "Graff" and "Wall" scenes depict

changes in viewpoint, and the "Bark" and "Boat" scenes involve zoom and rotation adjustments.

Additionally, we use the Homogr dataset [72], which contains a single image pair for each of the 16 scenes. Due to the small sample size and the similar performance of all methods, we generate synthetically transformed image pairs to allow for differentiation among the methods. Following the approach described in [73], we generate random homographies by perturbing the image corners. This process allows for the creation of any number of image pairs, and for our experiments, we generate 640 image pairs.

Lastly, we utilize the image sequences from the HPatches dataset [74]. This dataset comprises 580 image pairs, with 285 pairs featuring illumination changes and the remaining 295 pairs featuring view changes.

We employ the average corner error (ACE) [73, 75–84], a metric commonly used for assessing the accuracy of geometric transformations, for the evaluation of the obtained transformation. ACE quantifies the average discrepancy between the true and estimated corner positions in the transformed image. Success is declared if the ACE value falls below 1% of the image diagonal's length; otherwise, it is considered a failure [85].

In experiments we use $n = 4096$ features. In practice the number of features can be as high as 8k [25], 10k [41], 12k [40], 15k [27], 20k [86] or 40k [87]. The theoretical speedup of the proposed method increases as the number of features increases.

Table 1 presents the performance of the proposed method on the dataset and compares it with several classical and recent methods. The classical methods include nearest neighbors (NN), mutual nearest neighbors (MNN), and nearest neighbors with ratio test (SNN) [29]. Additionally, we evaluate FGINN [47], a variation of SNN that considers the geometry of keypoints. Recent methods such as AdaLAM [25] and SMNN [27], as implemented in Kornia [88], are also included in the comparison. We also include the preemptive feature matching strategy [11], which matches top-scale features across images to accelerate multiview matching. As a hierarchical matching baseline with a $\Theta(n\sqrt{n})$ time complexity, we include feature matching guided by matches of top-scale features, referred to as Hierarchical Nearest Neighbors (HNN). For linear-time approximate NN methods, we evaluate the widely-used FLANN [26], which is tree-based, and the state-of-the-art HNSW [58], which is graph-based. To ensure a fair comparison, all methods were applied to the same keypoints with identical descriptor vectors, followed by identical post-processing steps, including guided matching. For robust estimation, we used GC-RANSAC [65]. All methods were tuned for optimal performance, except for NN and MNN, which do not require parameter tuning.

The results indicate that GGNN achieves Pareto optimality in the efficiency-accuracy trade-off, meaning that no other method surpasses GGNN in both efficiency and accuracy. In other words, for its level of efficiency and beyond, GGNN demonstrates the highest accuracy across all three datasets. Furthermore, GGNN outperforms both NN and AdaLAM on all three datasets while maintaining greater efficiency.

Typically, methods that approximate the NN search such as FLANN and HNSW are expected to perform worse than exact NN search, with their best-case performance matching that of exact methods. However, this upper limit does not apply to our proposed method. If the group matching step is performed effectively, then matching features within

Table 1 Homography estimation results of GGNN and other methods

		Failure percentage									
Dataset	Image pairs	NN	MNN	SNN (2004)	FGINN (2015)	AdaLAM (2020)	SMNN (2021)	HNN	GGNN (Proposed)	FLANN (2009)	HNSW (2018)
Oxford$^+$ Bikes	36	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Oxford$^+$ Trees	36	2.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	2.8	2.8	**0.0**	**0.0**
Oxford$^+$ Leuven	36	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Oxford$^+$ UBC	36	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Oxford$^+$ Graff	36	27.8	36.1	**25.0**	**25.0**	33.3	30.6	50.0	**25.0**	41.7	33.3
Oxford$^+$ Wall	36	13.9	16.7	13.9	**11.1**	**11.1**	**11.1**	22.2	**11.1**	36.1	16.7
Oxford$^+$ Bark	36	69.4	**63.9**	66.7	69.4	66.7	66.7	72.2	**63.9**	75.0	69.4
Oxford$^+$ Boat	36	55.6	52.8	**44.4**	47.2	47.2	**44.4**	55.6	47.2	52.8	52.8
Oxford$^+$	288	21.2	21.2	**18.8**	19.1	19.8	19.1	25.3	**18.8**	25.7	21.5
Homogr-Random	640	8.1	7.2	6.7	6.7	12.5	6.9	18.0	**6.6**	22.5	8.9
HPatches Illum	285	10.5	8.4	**7.4**	8.8	11.2	8.8	18.9	10.2	18.2	10.5
HPatches View	295	18.0	15.9	15.6	14.9	17.6	**14.6**	33.2	16.3	32.9	19.3
HPatches	580	14.3	12.2	**11.6**	11.9	14.5	11.7	26.2	13.3	25.7	15.0
Time complexity		$\Theta(n^2)$						$\Theta(n\sqrt{n})$		$\Theta(n \log n)$	

