EFFICIENT IMAGE MATCHING USING HYPERDIMENSIONAL COMPUTING AND GROUP TESTING

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by
Ersin ÇİNE

July 2024
İZMİR

We approve the thesis of **Ersin ÇİNE**

Examining Committee Members:

Prof. Dr. Yalın BAŞTANLAR
Department of Computer Engineering, İzmir Institude of Technology

Prof. Dr. Bilge KARAÇALI
Department of Electrical and Electronics Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Kaya OĞUZ
Department of Computer Engineering, İzmir University of Economics

Assoc. Prof. Dr. Zerrin IŞIK
Department of Computer Engineering, Dokuz Eylül University

Asst. Prof. Dr. Nesli ERDOĞMUŞ
Department of Computer Engineering, İzmir Institude of Technology

Prof. Dr. Yalın BAŞTANLAR
Department of Computer Engineering
İzmir Institude of Technology

Assoc. Prof. Dr. Mustafa ÖZUYSAL
Department of Computer Engineering
İzmir Institude of Technology

Prof. Dr. Onur DEMIRÖRS
Head of the Department of
Computer Engineering

Prof. Dr. Mehtap EANES
Dean of the Graduate School

ACKNOWLEDGMENTS

First and foremost, I thank my family – my mother, Emine Bayram, my sister, Elvan Çine, and my wife, Madina Çine. They lightened the burden of tough days and amplified the joy of happy days. They are the ones giving life and all of this meaning. I cannot imagine life without any of them.

I am profoundly thankful to my supervisor, Dr. Yalın Baştanlar. Despite joining as my supervisor recently, his invaluable guidance, significant contributions, and empathetic support have been instrumental in completing this thesis.

I extend my deepest gratitude to my co-supervisor, Dr. Mustafa Özuysal, who was my supervisor until recently. He dedicated an enormous amount of time and effort to my research, introduced many ideas, and served as an inspiring role model. His commitment and contributions were pivotal in shaping this work.

I am grateful to Dr. Bilge Karaçalı and Dr. Nesli Erdoğmuş, members of my thesis monitoring committee, for their time, support, and feedback. My appreciation also goes to Dr. Kaya Oğuz and Dr. Zerrin Işık, who graciously served on my defense jury.

I thank my friends and colleagues: Dr. Semih Orhan, Kerem Delikoyun, Emre Cem Dönmez, Dr. Furkan Eren Uzyıldırım, Thimo Wellner, Büşra Çalmaz, Nuri Furkan Pala, and all others.

Last but not least, I extend gratitude to Alexandra Elbakyan and the many members of the open source communities for their contributions to a fairer world, and to ChatGPT for its assistance in enhancing my writing.

This journey has been a significant challenge due to personal difficulties and suboptimal decisions. In retrospect, even the greatest ideas of humanity appear straightforward. However, the process of turning the hierarchical matching pipeline from awful to mediocre took dozens of attempts and a couple of years, always with the risk of achieving nothing in the end; the rest was easy and quick. Although I believe that the suffering in the past is *always* there—our minds and bodies remember it, and in a cosmic sense, nothing is truly lost—I am grateful to have survived.

Thank you all for helping me push the boundaries of science, though the progress may have been in a specific direction and by a small amount.

ABSTRACT

EFFICIENT IMAGE MATCHING USING HYPERDIMENSIONAL COMPUTING AND GROUP TESTING

The widely adopted image matching approach remains dependent on exhaustive matching of local features across images. We challenge this approach and investigate enhancing matching efficiency by not approximating nearest neighbors but using a hierarchical approach. We hypothesize that efficiently identifying sufficiently similar geometrically meaningful feature matches, as opposed to the most similar but geometrically random ones, can improve or maintain matching performance, with lower computational complexity. We propose a novel method named group-guided nearest neighbors, which involves matching groups of features as one and then matching individual features across matched groups only. The hierarchical pipeline reduces the time complexity of feature matching from $\Theta(n^2)$ to $\Theta(n\sqrt{n})$. Empirical results on homography and pose estimation indicate that our method outperforms the nearest neighbors algorithm and achieves the performance level of other inefficient methods. We formulate the proposed method as a general framework that offers a continuum of methods with varying levels of computational cost. Additionally, we introduce a linear-time matching algorithm which first tests memberships of the most distinct features to feature groups of the other image, then matches these distinct features only with the members of the matched groups. Experiments show that this algorithm performs better than linear-time adaptations of quadratic-time algorithms. We also propose techniques for generating better synthetic image pair datasets for homography estimation and faster evaluation of image matching pipelines. These contributions result in an improved image matching framework with more realistic datasets, faster evaluation, and high-performance matchers with various time complexities.

ÖZET

HİPER BOYUTLU HESAPLAMA VE GRUP TESTİ KULLANARAK VERİMLİ İMGE EŞLEME

Yaygın olarak kullanılan imge eşleme yaklaşımı, imgeler arasında yerel özniteliklerin kapsamlı bir şekilde eşleştirilmesine dayanmaktadır. Bizler, bu yaklaşımı karşımıza alıyor ve en yakın komşular üzerinden tahmin yaparak değil de hiyerarşik bir yaklaşım kullanarak eşleme verimliliğinin artırılmasını inceliyoruz. En benzer ancak geometrik olarak rastgele öznitelik eşlemelerinin aksine, yeterince benzer ve geometrik olarak anlamlı öznitelik eşlemelerinin verimli bir şekilde saptanmasının, daha düşük hesaplama karmaşıklığı ile eşleşme performansını artırabileceğini veya koruyabileceğini varsayıyoruz. Grup güdümlü en yakın komşular adında yeni bir yöntem öneriyoruz. Bu yöntem, öznitelik gruplarının bir olarak eşleşmesini ve ardından yalnızca eşleşen gruplar arasında bireysel özniteliklerin eşleşmesini içerir. Hiyerarşik boru hattı, öznitelik eşlemenin zaman karmaşıklığını $\Theta(n^2)$'den $\Theta(n\sqrt{n})$'ye düşürür. Homografi ve poz tahminine ilişkin deneysel sonuçlar, bizim yöntemimizin en yakın komşu algoritmasından daha iyi bir sonuç verdiğini ve diğer verimsiz yöntemlerin performansını yakaladığını göstermektedir. Önerilen yöntemi, değişen seviyelerde hesaplama maliyetlerine sahip yöntemlerin devamlılığını sunan genel bir çerçeve olarak ifade ediyoruz. Ayrıca öncelikle en belirgin özniteliklerin diğer imgenin öznitelik gruplarına üyeliklerini test eden, ardından bu belirgin öznitelikleri yalnızca eşleşen grupların üyeleriyle eşleştiren bir doğrusal zamanlı eşleme algoritması sunuyoruz. Deneyler gösteriyor ki, bu algoritma karesel zaman algoritmalarının doğrusal zaman uyarlamalarından daha iyi bir performans sergiliyor. Homografi tahmini için daha iyi sentetik imge çifti veri kümeleri oluşturulması ve imge eşleme boru hatlarının daha hızlı değerlendirilmesi için teknikler de sunuyoruz. Bu katkılar sonucunda daha gerçekçi veri kümeleri içeren, daha hızlı değerlendirme yapan ve çeşitli zaman karmaşıklıklarına sahip yüksek performanslı eşleştiriciler içeren iyileştirilmiş bir imge eşleme çerçevesi ortaya çıkmaktadır.

To my family,

You are a *sufficient condition* for my happiness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 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 (Levin et al. 2004; Brown and Lowe 2007; Szeliski et al. 2007; Zaragoza et al. 2013; Adel, Elmogy, and Elbakry 2014; Lin et al. 2015; Wang and Yang 2020), 3D reconstruction (Ullman 1979; Wu 2011, 2013; Schonberger and Frahm 2016; Schönberger et al. 2016; Moulon et al. 2017; Lindenberger et al. 2021; Xiang Wang et al. 2021; Bastanlar et al. 2010), and simultaneous localization and mapping (Montemerlo et al. 2002; Durrant-Whyte and Bailey 2006; Bailey and Durrant-Whyte 2006; Thrun 2008; Mur-Artal, Montiel, and Tardos 2015; Fuentes-Pacheco, Ruiz-Ascencio, and Rendón-Mancha 2015; Stachniss, Leonard, and Thrun 2016; Placed et al. 2023). Figure 1.1 illustrates the inputs, outputs, and sample applications of image matching.

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 (Cavalli et al. 2020), 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 (Muja and Lowe 2009) result in a substantial degradation of performance (Jin et al. 2021).

To the best of our knowledge, no work has improved the efficiency of exhaustive feature matching without compromising its accuracy.

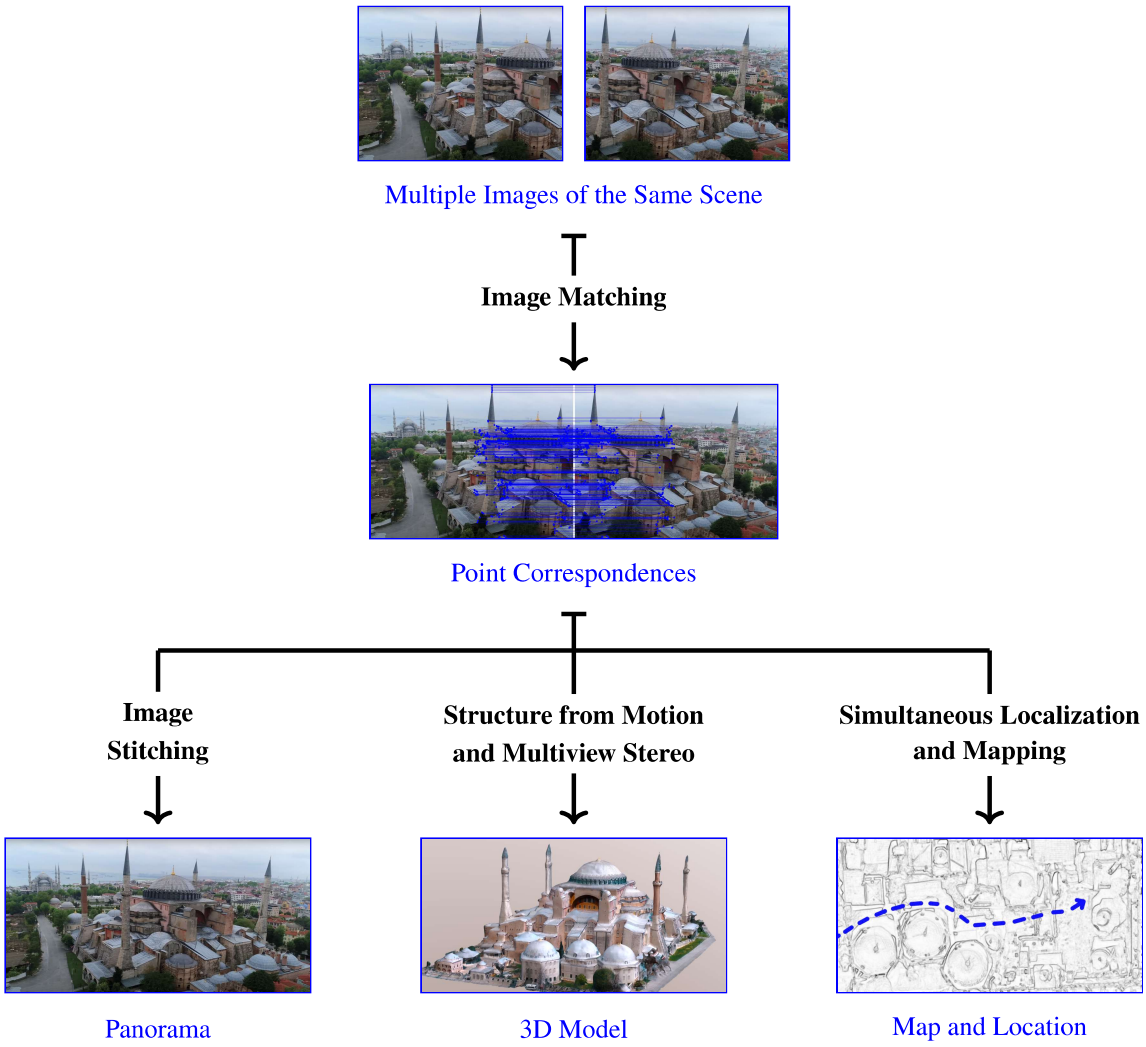


Figure 1.1. Image Matching with Sample Applications

1.1. Thesis Statement

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 correspondences, 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 correspondences, rather than the most similar but geometrically random ones, can potentially improve or at

least maintain matching performance.