Bold values indicate the best (lowest) failure percentage for each dataset

correctly matched groups becomes significantly easier compared to performing a global matching. This structured approach leads to a performance gain over other methods.

Figure 3 presents the intermediate results of the proposed method on a selected image pair, demonstrating how the refining process enhances the rate of correct matches at both the group and feature levels. Figure 4 provides sample results, showcasing both successful matches and various types of failures, including lack of region overlap, group mismatches, and feature mismatches within correctly matched groups.

4.1 Ablation study

Table 2 presents the results of the ablation study, wherein each column represents the proposed method with some components intentionally omitted. The ablation study reveals that performance deteriorates when certain components are removed or modified, highlighting the significance of these components in the overall effectiveness of the proposed method. This suggests that each component contributes positively to the model's performance and that their inclusion is essential for achieving optimal results. Notably, the results indicate that the removal of NMS led to the largest decrease in performance across all three datasets, underscoring its critical importance in maintaining high accuracy.

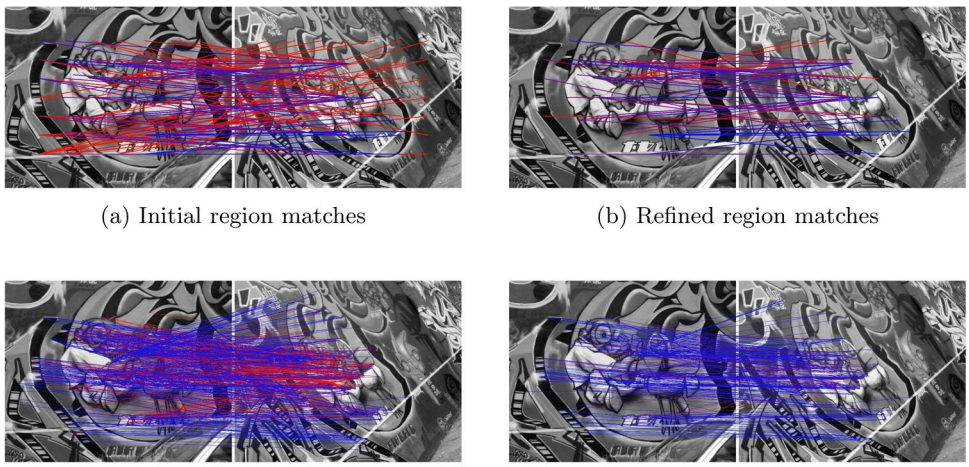
In Sect. 3.5 we mentioned that the number of groups, denoted by g, can actually be different from $\sqrt{n}$. We aim at minimizing both the number of comparisons and the failure percentage. Figure 5 shows the results obtained by varying the number of groups.

For $n = 4096$ features, the number of vector comparisons $g^2 + n^2/g$ is 16, 777, 217 for $g = 1$ group, 266, 240 for $g = 63$ groups, and 123, 855 for $g = 203$ groups. This demonstrates a significant difference between the NN algorithm and the proposed GGNN setting, although the difference in computational efficiency between the proposed GGNN setting and the absolute minimum is less pronounced. We observe that $g = \sqrt{n}$ generally provides a good balance between speed and performance, at least for $n = 4096$ using our features on our datasets. However, this parameter can be optimized for specific features, datasets, applications, and time limits.

4.2 Error analysis

Three primary types of errors can be identified in the matching process. The first type of error arises from a lack of region overlap, which occurs when the detected regions do not sufficiently overlap, making matching impossible. This limitation depends on the region computation algorithm. However, the systematic sampling approach used in our method ensures spatial overlap, while the incorporation of a region pyramid, consisting of regions of varying sizes, mitigates scale differences. The impact of this component is evident in Table 2, where the removal of the pyramid structure results in fixed region sizes, leading to poor overlap in cases involving significant scale variation.

The second type of error involves group mismatches, which occur when regions are incorrectly matched. These mismatches can result from insufficiently discriminative aggregate vectors or from repeating structures within the



(a) Initial region matches (b) Refined region matches

(c) Initial feature matches (d) Refined feature matches

Fig. 3 Intermediate results from GGNN: **a** initial region matches: Groups are matched with the most similar groups from the other image. Successful matches, indicated by blue lines, have a sufficient number of common members. **b** Refined region matches: Low-quality region matches are discarded. **c** Initial feature matches: Members are matched with the most similar members of the matched groups. Blue lines indicate matches with low localization errors. **d** Refined feature matches: Within each group, geometrically inconsistent matches are discarded. A spectrum of colors from blue to red is used to indicate the quality of matches, with blue representing the best matches and red representing the worst