We draw inspiration from the human visual system which recognizes objects based on high-level features, advanced feature filtering methods such as AdaLAM (Cavalli et al. 2020), which verifies feature matches locally around distinct matches, and recent group testing (Iscen and Chum 2018) and hyperdimensional computing (Neubert and Schubert 2021) methods applied to computer vision problems.

We propose two distinct feature matching strategies, both employing a hierarchical approach. Instead of directly matching individual features across images, we first group the features within each image. The Group-Guided Nearest Neighbors (GGNN) algorithm matches feature groups first and then matches individual features within those groups. Conversely, the Group-Tested Nearest Neighbors (GTNN) algorithm matches individual features to feature groups initially, and subsequently matches individual features within the matched groups. Figure 1.2 illustrates the matching strategies.

Both algorithms reduce the time complexity of the matching process from $\Theta(n^2)$ to $\Theta(n\sqrt{n})$ when considering $\sqrt{n}$ groups of features, where n represents the total number of features. Additionally, GTNN improves efficiency further and achieves $\Theta(n)$ time complexity by matching only the top $\sqrt{n}$ features, rather than all n features, to all groups in the other image.

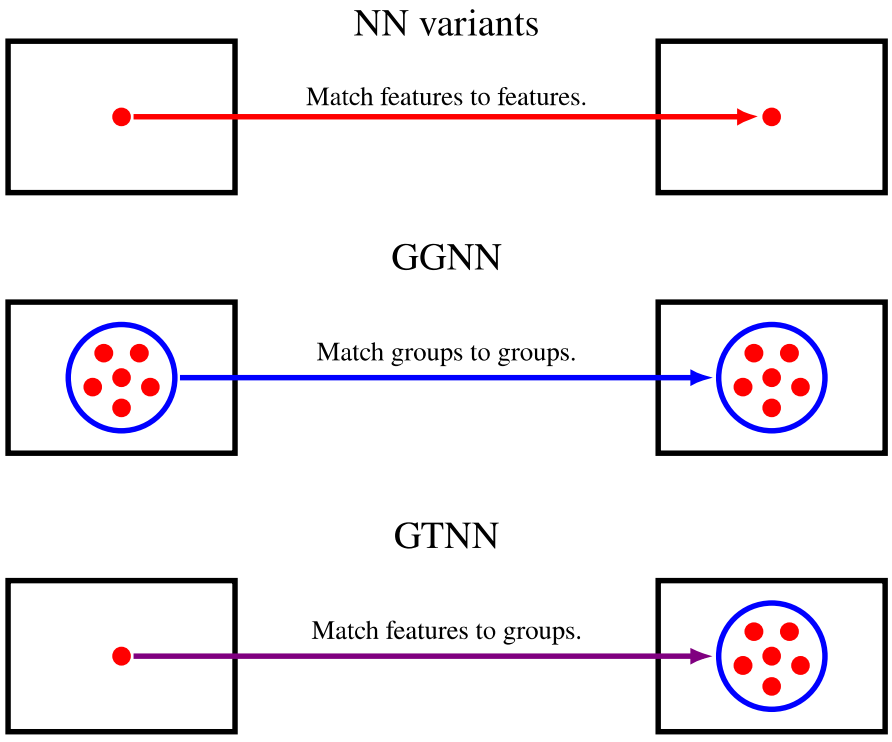


Figure 1.2. Feature Matching Strategies: This figure illustrates the initial stages of NN variants and two proposed group-based matching methods.

1.2. Thesis Structure

The organization of the thesis is as follows:

Chapter 1: We introduce the primary problem addressed in this research and outline our proposed approach.

Chapter 2: We provide the necessary background knowledge, covering fundamental concepts and discussing related work in the literature.

Chapter 3: We present a concrete solution to the problem. This includes the introduction of group-guided nearest neighbors (GGNN), along with its pre- and post-processing stages. We demonstrate the efficiency of this solution, which has a time complexity of $\Theta(n\sqrt{n})$ instead of $\Theta(n^2)$, and its effectiveness in various homography estimation tasks.

Chapter 4: We generalize the computation of hierarchical regions, enabling the use of any number of regions with any number of features. We introduce group-tested nearest neighbors (GTNN), an alternative to GGNN that compares top features with groups of features, achieving linear time complexity and thus being more efficient. We present methods for quick approximate evaluation of image matchers for hyperparameter optimization and an accurate image warping algorithm utilizing single image superresolution techniques to create realistic synthetic image pairs. Lastly, we extend estimation-guided matching to epipolar geometry and assess the pose estimation performance of the hierarchical matching algorithm.

Chapter 5: We summarize our contributions and findings and outline potential directions for future work.

CHAPTER 2

BACKGROUND

This chapter presents an overview of the fundamentals of image matching, the key concepts that inspire our work, and the relevant literature essential for understanding the discussions that follow in this thesis.

2.1. Image Matching Fundamentals

Image matching is the process of establishing point correspondences between images of the same scene to estimate the geometric relationship between the cameras.

Figure 2.1 shows a pair of images that can be matched. The process of estimating geometric transformation between images typically begins with the extraction of sparse, local features from each image. These features consist of keypoints, which are distinct interest points, and their associated descriptor vectors computed from the surrounding image patches. The features are then matched across the images by identifying the most similar pairs and retaining only the geometrically consistent matches. Once reliable point correspondences between the images are established, estimating the transformation parameters becomes straightforward by minimizing the squared errors.

Figure 2.2 visualizes keypoints of the images. Numerous algorithms exist for keypoint detection, each with its own strengths and characteristics. Some algorithms offer better localization and repeatability, while others prioritize speed. There are both handcrafted and learned methods. Certain algorithms detect corners, while others detect blobs. Some detect circles of uniform sizes, while others detect circles of varying sizes to better handle scale differences between images. Additionally, some algorithms calculate a dominant 2D orientation to normalize the patch to a canonical form, and some identify affine shapes to enhance robustness to viewpoint changes.

Once the keypoints are detected, the next step is to describe the image patches marked by keypoints with descriptor vectors that are invariant or robust to geometric and photometric differences, such as changes in viewpoint or brightness. This ensures that the keypoints can be reliably matched across different images despite these variations. Some algorithms only detect keypoints, some only compute descriptors, and some perform both tasks. We refer to a keypoint and its corresponding descriptor jointly as a ‘local feature’ (in short, ‘feature’).

Figure 2.1. Sample Inputs for Image Matching: Because the scene is planar, the two images are related by a projective transformation, which can be represented by a homography matrix with 8 degrees of freedom. In this case, the task of an image matcher is to accurately estimate these 8 parameters, which is possible by determining 4 perfect point correspondences or more imperfect ones. If the camera movement were constrained to simpler transformations, there would be fewer parameters to estimate. For example, if there were only a 2D rotation, only one parameter, the angle, would need to be estimated. In contrast, if the scene were three-dimensional, more parameters would need to be estimated for general camera movements, either the essential matrix or the fundamental matrix, depending on whether the cameras are calibrated.

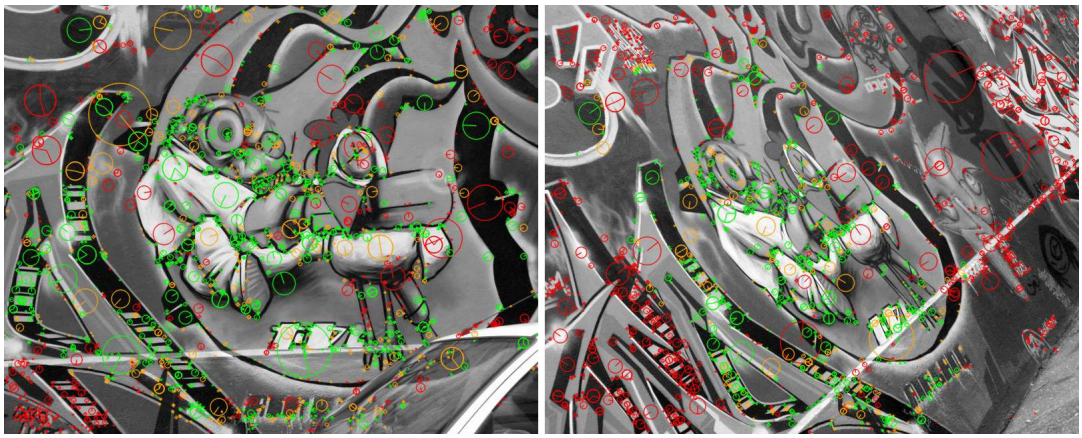


Figure 2.2. Outputs of Keypoint Detection: 2048 Difference of Gaussians (DoG) keypoints (represented as circles with orientations) are detected independently in both images. Some keypoints (marked in green) are repeated in the other image, meaning their centers, when projected using the true homography matrix, align closely with a keypoint center in the other image. Orange-marked keypoints are repeated with a less strict threshold, while red-marked keypoints are not repeated. High repeatability is a desired characteristic of keypoint detectors, as non-repeated keypoints cannot be matched, even with a hypothetical perfect feature matcher. Note that due to non-overlapping regions in the images, certain keypoints cannot be repeated, such as those close to the bottom-left and top-right corners of the right image.

Figure 2.3 visualizes the tentative feature matches between the images. Note that this matching procedure is not geometry-aware, and no filtering based on geometric verification has been applied yet.

Tentative feature matches often contain mismatches, known as outliers. Typically, outliers are much more common than inliers. However, it is often possible to filter out the outliers because they are generally geometrically random and, consequently, inconsistent with other matches. In contrast, inliers, even if few in number, are all consistent with each other and have an approximate consensus on the transformation

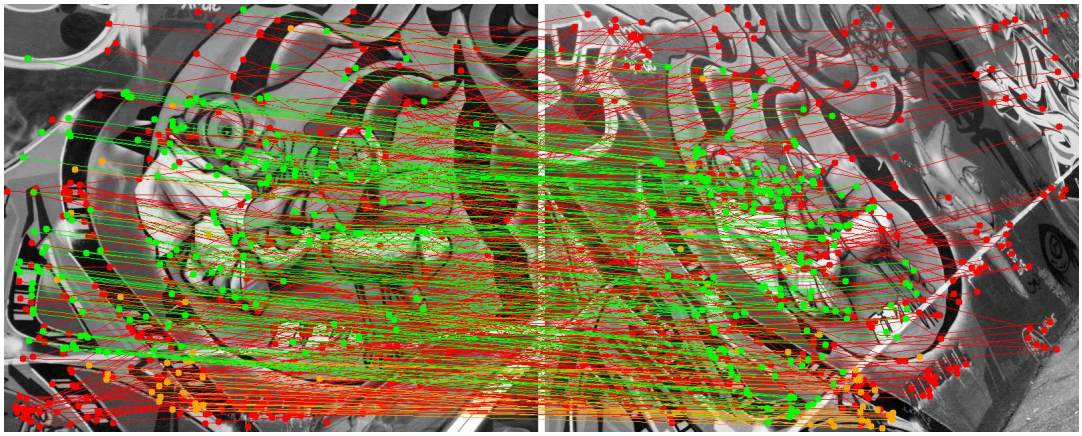


Figure 2.3. Outputs of Feature Matching: SIFT features (centers of the previously detected DoG keypoints and their corresponding descriptor vectors) are matched with their nearest neighbors in descriptor space across images. The search is bidirectional, retaining only mutual matches. Additionally, only the 512 matches with the smallest descriptor distances are kept. Green-marked matches are the best, orange-marked matches are moderate, and red-marked matches are the worst, as compared to the ground truth homography matrix.

parameters. Geometric verification can be applied either locally or globally. Local geometric verification can be used as a prefiltering method to reduce the contamination rate—the ratio of the number of outliers to all matches—thereby increasing the success chance of global geometric verification. Global geometric verification is performed using a Monte Carlo algorithm, where the time is constrained and success is probabilistic. In the global approach, a large consistent subset is identified among all matches, with the remaining matches being rejected. Figure 2.4 visualizes the inliers identified solely through global geometric verification, excluding any local verification among the previously established feature matches.