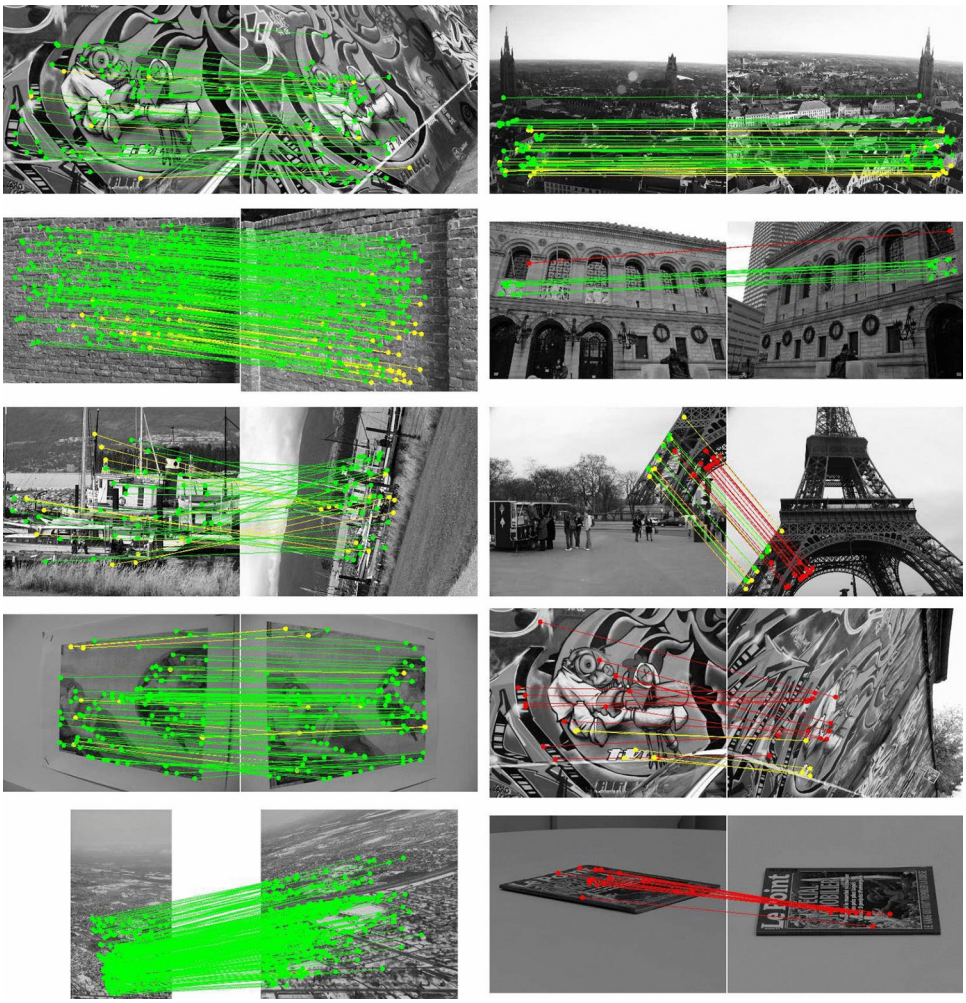


Fig. 4 Sample results from GGNN: more successful outcomes shown on the left and less successful ones on the right. Failures occur due to relatively high reprojection errors in matching (indicated by yellow or red matches, with red being worse), or when the number of matches is not sufficiently high, or when matched features are not spatially well-distributed

Table 2 Failure percentages for ablated components

Non-maximum suppression	✓		✓	✓	✓	✓
Pyramid for regions	✓	✓		✓	✓	✓
Region match elimination	✓	✓	✓		✓	✓
Local geometric verification	✓	✓	✓	✓		✓
Guided matching	✓	✓	✓	✓	✓	
Oxford$^+$ Bikes	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Oxford$^+$ Trees	2.8	2.8	2.8	**0.0**	2.8	2.8
Oxford$^+$ Leuven	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Oxford$^+$ UBC	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Oxford$^+$ Graff	**25.0**	41.7	30.6	33.3	27.8	36.1
Oxford$^+$ Wall	**11.1**	19.4	16.7	16.7	19.4	16.7
Oxford$^+$ Bark	**63.9**	66.7	66.7	69.4	69.4	**63.9**
Oxford$^+$ Boat	**47.2**	52.8	55.6	55.6	55.6	52.8
Oxford$^+$	**18.8**	22.9	21.5	21.9	21.9	21.5
Homogr-Random	**6.6**	12.0	7.2	6.7	8.9	8.9
HPatches Illum	**10.2**	13.0	13.3	**10.2**	13.7	12.3
HPatches View	**16.3**	27.5	17.6	16.6	22.0	18.3
HPatches	**13.3**	20.3	15.5	13.4	17.9	15.3
Time complexity	$\Theta(n\sqrt{n})$					

Bold values indicate the best (lowest) failure percentage for each dataset

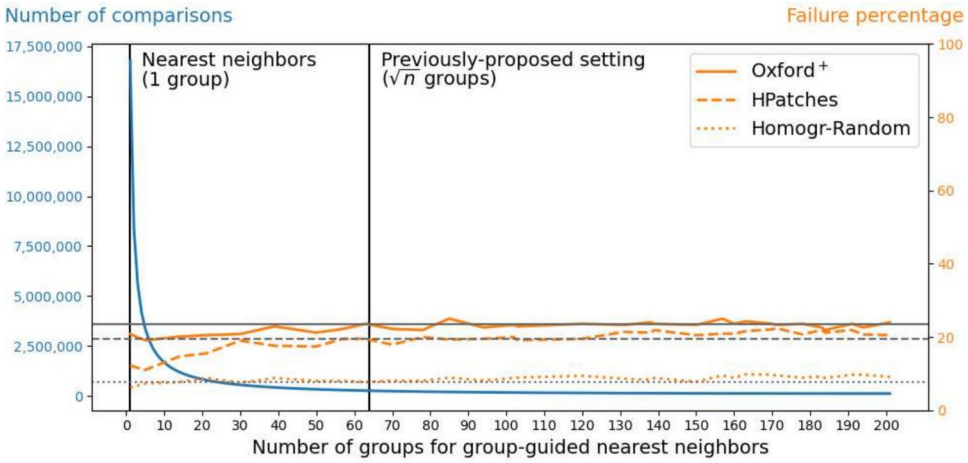


Fig. 5 Impact of the number of groups: the plot visualizes the impact of different numbers of groups g while keeping their cardinality at $c = n/g$ for $n = 4096$. The proposed setting is calculated as $g = \sqrt{n} = 64$. Horizontal lines mark the performance level of the proposed setting

images. The former issue is closely tied to feature extraction and may improve as feature extraction algorithms advance. The latter represents an inherent limitation of the method, as even perfectly described regions can mismatch due to structural repetition. This trade-off prioritizes efficiency; however, in practice, it does not pose a significant problem, as successful matching does not require all regions to be perfectly aligned-only a sufficient number of high-quality region matches. Additionally, the "Region match elimination" step, shown in Table 2, helps filter out low-quality matches, further mitigating this issue.