The Direct Linear Transform (DLT) algorithm works by solving a set of linear equations derived from point correspondences between two images. Since each 2D point provides two coordinates, solving for the transformation parameters requires establishing n point correspondences to satisfy $2n$ degrees of freedom. For instance, to estimate the parameters of a projective transformation, a minimum of four point correspondences is required. However, in practical applications, due to localization errors, approximately 20 point correspondences are typically needed to achieve accurate results. In this case, the system is overdetermined, meaning there are more correspondences than the theoretical minimum. Singular Value Decomposition (SVD) is used to find a least-squares solution by minimizing the sum of squared residuals, considering all the correspondences. Figure 2.5 visualizes the estimated parameters alongside the ground truth.

Table 2.1 shows the important properties of the most common 2-dimensional geometric transformations. A rigid transformation is also known as a Euclidean transformation. These transformations may occur in planar scenes when the cameras are moved. For example, a similarity transformation occurs when the camera moves in three dimensions but only changes its orientation in the roll direction. Roll refers to the

Figure 2.4. Outputs of Outlier Filtering: Outlier filtering, or geometric verification, involves determining the transformation parameters that have the largest consensus among the provided point correspondences. The most popular algorithm for performing this task is Random Sample Consensus (RANSAC). The RANSAC algorithm is used to reject outliers among the matches. In its simplest form, RANSAC randomly samples a minimal subset of matches to estimate the transformation parameters, which in this case requires 4 point correspondences since they produce 8 equations to estimate 8 parameters. Then, all matches that are sufficiently consistent with the parameters suggested by the minimal subset are counted. This process is repeated many times, and the minimal subset with the best support (highest count) is chosen as the best hypothesis. Matches inconsistent with this best hypothesis are then eliminated. RANSAC has many variants that improve upon its basic components.

Figure 2.5. Estimated and True Homographies: The corners of the left image were projected onto the right image using both the estimated homography (marked in red) and the true homography (marked in green). The estimated quadrilateral is nearly invisible because it almost perfectly aligns with the true one. Note that neither the automobile visible in the first image nor the ground visible in the second image prevents accurate determination of the camera movement. Since the scene is mostly planar, these elements are treated as occlusions that must be discarded, and local features are capable of handling them effectively.

rotation of the camera around the axis that points in the direction the camera is facing, causing the image plane to rotate in 2D. Figure 2.6 illustrates these transformations. All these transformations are closed under composition and inversion. If H_1 and H_2 are homography matrices representing projective transformations, then their products H_1H_2 and H_2H_1, as well as their inverses H_1^{-1} and H_2^{-1}, are also homography matrices.

Table 2.1. Expressive Powers of 2D Geometric Transformations

	Translation	Rigid	Similarity	Affine	Projective
Degrees of freedom	2	3	4	6	8
Translation	✓	✓	✓	✓	✓
Rotation		✓	✓	✓	✓
Uniform scaling			✓	✓	✓
Nonuniform scaling				✓	✓
Shear				✓	✓
Perspective projection					✓
Composition of projections					✓

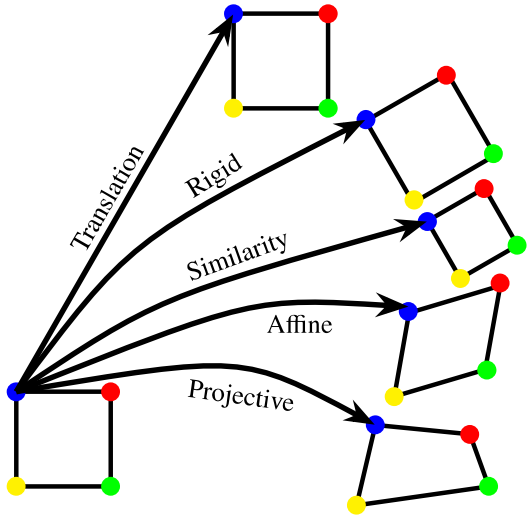


Figure 2.6. 2D Geometric Transformations: This figure illustrates how a square can be transformed under various classes of geometric transformations. The most general transformation is the projective transformation, where all four corners of the square can move independently.

2.2. Key Inspirations

Hyperdimensional computing (HDC) (Kanerva 2009, 2022) is a computational paradigm inspired by high-dimensional vector spaces and human cognitive processes. It uses hypervectors to efficiently encode, process, and retrieve information. In HDC, data points in high-dimensional space undergo operations like binding and bundling (superposition), enabling robust, noise-tolerant computations due to redundant representation. This redundancy ensures data integrity even when some components are altered or lost. HDC excels in associative memory and pattern recognition tasks, manipulating complex data structures with simple mathematical operations. Its resilience to errors, noise, and capacity for parallel processing make HDC a scalable, energy-efficient alternative for AI, cognitive computing, and data-intensive applications. Recently, HDC has been utilized in computer vision for systematically aggregating image descriptors. This approach leverages binding and bundling operations to combine multiple image descriptors into a

single holistic vector, significantly improving performance in tasks like place recognition in mobile robotics (Neubert and Schubert 2021).

Group testing (Du, Hwang, and Hwang 2000; Aldridge, Johnson, and Scarlett 2019) is a combinatorial method for identifying defective items within a large population by testing groups rather than each item individually. Initially proposed during World War II for detecting syphilis in the U.S. Army, this technique involves pooling samples and testing the pools. If a pool tests negative, all items within it are deemed non-defective; if positive, further testing identifies the defectives. Group testing leverages the principles of combinatorial mathematics to design efficient pooling strategies, making it particularly effective in low-prevalence scenarios by significantly reducing the number of tests and saving time and resources. This method is adaptable to various problem settings and is applied in fields such as medical diagnostics, manufacturing quality control, and network security, optimizing the testing process while maintaining high accuracy and reliability. Recently, group testing has been applied in computer vision for efficient approximate nearest neighbor search in large-scale image retrieval (Iscen and Chum 2018). This approach enhances search accuracy while reducing computational complexity, making it suitable for processing large datasets in parallel and in batches.

2.3. Related Work

2.3.1. Features for Sparse Matching

In image matching systems handcrafted feature extractors such as SIFT (David G Lowe 1999, 2004) along with its variants (Ke and Sukthankar 2004; Bastanlar, Temizel, and Yardimci 2010; Yu and Morel 2011; Brown and Süsstrunk 2011; Arandjelović and Zisserman 2012) and alternative real-valued descriptors (Mishchuk et al. 2017; DeTone, Malisiewicz, and Rabinovich 2018; Barroso-Laguna et al. 2019; Z. Luo et al. 2019; Tian et al. 2020; Tyszkiewicz, Fua, and Trulls 2020; Gleize, Wang, and Feiszli 2023) 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 (Calonder et al. 2010; Calonder et al. 2012) and ORB (Rublee et al. 2011), which are handcrafted descriptors, along with LATCH (Levi and Hassner 2016) and BEBLID (Suárez et al. 2020), which are learned descriptors, provide a faster alternative to real-valued descriptors.

2.3.2. Matching of Sparse Features

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) (David G. Lowe 2004) 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 (Jin et al. 2021), 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) (Mishkin, Matas, and Perdoch 2015) and Adaptive Locally-Affine Matching (AdaLAM) (Cavalli et al. 2020) are notable for incorporating geometric constraints into the matching process. FGINN 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. AdaLAM is a more advanced system, filtering NN matches with local geometric verification around SNN matches using Random Sample Consensus (RANSAC) (Fischler and Bolles 1981). Figure 2.7 illustrates the Hasse diagram of the matches produced by these algorithms.

Efficient alternatives to exact NN methods have been extensively explored in the literature. Notably, the Fast Library for Approximate Nearest Neighbors (FLANN) (Muja and Lowe 2009) employs multiple randomized kd-trees (Silpa-Anan and Hartley 2008) and hierarchical k-means trees (Muja and Lowe 2009) to approximate NN. Although these methods offer improved efficiency, they perform worse than exhaustive nearest neighbor searches in terms of matching accuracy (Jin et al. 2021).

Additionally, hash-based methods such as LDAHash (Strecha et al. 2011) and CasHash (Cheng et al. 2014) 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 (M. Wang et al. 2021), which scale well for searching. Examples include the Navigating Spreading-out Graph (NSG)

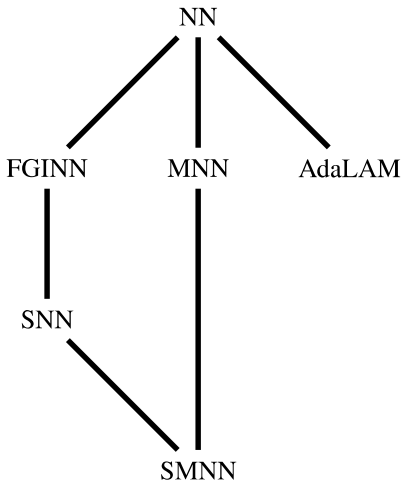


Figure 2.7. Hasse Diagram of the Selected Feature Matching Algorithms: The matches identified by the algorithms are ordered by set inclusion. For instance, MNN is a subset of NN. This partial order is valid only when identical parameters are applied. For example, FGINN is a subset of SNN under the same threshold conditions. This is particularly significant for SMNN, as it is a subset of both MNN and FGINN. To mitigate the strict constraints, SMNN is generally employed with less strict thresholds.

(Fu et al. 2017) and Hierarchical Navigable Small World (HNSW) (Malkov and Yashunin 2018). These algorithms are utilized for general searches in large vector databases and are not limited to local feature matching across two images.

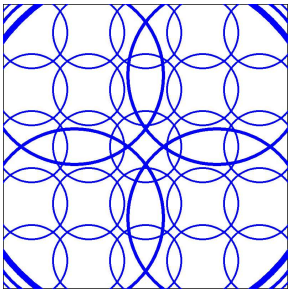
All these tree-based, hash-based, and graph-based approximate nearest neighbor methods are purely descriptor-based and do not incorporate geometric awareness. Consequently, even under optimal conditions, these algorithms can only perform as well as exact NN methods and not better, for sufficiently large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

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

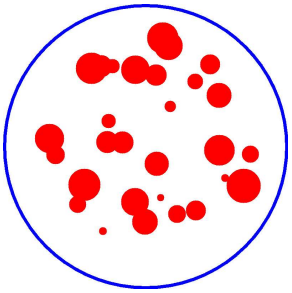
CHAPTER 3

HIERARCHICAL IMAGE MATCHING WITH GROUP-GUIDED NEAREST NEIGHBORS

Exact Nearest Neighbor (NN) search with heuristic filtering remains on the Pareto front for balancing image matching performance and computational efficiency. We propose a hierarchical pipeline that employs hyperdimensional computing for efficient group testing of feature similarities. Figure 3.1 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.



(a) Image with a pyramid of circular regions



(b) Region as a group of local features

Figure 3.1. 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.

3.1. Approach

We propose Group-Guided Nearest Neighbors (GGNN), a hierarchical approach to matching image pairs. Initially, we describe features with vectors that are probably approximately orthogonal to each other. We then spatially group keypoints and represent each group with a single descriptor vector formed through aggregation. Our process matches these groups across images, and we carry out feature matching and

match filtering within these matched groups. By matching these groups across images and performing feature matching across group matches instead of conducting a global search, we enhance efficiency. Our method showcases a lower time complexity than NN search. Figure 3.2 illustrates both matching strategies. Subsequently, we perform a robust estimation of the geometric transformation. Finally, we apply a guided matching procedure to further refine the estimated transformation. Figure 3.3 shows an overview of the proposed system.

3.1.1. 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 (Kanerva 2022), 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 (Levi and Hassner 2016) and 512-dimensional BEBLID (Suárez et al. 2020), computed on Oriented FAST (Rublee et al. 2011) keypoints, emerged as an effective solution in terms of orthogonality and matchability. To enhance orthogonality, we identified more suitable configurations of these algorithms than their default settings in OpenCV (Bradski 2000). Additionally, we implemented non-maximum suppression on keypoints based on their response scores to enhance orthogonality further, in response to our observation that keypoints in close spatial proximity are less likely to yield orthogonal vectors.

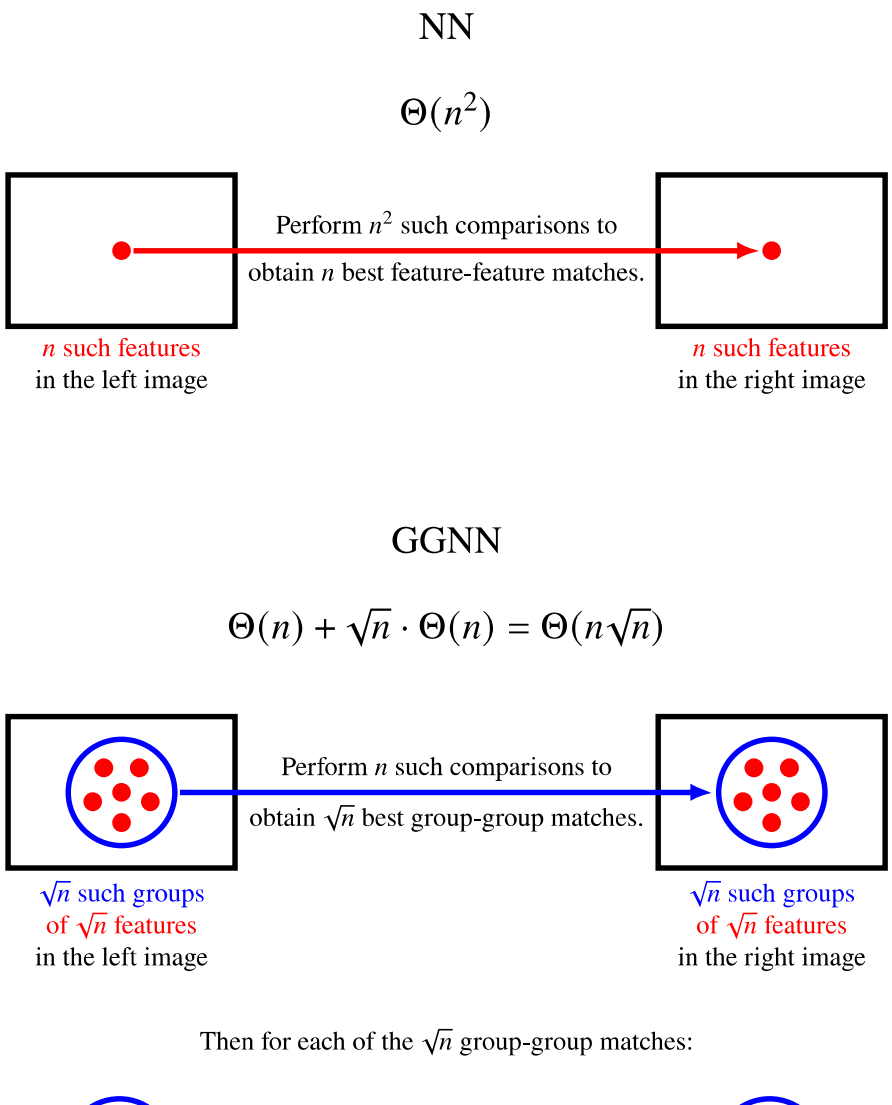


Figure 3.2. Feature matching in $\Theta(n\sqrt{n})$ time: The widely adopted strategy for matching features involves searching for the most similar feature pairs. The NN strategy encompasses not only the basic NN algorithm but also various algorithms that approximate NN, as well as those that rely on NN before filtering the matches. GGNN has a lower time complexity because it does not require an exact NN search. Additionally, it does not approximate NN, meaning that in the best-case scenario, its output differs from NN. The objective of GGNN is to find sufficiently similar, geometrically meaningful matches.

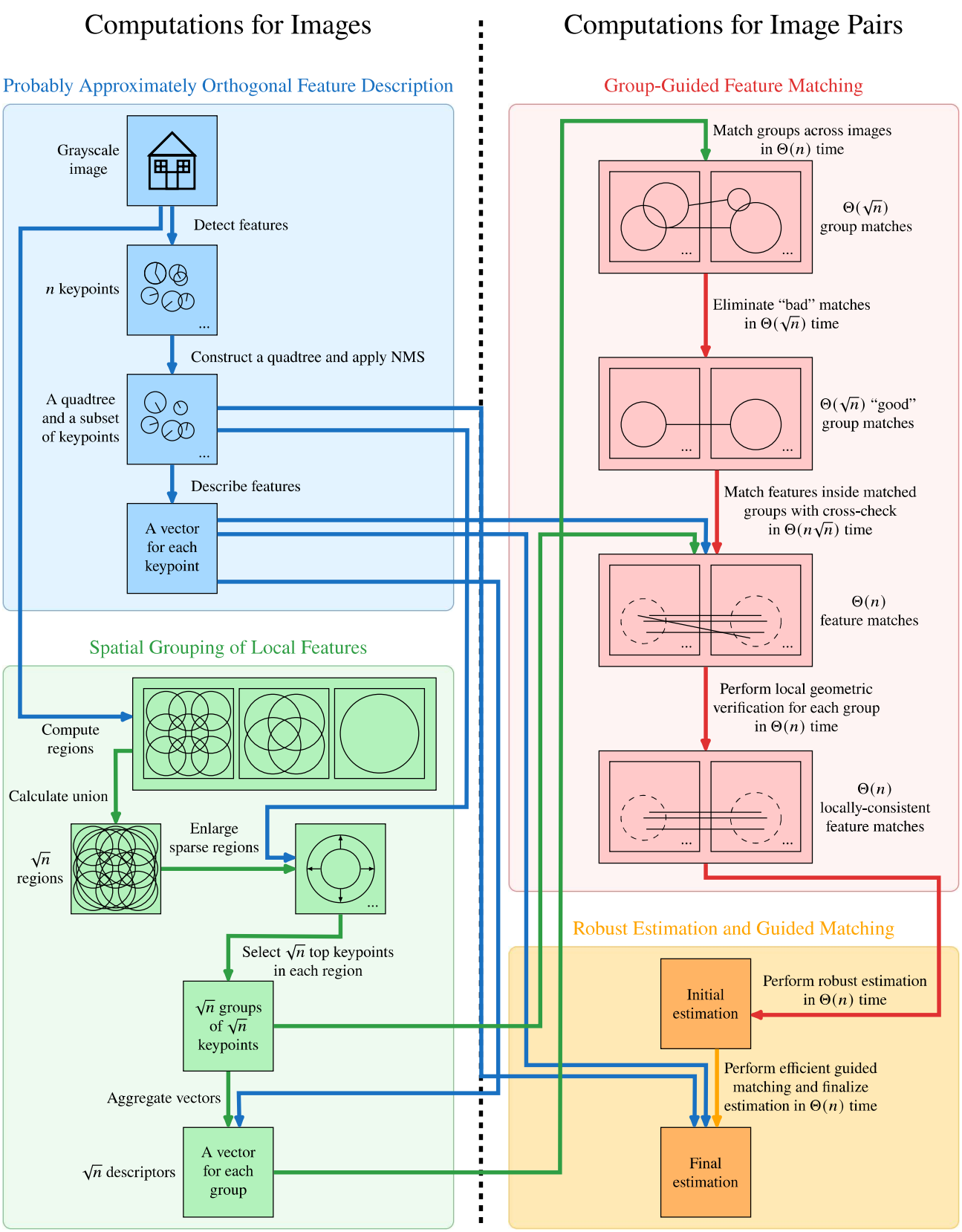


Figure 3.3. Comprehensive System Overview: The diagram illustrates the computational processes applied to individual images (left) and to pairs of images (right). It is important to note that for a dataset of m images, the number of potential image pairs escalates to $\Theta(m^2)$. This exponential increase 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.

3.1.2. 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 handpicked without overfitting. 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.

It was observed that groups are biased towards matching with groups of higher cardinalities, severely limiting the spatial distribution of group matches across the images and significantly reducing image matching performance. To address this issue, we ensured a consistent feature count within each group. In regions with an excess of keypoints, only those with the highest scores were selected. Conversely, if a region had insufficient keypoints, the circle size was increased to include the necessary number. This approach allows for unbaised comparison of aggregated descriptors and leads to better utilization of group count budget.

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 since they all possess the same L_2 norm, with components being either 1 or -1.

3.1.3. 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. 3.3). We compare the group descriptors using cosine similarity, as our experiments demonstrated that it is more effective 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 correspondences.

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 (Fischler and Bolles 1981) in parallel, similar to the method described in (Cavalli et al. 2020). 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 (Aldridge, Johnson, and Scarlett 2019). 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.

3.1.4. 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 (Fischler and Bolles 1981) or its variants (Chum, Matas, and Kittler 2003; Chum and Matas 2005; Brachmann et al. 2017; Barath and Matas 2018; Brachmann and Rother 2019; Barath et al. 2020; Ivashechkin, Barath, and Matas 2021; Barath, Cavalli, and Pollefeys 2022; Cavalli et al. 2023). 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 (David G. Lowe 2004) and scale-based filtering for keypoints as in (Cavalli et al. 2020). 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.

3.1.5. Computational Complexity

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 $\Theta(n\sqrt{n})$.

3.2. Evaluation

We evaluate the performance of the proposed method on the task of homography estimation using the Oxford classic image matching dataset (Mikolajczyk and Schmid 2005). This dataset consists of 6 images for each of the 8 scenes, resulting in 48 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$^+$. In the original dataset, 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 (Lebeda, Matas, and Chum 2012), 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 (DeTone, Malisiewicz, and Rabinovich 2016), 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 (Balntas et al. 2017). 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) (DeTone, Malisiewicz, and Rabinovich 2016; Le et al. 2020; S.-Y. Cao et al. 2022; Li et al. 2022; Hong et al. 2022; Y. Luo et al. 2022; Luo, Wang, Liao, et al. 2023; S. Cao et al. 2023; Liao, Luo, and Wang 2023; Luo, Wang, Wu, et al. 2023; Xingyi Wang et al. 2023), 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 (Ivashechkin, Baráth, and Matas 2021).

In experiments we use $n = 4096$ features. In practice the number of features can be as high as 8k (Cavalli et al. 2020), 10k (Gleize, Wang, and Feiszli 2023), 12k (Tyszkiewicz, Fua, and Trulls 2020), 15k (Jin et al. 2021), 20k (Santellani et al. 2022) or 40k (Suwanwimolkul, Komorita, and Tasaka 2021). The theoretical speedup of the proposed method increases as the number of features increases.

Table 3.1 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) (David G. Lowe 2004). Additionally, we evaluate FGINN (Mishkin, Matas, and Perdoch 2015), a variation of SNN that considers the geometry of keypoints. Recent methods such as AdaLAM (Cavalli et al. 2020) and SMNN (Jin et al. 2021), as implemented in Kornia (Riba et al. 2020), are also included in the comparison. We also include the preemptive feature matching strategy (Wu 2013), which matches top-scale features across images to accelerate multi-view 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 (Muja and Lowe 2009), which is tree-based, and the state-of-the-art HNSW (Malkov and Yashunin 2018), 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 (Barath and Matas 2018). All methods were tuned for optimal performance, except for NN and MNN, which do not require parameter tuning.

Table 3.1. Homography Estimation Results of GGNN and Other Methods

Dataset	**Image Pairs**	**Failure Percentage**									
		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		$\Theta(n^2)$						$\Theta(n\sqrt{n})$		$\Theta(n \log n)$	

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.

Figure 3.4 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 3.5 provides sample results, showcasing both successful matches and various types of failures.