The third type of error pertains to feature mismatches within correctly matched groups. Even when groups are successfully matched, individual features within them may still be misaligned. However, compared to baseline methods that perform global feature matching, the hierarchical approach adopted in our method significantly reduces the search space. Rather than matching n features globally, our method confines the search to $\sqrt{n}$ features within each of the $\sqrt{n}$ regions, making the task more manageable and less prone to error. This structured approach enhances both efficiency and accuracy, as demonstrated in Table 1.

5 Conclusions

For image matching, increased field-of-views captured in increased image resolutions necessitates a quadratic increase in keypoints due to the two-dimensional nature of images. Our research introduces a novel approach, proposing an image matching method that circumvents the exhaustive NN search in descriptor space for each keypoint. Central to this approach is the concept of first matching clusters of spatial features, and subsequently matching individual features within these matched clusters. This strategy leverages the spatial relationships between features, which improves the efficiency of the method. Empirical evaluations support the effectiveness of this approach.

The computational overhead resulting from the preparation of groups is asymptotically negligible. In practice, even if the feature count is small and overhead becomes significant, the additional computation is applied to all images individually rather than to each pair of images. Consequently, as images are repeatedly used (e.g., in multi-view image matching), the additional cost diminishes rapidly.

The current version of our proposed feature matching process is not feature-agnostic, which could restrict its compatibility with future feature extractors. However, most extractors can be adapted to yield high-dimensional descriptors. In higher-dimensional spaces, these descriptors tend to be pairwise orthogonal, fulfilling our primary requirement. Moreover, using a single descriptor extractor would generate vectors with components that are less correlated than those in concatenated descriptors. This reduction in correlation enhances both discriminability and orthogonality, potentially making our approach more effective than it currently is.

Author contributions E.C. wrote the main manuscript text, run the experiments and prepared the figures. Y.B. and M.O. co-supervised the research. All authors reviewed the manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability Data and code are available at https://drive.google.com/drive/folders/1tgtyRi3vob-pE8WqJtczPGENJpG_Yzk_.

Declarations

Conflict of interests The authors declare no competing interests.