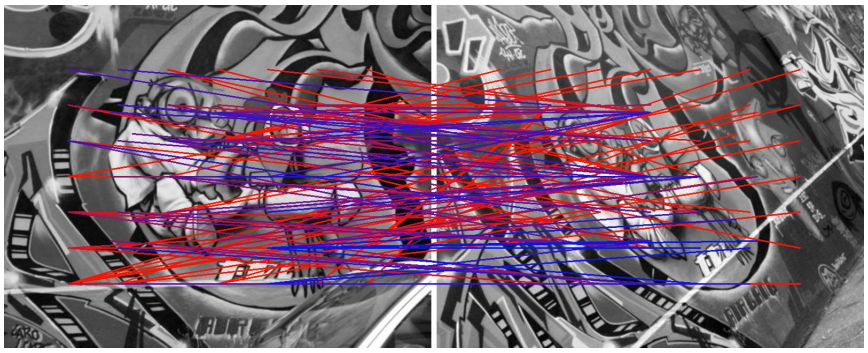
Table 3.2 presents the results of the ablation study, wherein each column represents the proposed method with some components intentionally omitted. The ablation study reveals that performance deteriorates when certain components are removed or modified, highlighting the significance of these components 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.

Table 3.2. Ablation Study

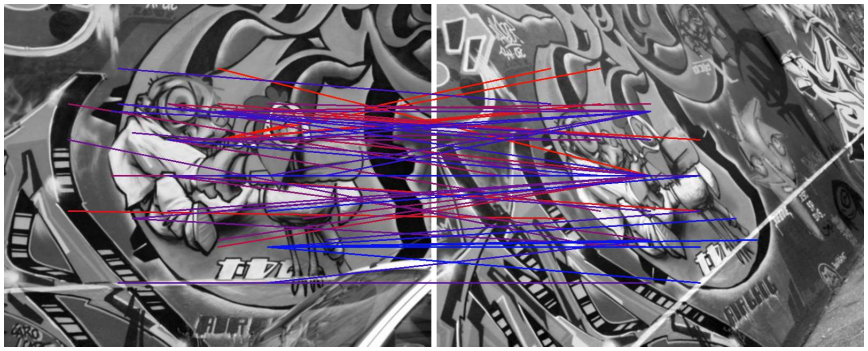
GGNN	**Non-maximum suppression**	✓		✓	✓	✓	✓
	Pyramid for regions	✓	✓		✓	✓	✓
	Region match elimination	✓	✓	✓		✓	✓
	Local geometric verification	✓	✓	✓	✓		✓
	Guided matching	✓	✓	✓	✓	✓	
Failure Percentage	Oxford$^+$ Bikes	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
	Oxford$^+$ Trees	2.8	2.8	2.8	**0.0**	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		$\Theta(n\sqrt{n})$					

3.3. Discussion

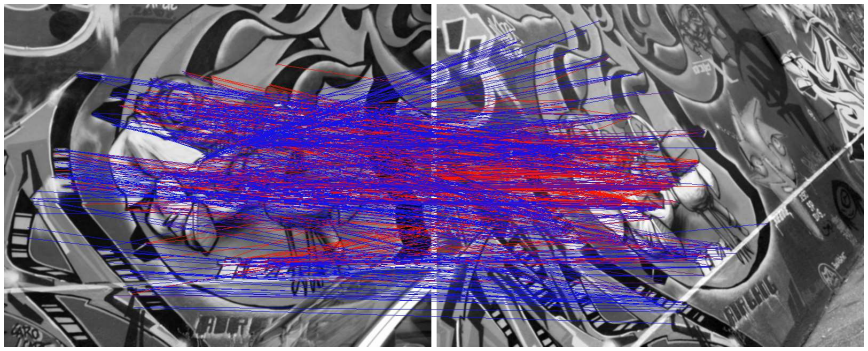
For image matching, increased field-of-views captured in increased image resolutions necessitates a quadratic increase in keypoints due to the two-dimensional nature of images. Our research introduces a novel approach, proposing an image matching method that circumvents the exhaustive NN search in



(a) Initial region matches



(b) Refined region matches



(c) Initial feature matches

(d) Refined feature matches

Figure 3.4. 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.

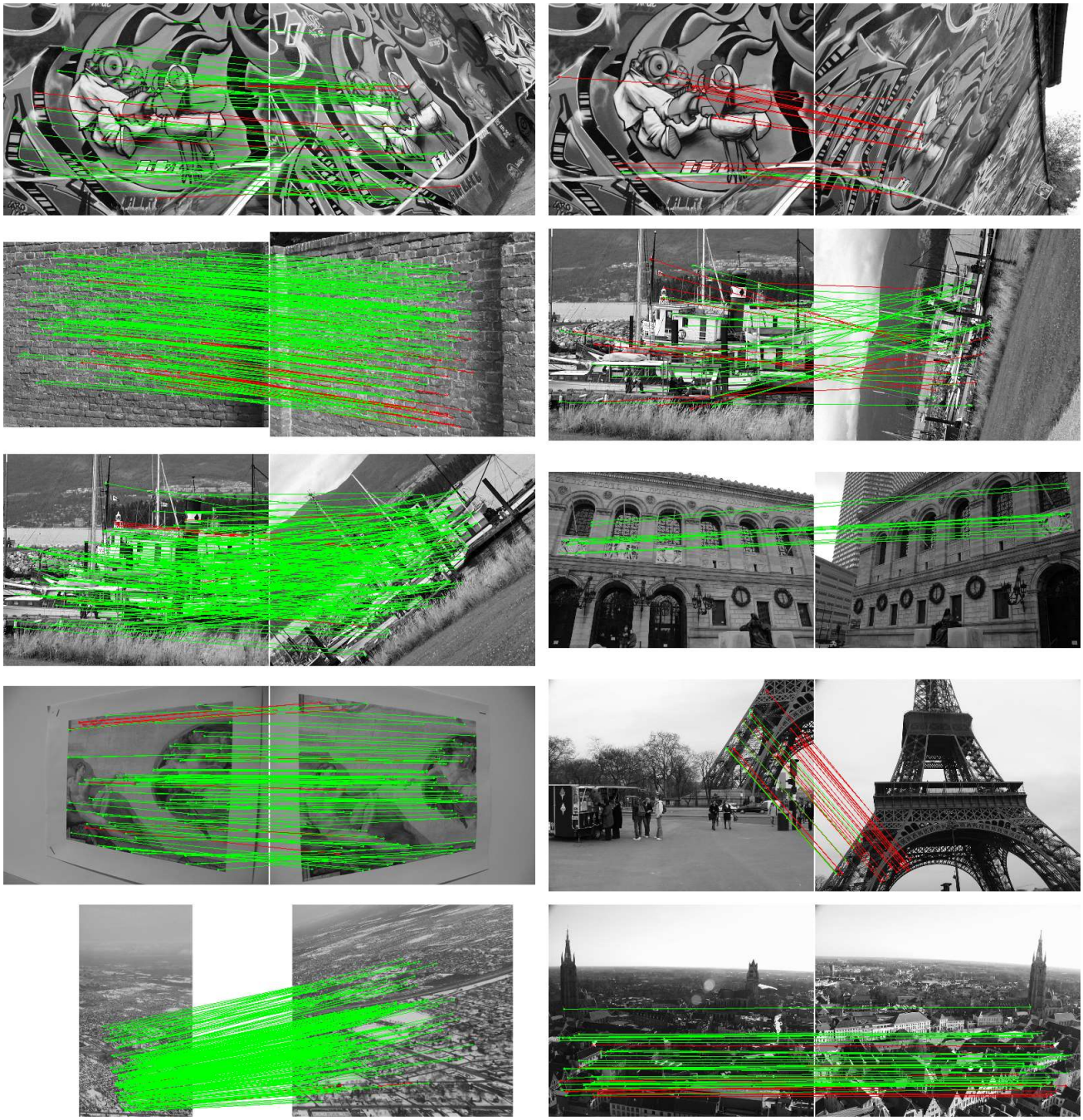


Figure 3.5. Sample Results From GGNN: More successful outcomes shown on the left and less successful ones on the right. Failures occur due to high reprojection errors in matching (indicated by red matches), or when the number of matches is not sufficiently high, or when matched features are not spatially well-distributed.

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, enhancing the efficiency of the method. Empirical evaluations support the effectiveness of this approach.

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 generates vectors with components less correlated than those in concatenated descriptors. This reduction in correlation enhances both discriminability and orthogonality, making our approach potentially more effective than the currently evaluated method for descriptor extraction.

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 extra computation is only performed for all images, not for all image pairs. Practically, even in cases where the feature count is low and overhead appears more 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 extra computational cost diminishes rapidly.

CHAPTER 4

EXTENSIONS AND IMPROVEMENTS

In the previous chapter, we proposed a hierarchical approach to image matching, detailing a method that utilizes this approach along with the necessary pre- and post-processing steps. We demonstrated its efficiency and accuracy in solving the problem of homography estimation across image pairs.

In this chapter, we extend and enhance our work in several independent dimensions. First, we generalize the number of regions (and thus spatial groups). Next, we explore an alternative to Group-Guided Nearest Neighbors (GGNN) to increase efficiency. We then propose a quick evaluation method for image matchers, useful for hyperparameter optimization. Additionally, we introduce a more accurate warping method compared to classical warping, which can generate more realistic image pairs from captured images. Finally, we extend our approach from planar to 3D scenes and evaluate the proposed method on pose estimation.

4.1. Pyramid of Grids for Hierarchical Regions

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.

The previous chapter focused exclusively on 64 groups, using $n = 4096$ and setting g to $\sqrt{n} = 64$. To vary the computational cost for a fixed n and to apply the previously proposed setting of $g = \sqrt{n}$ for

different n values, a generic formulation for generating regions is required.

4.1.1. Computation of Regions

Our image regions are arranged in grids, with these grids stacked as levels of a pyramid. Below is the formulation for computing the pyramid for a given number of regions. Each region is defined by the group of features within it. We use the terms "region count" and "group count" interchangeably, as they correspond one-to-one.

The function $\text{pyr}(g)$ is defined to compute a pyramid for a given number of groups g. This function constructs a set of integers that satisfy a series of specific conditions, ensuring the set is uniquely determined for each g. We define the function $\text{pyr}(g)$ as a piecewise function:

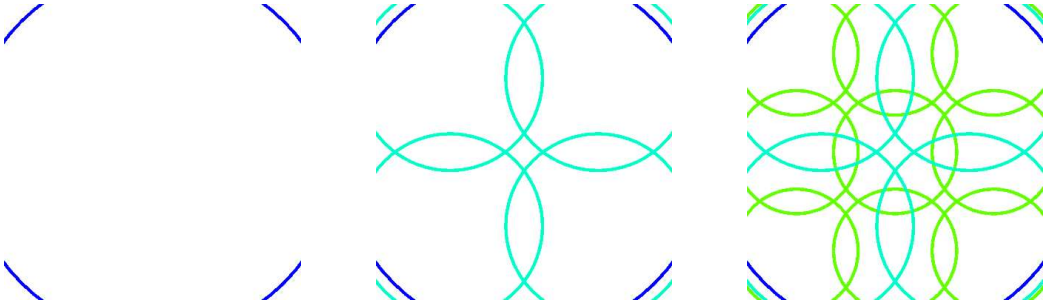
$$\text{pyr}(g) = \begin{cases} \{1\} & \text{if } g = 1 \\ \text{first}\left(\left\{ X \subset \mathbb{Z}^+ \;\middle|\; \begin{array}{l} \sum_{x \in X} x^2 = g \\ 1 \in X \\ \forall x_1 \in X \, \forall x_2 \in X \, (x_2 < x_1 \implies \frac{x_1}{x_2} > \sqrt{2}) \\ \forall x_1 \in X \left((\forall x_2 \in X \, (x_1 \leq x_2)) \vee (\exists x_2 \in X \, (x_2 < x_1 \wedge \frac{x_1}{x_2} < 2\sqrt{2})) \right) \\ \forall g' < g \, (|\text{pyr}(g')| \leq |X|) \end{array} \right\}\right) & \text{if } g > 1 \end{cases}$$

where first(sets) selects the lexicographically first element among the sorted sets.