References

1. Levin A, Zomet A, Peleg S, Weiss Y (2004) Seamless image stitching in the gradient domain. In: Computer vision-ECCV 2004: 8th European conference on computer vision, Prague, Czech Republic, May 11–14, 2004. Proceedings, Part IV, vol 8. Springer, pp 377–389
2. Brown M, Lowe DG (2007) Automatic panoramic image stitching using invariant features. Int J Comput Vis 74:59–73
3. Szeliski R (2007) Image alignment and stitching: a tutorial. Found Trends Comput Graph Vis 2(1):1–104
4. Zaragoza J, Chin T-J, Brown MS, Suter D (2013) As-projective-as-possible image stitching with moving dlt. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2339–2346
5. Adel E, Elmogy M, Elbakry H (2014) Image stitching based on feature extraction techniques: a survey. Int J Comput Appl 99(6):1–8
6. Lin C-C, Pankanti SU, Natesan Ramamurthy K, Aravkin AY (2015) Adaptive as-natural-as-possible image stitching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1155–1163
7. Wang Z, Yang Z (2020) Review on image-stitching techniques. Multimedia Syst 26(4):413–430
8. Ullman S (1979) The interpretation of structure from motion. Proc Roy Soc Lond Ser B Biol Sci 203(1153):405–426
9. Bastanlar Y, Temizel A, Yardimci Y, Sturm P (2010) Effective structure-from-motion for hybrid camera systems. In: Proceedings of the international conference on pattern recognition
10. Wu C (2011) VisualSFM: A Visual Structure from Motion System. Accessed 13 Apr. 2025. http://ccwu.me/vsfm/
11. Wu C (2013) Towards linear-time incremental structure from motion. In: 2013 International conference on 3D vision-3DV 2013. IEEE, pp 127–134
12. Schonberger JL, Frahm J-M (2016) Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4104–4113
13. Schönberger JL, Zheng E, Frahm J-M, Pollefeys M (2016) Pixelwise view selection for unstructured multi-view stereo. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III, vol 14. Springer, pp 501–518
14. Moulon P, Monasse P, Perrot R, Marlet R (2017) Openmvg: Open multiple view geometry. In: Reproducible research in pattern recognition: first international workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised selected papers 1. Springer, pp. 60–74
15. Lindenberger P, Sarlin P-E, Larsson V, Pollefeys M (2021) Pixel-perfect structure-from-motion with featuremetric refinement. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5987–5997
16. Wang X, Wang C, Liu B, Zhou X, Zhang L, Zheng J, Bai X (2021) Multi-view stereo in the deep learning era: a comprehensive review. Displays 70:102102
17. Montemerlo M, Thrun S, Koller D, Wegbreit B (2002) Fastslam: A factored solution to the simultaneous localization and mapping problem. Aaai/iaai 593598:593–598
18. Durrant-Whyte H, Bailey T (2006) Simultaneous localization and mapping: part i. IEEE Robot Autom Mag 13(2):99–110
19. Bailey T, Durrant-Whyte H (2006) Simultaneous localization and mapping (slam): part ii. IEEE Robot Autom Mag 13(3):108–117
20. Thrun S (2008) Simultaneous localization and mapping. In: Robotics and cognitive approaches to spatial mapping. Springer, pp 13–41
21. Mur-Artal R, Montiel JMM, Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot 31(5):1147–1163
22. Fuentes-Pacheco J, Ruiz-Ascencio J, Rendón-Mancha JM (2015) Visual simultaneous localization and mapping: a survey. Artif Intell Rev 43:55–81
23. Stachniss C, Leonard JJ, Thrun S (2016) Simultaneous localization and mapping. Springer handbook of robotics. Springer, New York, pp 1153–1176
24. Placed JA, Strader J, Carrillo H, Atanasov N, Indelman V, Carlone L, Castellanos JA (2023) A survey on active simultaneous localization and mapping: state of the art and new frontiers. IEEE Trans Rob 39(3):1686–1705
25. Cavalli L, Larsson V, Oswald MR, Sattler T, Pollefeys M (2020) Handcrafted outlier detection revisited. In: Computer vision-ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, Proceedings, Part XIX vol 16. Springer, pp 770–787
26. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1), 2(331–340), 2
27. Jin Y, Mishkin D, Mishchuk A, Matas J, Fua P, Yi KM, Trulls E (2021) Image matching across wide baselines: from paper to practice. Int J Comput Vis 129(2):517–547
28. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision, vol.2. IEEE, pp 1150–1157
29. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
30. Ke Y, Sukthankar R (2004) Pca-sift: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004, vol 2. IEEE
31. Bastanlar Y, Temizel A, Yardimci Y (2010) Improved sift matching for image pairs with scale difference. Electron Lett 46(5):346–348
32. Yu G, Morel J-M (2011) Asift: an algorithm for fully affine invariant comparison. Image Process On Line 1:11–38
33. Brown M, Süsstrunk S (2011) Multi-spectral sift for scene category recognition. In: CVPR 2011. IEEE, pp 177–184
34. Arandjelović R, Zisserman A (2012) Three things everyone should know to improve object retrieval. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 2911–2918
35. Mishchuk A, Mishkin D, Radenovic F, Matas J (2017) Working hard to know your neighbor's margins: local descriptor learning loss. Adv Neural Inf Proc Syst 30
36. DeTone D, Malisiewicz T, Rabinovich A (2018) Superpoint: self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 224–236
37. Barroso-Laguna A, Riba E, Ponsa D, Mikolajczyk K (2019) Key.net: keypoint detection by handcrafted and learned cnn filters. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5836–5844
38. Luo Z, Shen T, Zhou L, Zhang J, Yao Y, Li S, Fang T, Quan L (2019) Contextdesc: local descriptor augmentation with

cross-modality context. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2527–2536
39. Tian Y, Barroso Laguna A, Ng T, Balntas V, Mikolajczyk K (2020) Hynet: learning local descriptor with hybrid similarity measure and triplet loss. Adv Neural Inf Process Syst 33:7401–7412
40. Tyszkiewicz M, Fua P, Trulls E (2020) DISK: learning local features with policy gradient. Adv Neural Inf Process Syst 33:14254–14265
41. Gleize P, Wang W, Feiszli M, Sil K (2023) Simple learned keypoints. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 22499–22508
42. Calonder M, Lepetit V, Strecha C, Fua P (2010) Brief: binary robust independent elementary features. In: Computer vision–ECCV 2010: 11th European conference on computer vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV, vol 11. Springer, pp 778–792
43. Calonder M, Lepetit V, Özuysal M, Trzcinski T, Strecha C, Fua PV (2012) Brief: computing a local binary descriptor very fast. IEEE Trans Pattern Anal Mach Intell 34:1281–1298
44. Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: an efficient alternative to SIFT or SURF. In: 2011 international conference on computer vision. IEEE, pp 2564–2571
45. Levi G, Hassner T (2016) LATCH: learned arrangements of three patch codes. In: IEEE winter conference on applications of computer vision (WACV). IEEE, pp 1–9
46. Suárez I, Sfeir G, Buenaposada JM, Baumela L (2020) BEBLID: boosted efficient binary local image descriptor. Pattern Recognit Lett 133:366–372
47. Mishkin D, Matas J, Perdoch M (2015) Mods: Fast and robust method for two-view matching. Comput Vis Image Underst 141:81–93
48. Wang L, Tang D, Guo Y, Do MN (2015) Common visual pattern discovery via nonlinear mean shift clustering. IEEE Trans Image Process 24(12):5442–5454
49. Bian J, Lin W-Y, Matsushita Y, Yeung S-K, Nguyen T-D, Cheng M-M (2017) Gms: grid-based motion statistics for fast, ultra-robust feature correspondence. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4181–4190
50. Ma J, Zhao J, Jiang J, Zhou H, Guo X (2019) Locality preserving matching. Int J Comput Vis 127:512–531
51. Wang L, Chen B, Xu P, Ren H, Fang X, Wan S (2020) Geometry consistency aware confidence evaluation for feature matching. Image Vis Comput 103:103984
52. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
53. Silpa-Anan C, Hartley R (2008) Optimised kd-trees for fast image descriptor matching. In: 2008 IEEE conference on computer vision and pattern recognition. IEEE, pp 1–8
54. Strecha C, Bronstein A, Bronstein M, Fua P (2011) Ldahash: improved matching with smaller descriptors. IEEE Trans Pattern Anal Mach Intell 34(1):66–78
55. Cheng J, Leng C, Wu J, Cui H, Lu H (2014) Fast and accurate image matching with cascade hashing for 3d reconstruction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–8
56. Wang M, Xu X, Yue Q, Wang Y (2021) A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. arXiv:2101.12631
57. Fu C, Xiang C, Wang C, Cai D (2017) Fast approximate nearest neighbor search with the navigating spreading-out graph. arXiv preprint arXiv:1707.00143
58. Malkov YA, Yashunin DA (2018) Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans Pattern Anal Mach Intell 42(4):824–836
59. Kanerva P (2022) Hyperdimensional computing: an algebra for computing with vectors. Wiley Online Library, pp 25–42
60. Bradski G (2000) The opencv library. J Softw Tools 25(11):120–123
61. Aldridge M, Johnson O, Scarlett J (2019) Group testing: an information theory perspective. Found Trends Commun Inf Theory 15(3):196–392
62. Chum O, Matas J, Kittler J (2003) Locally optimized ransac. In: Pattern recognition: 25th DAGM symposium, Magdeburg, Germany, September 10–12, 2003. Proceedings, vol 25. Springer, pp 236–243
63. Chum O, Matas J (2005) Matching with prosac-progressive sample consensus. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1. IEEE, pp 220–226
64. Brachmann E, Krull A, Nowozin S, Shotton J, Michel F, Gumhold S, Rother C (2017) Dsac-differentiable ransac for camera localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6684–6692
65. Barath D, Matas J Graph-cut RANSAC. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6733–6741
66. Brachmann E, Rother C (2019) Neural-guided ransac: learning where to sample model hypotheses. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 4322–4331
67. Barath D, Noskova J, Ivashechkin M, Matas J (2020) Magsac++, a fast, reliable and accurate robust estimator. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1304–1312
68. Ivashechkin M, Barath D, Matas J (2021) Vsac: efficient and accurate estimator for h and f. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 15243–15252
69. Barath D, Cavalli L, Pollefeys M (2022) Learning to find good models in ransac. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 15744–15753
70. Cavalli L, Barath D, Pollefeys M, Larsson V (2023) Consensus-adaptive ransac. arXiv preprint arXiv:2307.14030
71. Mikolajczyk K, Schmid C (2005) A performance evaluation of local descriptors. IEEE Trans Pattern Anal Mach Intell 27(10):1615–1630
72. Lebeda K, Matas J, Chum O (2012) Fixing the locally optimized ransac—full experimental evaluation. In: British machine vision conference, vol 2. Citeseer
73. DeTone D, Malisiewicz T, Rabinovich A (2016) Deep image homography estimation. arXiv preprint arXiv:1606.03798
74. Balntas V, Lenc K, Vedaldi A, Mikolajczyk K (2017) Hpatches: a benchmark and evaluation of handcrafted and learned local descriptors. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 3852–3861
75. Le H, Liu F, Zhang S, Agarwala A (2020) Deep homography estimation for dynamic scenes. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 7649–7658
76. Cao S-Y, Hu J, Sheng Z, Shen H-L Iterative deep homography estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1879–1888
77. Li Y, Chen K, Sun S, He C (2022) Multi-scale homography estimation based on dual feature aggregation transformer. IET Image Process 17:1403–1416
78. Hong M, Lu Y, Ye N, Lin C, Zhao Q, Liu S (2022) Unsupervised homography estimation with coplanarity-aware gan. In: 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 17642–17651
79. Luo Y, Wang X, Wu Y, Shu C (2022) Detail-aware deep homography estimation for infrared and visible image. Electronics 11(24):4185