For example, $\text{pyr}(1) = \{1\}$, $\text{pyr}(5) = \{1, 2\}$, $\text{pyr}(14) = \{1, 2, 3\}$ and $\text{pyr}(21) = \{1, 2, 4\}$. It holds that $\sum_{x \in \text{pyr}(g)} x^2 = g$ for all g if $\text{pyr}(g)$ is not an empty set.

The function first(sets) selects the set whose smallest number is the smallest among the sets. If there are multiple candidates, it selects the one whose second smallest number is the smallest among the candidates, and so on. The smallest number for which there are multiple possible pyramids is 247. $\text{pyr}(247)$ is equal to $\{1, 2, 3, 5, 8, 12\}$ rather than $\{1, 2, 3, 8, 13\}$, as $\text{first}(\{\{1, 2, 3, 5, 8, 12\}, \{1, 2, 3, 8, 13\}\}) = \{1, 2, 3, 5, 8, 12\}$ because $1 = 1$, $2 = 2$, and $3 = 3$, but $5 < 8$.

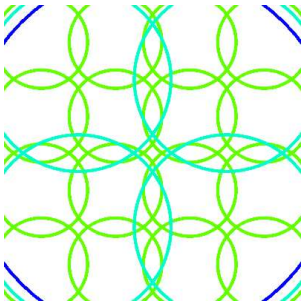
The semantics of these sets is that each level $x \in \text{pyr}(g)$ contains x^2 identical circles arranged in a grid with x rows and x columns. For instance, $\text{pyr}(14) = \{1, 2, 3\}$ indicates that there are 3 levels in the pyramid for 14 regions, with $1 \times 1 = 1$ circle, $2 \times 2 = 4$ circles, and $3 \times 3 = 9$ circles. If the value of the function is the empty set as in $\text{pyr}(2) = \{\}$, this means that such a pyramid cannot be constructed. Figures 4.1 and 4.2 collectively illustrate $\text{pyr}(g)$ where $1 \leq g \leq 200$.



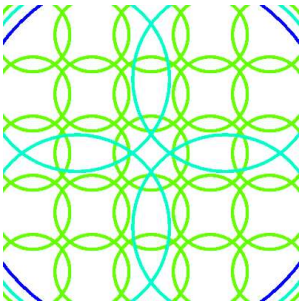
(a) $pyr(1) = \{1\}$

(b) $pyr(5) = \{1, 2\}$

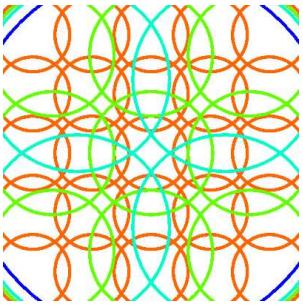
(c) $pyr(14) = \{1, 2, 3\}$



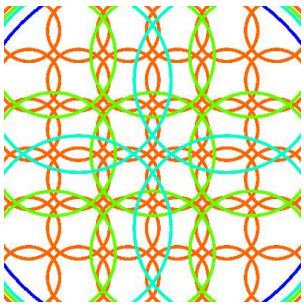
(d) $pyr(21) = \{1, 2, 4\}$



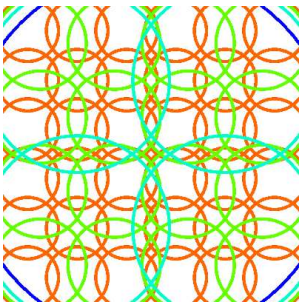
(e) $pyr(30) = \{1, 2, 5\}$



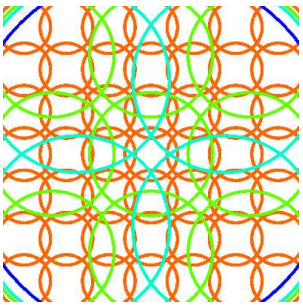
(f) $pyr(39) = \{1, 2, 3, 5\}$



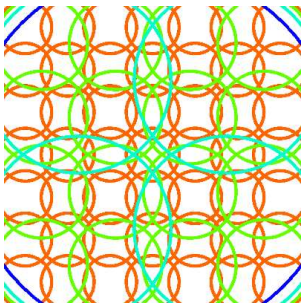
(g) $pyr(50) = \{1, 2, 3, 6\}$



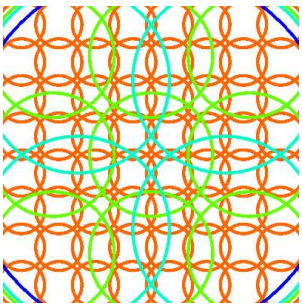
(h) $pyr(57) = \{1, 2, 4, 6\}$



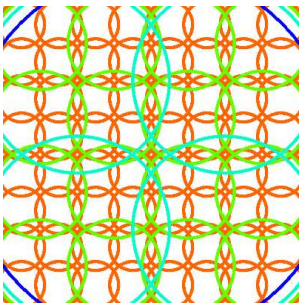
(i) $pyr(63) = \{1, 2, 3, 7\}$



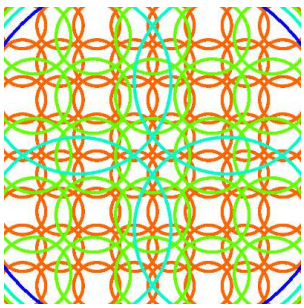
(j) $pyr(70) = \{1, 2, 4, 7\}$



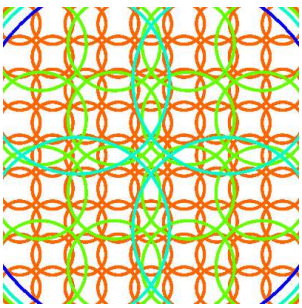
(k) $pyr(78) = \{1, 2, 3, 8\}$



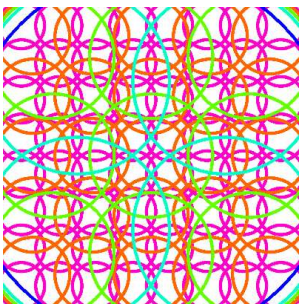
(l) $pyr(85) = \{1, 2, 4, 8\}$



(m) $pyr(94) = \{1, 2, 5, 8\}$

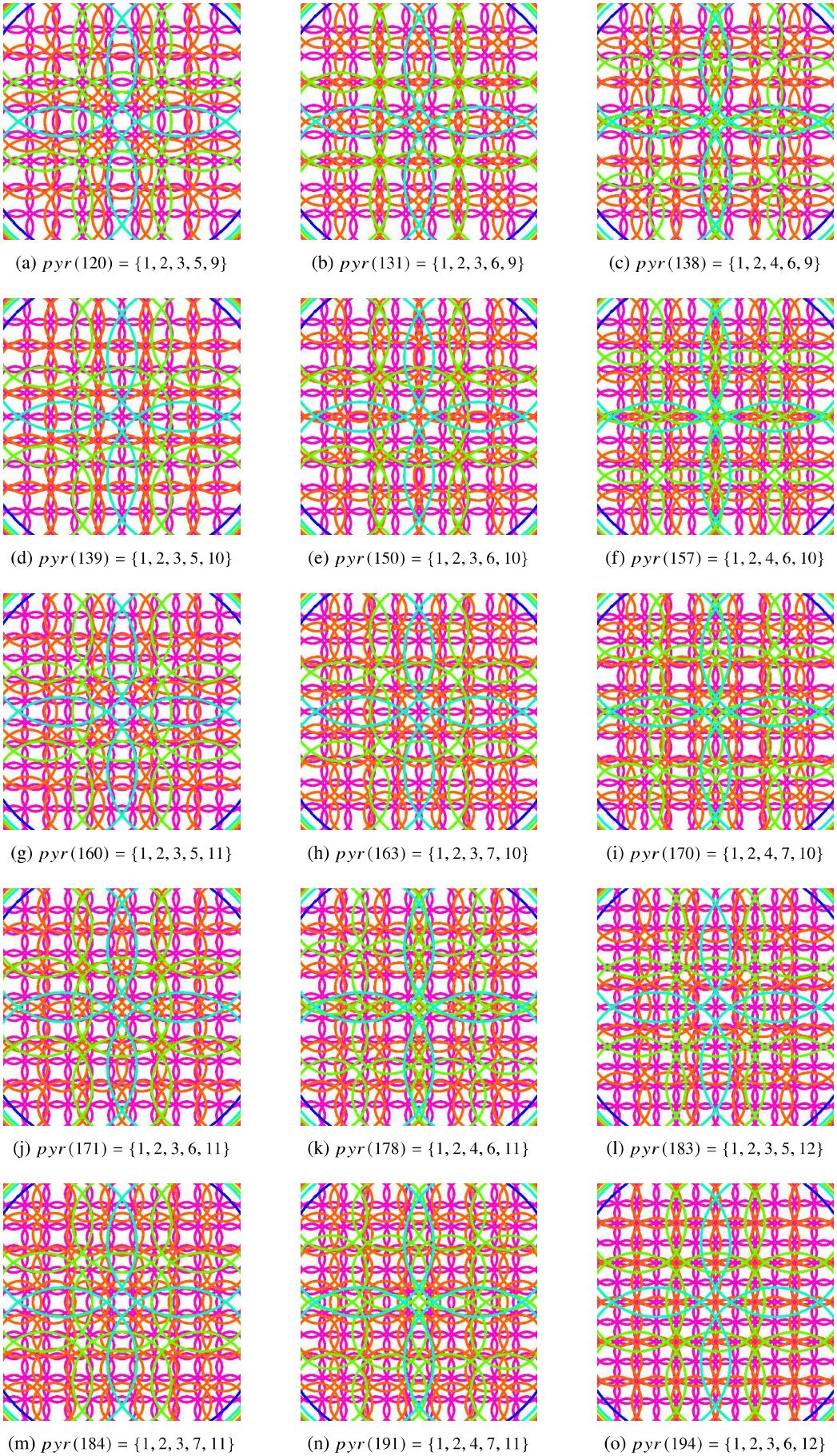


(n) $pyr(102) = \{1, 2, 4, 9\}$



(o) $pyr(103) = \{1, 2, 3, 5, 8\}$

Figure 4.1. Pyramid of Grids (Part I): $pyr(g)$ represents a pyramid with a total of g regions. Each level $x \in pyr(g)$ contains x^2 identical circles arranged in a grid with x rows and x columns.



(a) $pyr(120) = \{1, 2, 3, 5, 9\}$ (b) $pyr(131) = \{1, 2, 3, 6, 9\}$ (c) $pyr(138) = \{1, 2, 4, 6, 9\}$

(d) $pyr(139) = \{1, 2, 3, 5, 10\}$ (e) $pyr(150) = \{1, 2, 3, 6, 10\}$ (f) $pyr(157) = \{1, 2, 4, 6, 10\}$

(g) $pyr(160) = \{1, 2, 3, 5, 11\}$ (h) $pyr(163) = \{1, 2, 3, 7, 10\}$ (i) $pyr(170) = \{1, 2, 4, 7, 10\}$

(j) $pyr(171) = \{1, 2, 3, 6, 11\}$ (k) $pyr(178) = \{1, 2, 4, 6, 11\}$ (l) $pyr(183) = \{1, 2, 3, 5, 12\}$

(m) $pyr(184) = \{1, 2, 3, 7, 11\}$ (n) $pyr(191) = \{1, 2, 4, 7, 11\}$ (o) $pyr(194) = \{1, 2, 3, 6, 12\}$

Figure 4.2. Pyramid of Grids (Part II): $pyr(g)$ represents a pyramid with a total of g regions. Each level $x \in pyr(g)$ contains x^2 identical circles arranged in a grid with x rows and x columns.

Note that 1 is always an element, meaning that there must be a global region. For every pyramid, the ratio of two adjacent levels is a number between $\sqrt{2}$ and $2\sqrt{2}$. For example, $\text{pyr}(14) = \{1, 2, 3\}$ because $2/1$ is greater than $\sqrt{2}$ and less than $2\sqrt{2}$, and $3/2$ is greater than $\sqrt{2}$ and less than $2\sqrt{2}$. This constraint distributes regions well in scale space. The last condition ensures the number of levels is monotonically increasing as the number of regions increases.

Due to the recursive definition, we compute the values for all possible region numbers in ascending order up to an arbitrary upper bound. Once these values are computed, we implement the function for determining the regions using a simple lookup table.

For some region counts, the function is undefined. We believe the defined values are sufficiently dense, allowing the largest g value less than the required value, for which $pyr(g)$ is defined, to be used. However, if necessary, a pyramid with any number of regions can be created. To achieve this, identify the smallest g value greater than the required number for which $pyr(g)$ is defined. Then, remove the excess regions from the densest level, which is guaranteed to have more regions than required for removal. For example, $pyr(100)$ is not defined. Instead, one can use $pyr(94) = 1, 2, 5, 8$, or $pyr(102) = 1, 2, 4, 9$ and remove $102 - 94 = 8$ regions from the $9 \times 9 = 81$ grid.

4.1.2. Evaluation

In the experiments, we use the same pipeline as in the previous chapter. However, we omit region match elimination (refining region matches) and local geometric verification (refining feature matches) to ensure a fair comparison across varying numbers of regions. Fig. 4.3 shows the results obtained by varying the number of groups. We aim at minimizing both the number of comparisons and the failure percentage.

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 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. Efficient Image Matching with Group-Tested Nearest Neighbors

The GGNN approach to feature matching involves first matching groups of features and subsequently matching individual features within those matched groups. In this section, we propose an alternative

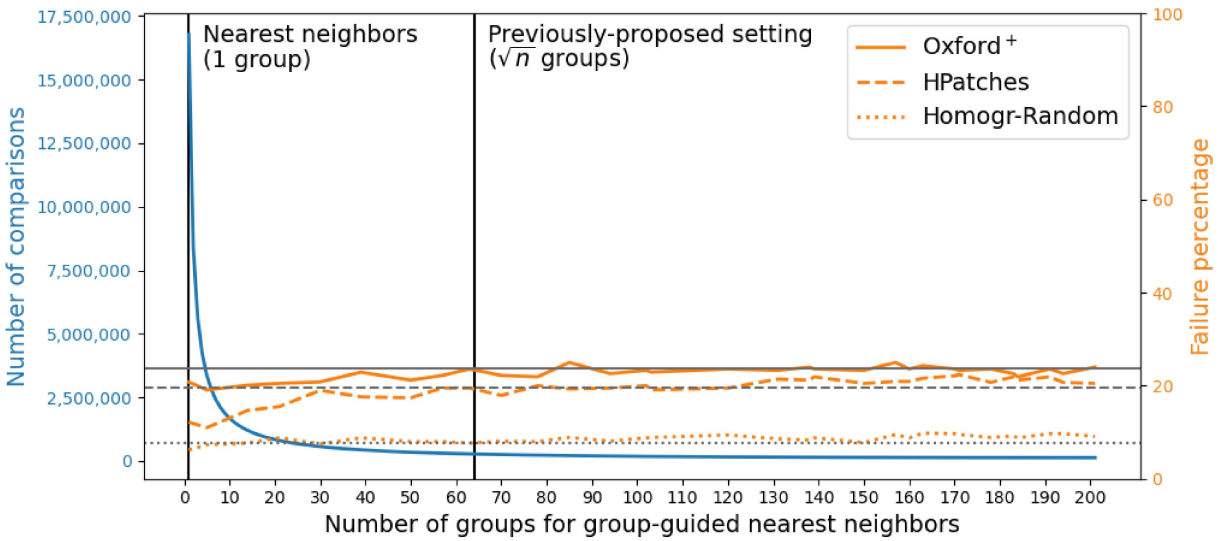


Figure 4.3. Impact of the Number of Groups: The plot visualizes the impact of different number of groups g while keeping their cardinality at $c = n/g$ for $n = 4096$. The proposed setting is calculated as $g = \sqrt{n} = 64$. Horizontal lines mark the performance level of the proposed setting.

approach called Group-Tested Nearest Neighbors (GTNN).

4.2.1. Group-Tested Nearest Neighbors

Group-Tested Nearest Neighbors (GTNN) is a two-step matching algorithm that parallels the GGNN methodology. However, it initially matches individual features to feature groups, and then matches these features to the members within the corresponding groups.

Assuming the existence of $\sqrt{n}$ groups, each containing $\sqrt{n}$ features—consistent with the proposed GGNN framework—the time complexity for matching all n features to feature groups of the other image is $\Theta(n\sqrt{n})$. This step yields n feature-group matches. For each of these matches, comparing a feature to the members of the matched group requires $\Theta(\sqrt{n})$ time, resulting in a total of $\Theta(n\sqrt{n})$. Thus, the overall complexity remains $\Theta(n\sqrt{n})$, identical to that of GGNN.

In scenarios where matching between features and groups is performed bidirectionally or multiple nearest neighbors are selected, the number of feature-group matches may exceed n. Conversely, if filtering techniques such as the ratio test or thresholding based on absolute distances are applied, the number of matches may decrease. Regardless of these variations, the asymptotic complexity does not increase.

Unlike GGNN, GTNN allows for a further reduction in time complexity to $\Theta(n)$ by considering only the top $\sqrt{n}$ features on the side of the individual features, even if the grouped features still consider all n features. This linear time complexity can also be achieved with NN when both sides are limited to the top $\sqrt{n}$ features, which are the features with the highest response. It is important to note that this approach is

different than detecting $\sqrt{n}$ features and matching them. We use efficient guided matching with all methods, allowing the use of all n features once the transformation is coarsely estimated.

Figure 4.4 illustrates and compares the linear time adaptation of the quadratic NN approach with the linear time version of GTNN. Unlike NN variations and GGNN, GTNN is asymmetrical in terms of the types of matched items. However, similar to the other methods, it can be performed bidirectionally, making the overall algorithm symmetrical.

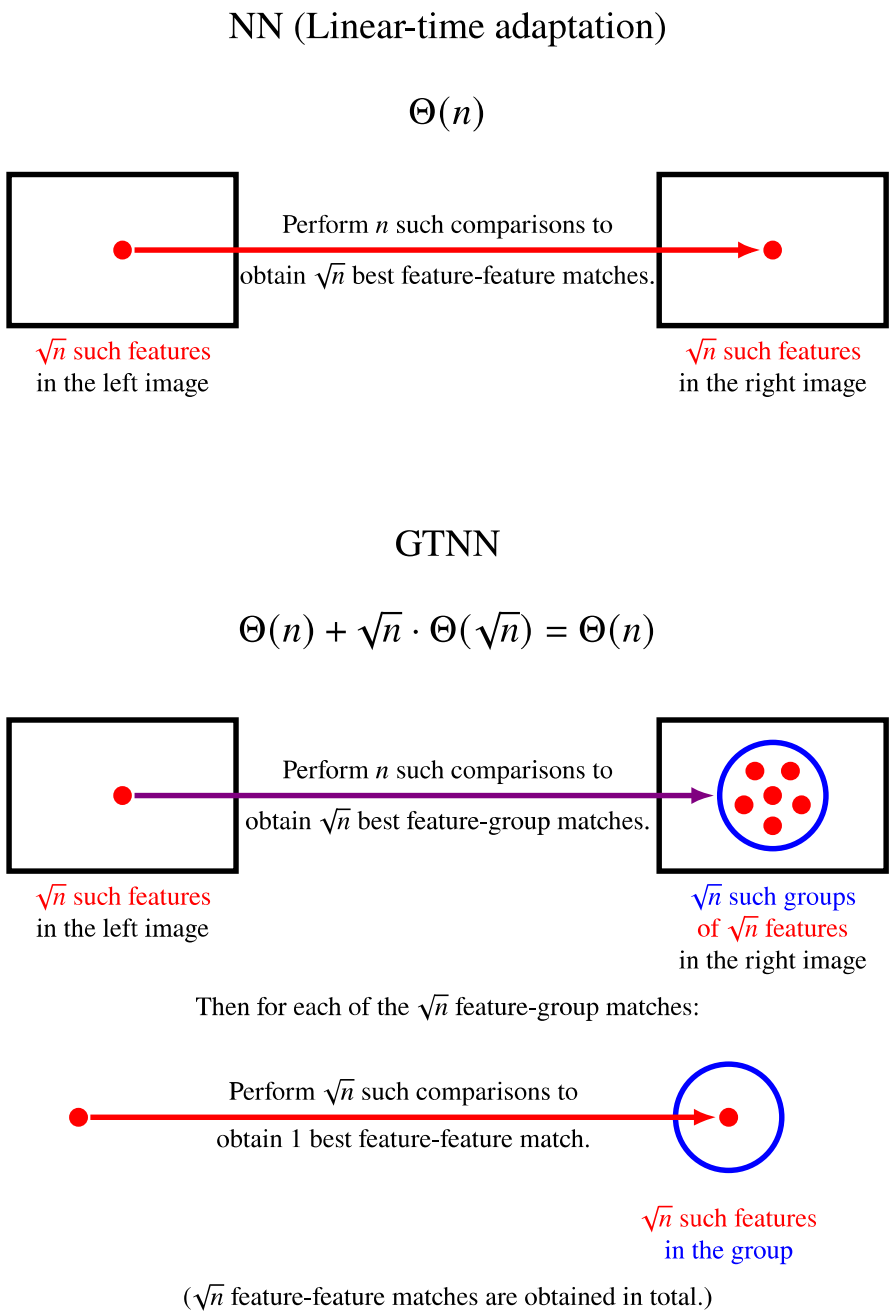


Figure 4.4. Feature Matching in $\Theta(n)$ Time: The GTNN algorithm differs from the NN algorithm by filtering features in the right image using group testing instead of relying on response scores. For each feature in the left image, GTNN dynamically determines a group of $\sqrt{n}$ candidate features from the right image for matching. In contrast, NN searches for a match from the fixed set of $\sqrt{n}$ top features in the right image for every feature in the left image.

4.2.2. Evaluation

We use the same datasets and pipeline as described in Chapter 3. However, we do not perform local geometric verification since it was defined for group-group matches, which are not applicable here. Due to the low number of matches, we utilize both the nearest neighbors and the second nearest neighbors. Although this approach significantly increases the rate of mismatches, the robust estimation can effectively manage them given the small number of matches.

We refer to the linear time adaptation of the NN algorithm, achieved by reducing the number of features to their square root, as Linear Nearest Neighbors (LNN). LNN employs the same preprocessing and postprocessing steps as GTNN. Other methods that filter the NN show worse performance in our experiments compared to the vanilla NN. Since the number of matches is already small, global geometric verification applied for robust estimation is sufficient and performs better without these prefiltering techniques. To save space, we omit these methods in the results and only present LNN as the baseline.

Table 4.1 shows the experiment results. The GTNN algorithm outperforms LNN on all three datasets, though the margins are small in two of them. Despite this, both linear time algorithms perform significantly worse than the other algorithms. The positive aspect is that we can almost always detect if we fail to estimate the correct transformation, even without knowing the ground truth. The number of consistent matches found indicates whether the result is likely due to pure chance. Therefore, GTNN can be used initially, especially if some image pairs are easier to match or if matching a subset of image pairs is sufficient, as in multiview cases. More complex algorithms can then be used only if needed.