80. Luo Y, Wang X, Liao Y, Fu Q, Shu C, Wu Y, He Y (2023) A review of homography estimation: advances and challenges. Electronics 12(24):4977
81. Cao S, Zhang R, Luo L, Yu B, Sheng Z, Li J, Shen H (2023) Recurrent homography estimation using homography-guided image warping and focus transformer. In: 2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 9833–9842
82. Liao Y, Luo Y, Wang X (2023) Unsupervised deep infrared and visible homography estimation algorithm based on content-aware. Proceedings of the 2023 3rd International Conference on Big Data, Artificial Intelligence and Risk Management, pp 397–402
83. Luo Y, Wang X, Wu Y, Shu C (2023) Infrared and visible image homography estimation using multiscale generative adversarial network. Electronics 12:788
84. Wang X, Luo Y, Fu Q, Rui Y, Shu C, Wu Y, He Z, He Y (2023) Infrared and visible image homography estimation based on feature correlation transformers for enhanced 6g space-air-ground integrated network perception. Remote Sens 15:3535
85. Ivashechkin M, Baráth D, Matas J (2021) Usacv20: robust essential, fundamental and homography matrix estimation. ArXiv arXiv:abs/2104.05044
86. Santellani E, Sormann C, Rossi M, Kuhn A, Fraundorfer F (2022) Md-net: multi-detector for local feature extraction. In: 2022 26th international conference on pattern recognition (ICPR). IEEE, pp 3944–3951
87. Suwanwimolkul S, Komorita S, Tasaka K (2021) Learning of low-level feature keypoints for accurate and robust detection. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 2262–2271
88. Riba E, Mishkin D, Ponsa D, Rublee E, Bradski G (2020) Kornia: an open source differentiable computer vision library for pytorch. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 3674–3683