Table 4.1. Homography Estimation Results of GTNN and Other Methods

Dataset	**Image Pairs**	**Failure Percentage**											
		NN	MNN	SNN (2004)	FGINN (2015)	AdaLAM (2020)	SMNN (2021)	HNN	**GGNN** (Proposed)	FLANN (2009)	HNSW (2018)	LNN	**GTNN** (Proposed)
Oxford⁺ Bikes	36	**0.0**	**0.0**	**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**	25.0	8.3
Oxford⁺ Leuven	36	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	13.9	2.8
Oxford⁺ UBC	36	**0.0**	**0.0**	**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	47.2	36.1
Oxford⁺ Wall	36	13.9	16.7	13.9	**11.1**	**11.1**	**11.1**	22.2	**11.1**	36.1	16.7	44.4	44.4
Oxford⁺ Bark	36	69.4	**63.9**	66.7	69.4	66.7	66.7	72.2	**63.9**	75.0	69.4	75.0	80.6
Oxford⁺ Boat	36	55.6	52.8	**44.4**	47.2	47.2	**44.4**	55.6	47.2	52.8	52.8	52.8	50.0
Oxford⁺	288	21.2	21.2	**18.8**	19.1	19.8	19.1	25.3	**18.8**	25.7	21.5	32.3	27.8
Homogr-Random	640	8.1	7.2	6.7	6.7	12.5	6.9	18.0	**6.6**	22.5	8.9	30.0	29.8
HPatches Illum	285	10.5	8.4	**7.4**	8.8	11.2	8.8	18.9	10.2	18.2	10.5	52.6	31.2
HPatches View	295	18.0	15.9	15.6	14.9	17.6	**14.6**	33.2	16.3	32.9	19.3	56.3	48.1
HPatches	580	14.3	12.2	**11.6**	11.9	14.5	11.7	26.2	13.3	25.7	15.0	54.5	39.8
Time Complexity		$\Theta(n^2)$						$\Theta(n\sqrt{n})$		$\Theta(n \log n)$		$\Theta(n)$	

Please note that there remains a $\Theta(n\sqrt{n})$ time complexity for computing exact spatial groups before

matching the features. This preprocessing step prevents the overall time complexity from reducing to linear time. However, the necessity for linear time complexity arises particularly in the multi-view scenario, where a set of images are matched against each other. In such cases, there can be $\Theta(m^2)$ potential image pairs for m images. Consequently, image-based computations are performed m times, while image pair-based computations are required $\Theta(m^2)$ times. Therefore, if the number of image pairs to match exceeds $m\sqrt{n}$, the amortized time complexity, including the preprocessing of feature groups, remains linear. Conversely, if this condition is not met and a reduction in complexity is still desired, approximate grouping methods such as locality-sensitive hashing (Indyk and Motwani 1998; Gionis, Indyk, Motwani, et al. 1999; Jafari et al. 2021) or grid-based methods can be employed to achieve linear time complexity.

4.3. Quick Evaluation of Image Matching Pipelines

We observe that the performance of an image matching pipeline can be efficiently predicted without running it on the entire dataset. For instance, the pipeline can be initially tested on a medium-difficulty problem. If it succeeds, it can then be applied to a more challenging problem; if it fails, it can be applied to an easier problem instead. This approach is particularly useful when pipelines have alternative modules and multiple parameters, leading to a combinatorial explosion. In such cases, quick evaluation is essential for hyperparameter optimization on a validation set.

To address the problem of quick evaluation, we propose using a decision tree regressor (Kingsford and Salzberg 2008; Loh 2011, 2014; Breiman 2017). Training data for these quick evaluation models can be generated by running random pipelines on the full validation set, making this a meta-learning problem where the training occurs on the validation set. The resulting tabular data consists of rows representing different pipelines (image matchers), columns representing individual problems (image pairs to match), and values indicating how each pipeline performs on each problem.

Image matching algorithms generate a numeric error instead of a simple success or failure, enabling more precise performance predictions. Consequently, the values in the table are numeric, representing the Relative Average Corner Error (RACE), which is the average corner error divided by the image's diagonal length. It is important to note that if a matcher completely fails to predict a transformation, the error is infinite, and thus the table may contain infinite values. The final column in the table represents the failure percentage across the entire dataset, based on an arbitrary failure tolerance as detailed in previous chapters. For this study, we consider values above 0.01 as failures. Figure 4.5 illustrates the dataset.

	Image Pair 1	Image Pair 2	⋯	Image Pair j	Overall
Matcher 1	$RACE_{1,1}$	$RACE_{1,2}$	⋯	$RACE_{1,j}$	$percentage_1$
Matcher 2	$RACE_{2,1}$	$RACE_{2,2}$	⋯	$RACE_{2,j}$	$percentage_2$
⋮	⋮	⋮	⋱	⋮	⋮
Matcher i	$RACE_{i,1}$	$RACE_{i,2}$	⋯	$RACE_{i,j}$	$percentage_i$

Figure 4.5. Illustration of the dataset for learning and testing quick evaluation: The objective is to predict overall performance, indicated by the failure percentage (last column, ranging from 0 to 100), using performance data from selected problems (other columns, represented by the relative average corner error, ranging from 0 to ∞). This method enables estimation of the failure percentage without calculating all performance values. The failure percentage is determined based on an arbitrary failure tolerance. The training-testing split must be performed on the rows. This is a meta-learning problem because the components of the matchers are also typically being learned.

4.3.1. Methods for Learning Quick Evaluation

Classification and Regression Trees (CART) (Breimann et al. 1984) is a widely used decision tree learning algorithm. Existing implementations, such as Scikit-Learn's (Pedregosa et al. 2011) 'DecisionTreeRegressor', cannot handle infinite values. To address this, we implement a decision tree regressor from scratch that minimizes mean squared error (MSE) at each split. We verify its correctness by running tests both on the custom implementation and the Scikit-Learn implementation, and comparing the results. We then modify the implementation to change the operator for the split condition from '$\leq$' to '$<$'. This simple modification enables the algorithm to handle data containing positive infinities. The rationale is that the midpoint between a finite number and positive infinity is calculated as infinity, and the '$<$' operator can then split the data into finite numbers and infinities when the threshold is set to ∞.

We introduce another important modification to the learning algorithm by adding a monotonicity constraint (Potharst and Feelders 2002) for splitting the tree: the predictions must be monotonically increasing from the left-most leaf to the right-most leaf. This is because it is never indicative of superior performance when an algorithm performs worse on an individual problem, even if it contradicts part of the training data considered at the time. This means that the rule 'if $feature < threshold$ then predict x, otherwise predict y' can be added to the tree only if $x < y$. This constraint helps to avoid overfitting. We refer to the resulting decision tree as a Monotone Decision Tree (MDT). Figure 4.6 shows a decision tree learned on Oxford$^+$ homography estimation dataset.

Note that the runtime for condition checks is negligible compared to running a pipeline on an image pair. Nearly all the time needed for inference is spent calculating the values of the required features. As a result, reusing the features in a path from the root to a leaf does not increase the runtime of the quick evaluation. Therefore, we propose an improvement to MDTs for this problem: We increase the "maximum

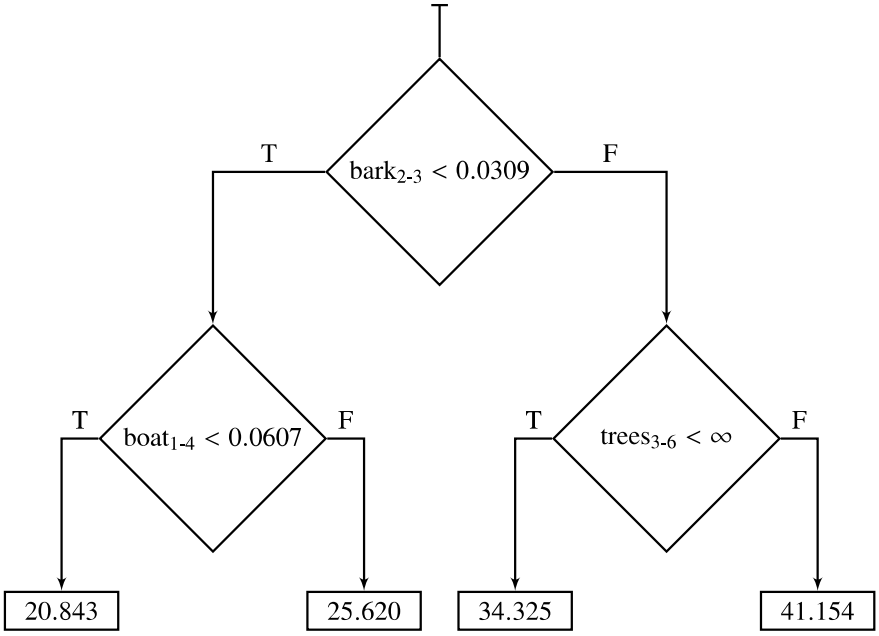


Figure 4.6. Example decision tree for quick evaluation: This tree predicts an image matcher's performance on the Oxford$^+$ dataset, which contains 288 image pairs, based on evaluations of only two image pairs. Although the model considers three features in total, the number of required evaluations equals the depth of the tree, which is 2, assuming lazy evaluation (i.e., features are computed only when needed). For instance, if the pipeline in question has a RACE of 0.01 when estimating the homography matrix between the 2nd and the 3rd images of the 'bark' scene, and has a RACE of 0.1 when estimating the homography matrix between the 1st and the 4th images of the 'boat' scene, the model predicts its overall failure percentage as 25.62%. Note that the tree is monotone, with predictions in ascending order from left to right.

depth" but constrain the tree to perform splits solely based on the features that are already used in that path after reaching the original maximum depth. We refer to the resulting decision tree as a Feature-Efficient Monotone Decision Tree (FEMDT). Similar ideas related to learning based on costly features can be found in the machine learning literature (Reyzin 2011; Xu et al. 2014; Peter et al. 2017; Janisch, Pevnỳ, and Lisỳ 2019; Erion et al. 2022).

Decision forests (Rokach 2016) consist of a set of decision trees whose predictions are averaged (or determined by majority voting in the case of classification). Like other model ensembles, which are generally more robust than individual models, decision forests are generally more robust than individual decision trees. Therefore, we propose using a feature-efficient variant of random forests (Biau and Scornet 2016; Parmar, Katariya, and Patel 2019) for quick evaluation. Since decision trees can be used for feature selection, we first build a decision tree to obtain feature importances, then use the most important features. The number of selected features correlates with the inference time budget. Increasing the number of trees in the ensemble or their maximum depths does not increase the inference runtime beyond the time needed for calculating all the selected features. We refer to this type of random forest as a Feature-Efficient Random Forest (FERF). Note that these are regular random forests but trained after feature selection.

4.3.2. Evaluation

Table 4.2 shows the most important features for quick evaluation of Oxford$^+$ dataset. These values were calculated using 500 random image matching pipelines obtained with random modules (such as a feature matching algorithm) and random parameters (such as a threshold). A feature's importance is calculated based on its contribution to the model, considering that more important features are already in use. This diminishes the returns with each additional feature, which explains the significant difference between the first and second highest importance values, and not necessarily their independent predictive powers.

Table 4.2. Feature Importances for Quick Evaluation

	1	**2**	**3**	**4**	**5**	**6**	**7**
Feature	$\text{bark}_{2\text{-}3}$	$\text{boat}_{1\text{-}4}$	$\text{wall}_{4\text{-}2}$	$\text{graff}_{4\text{-}6}$	$\text{wall}_{5\text{-}2}$	$\text{boat}_{4\text{-}5}$	$\text{leuven}_{3\text{-}6}$
Importance	0.737	0.141	0.040	0.016	0.016	0.006	0.006

Table 4.3 presents a comparison of the learning algorithms tested using 10-times repeated 10-fold cross-validation. Notably, even the simplest solution, MDT, predicts the overall performance of an image matching pipeline at one or two orders of magnitude faster than exhaustive evaluation, with relatively small error. For example, MDT can predict the failure percentage approximately 100 times faster with an average error of about 1.4%. The feature-efficient variant, FEMDT, further improves performance over the standard version. We show two versions: FEMDT_1, which adds a single cost-free level at the bottom of MDT, and FEMDT_2, which adds two such levels. While more levels can be added, the models will overfit after a certain number of levels. The ensemble variant, FERF, with 3 decision trees, has three versions: FERF_1 and FERF_2, with one and two extra cost-free levels respectively, and FERF∞, with unbounded depth. FERF∞ outperforms all other methods at every cost level.

Table 4.3. Prediction Performance of Quick Evaluation Models

Number of Evaluations	**Approximate Speedup**	**Root Mean Squared Error**						
		MDT	**FEMDT_1**	**FEMDT_2**	**FERF_0**	**FERF_1**	**FERF_2**	**FERF_∞**
1	288×	2.992	2.719	2.703	2.951	2.040	1.530	**0.874**
2	144×	1.804	1.534	1.510	2.028	1.533	1.390	**0.880**
3	96×	1.420	1.342	1.296	1.541	1.389	1.285	**0.899**
4	72×	1.266	1.205	1.385	1.385	1.281	1.148	**0.921**
5	58×	1.167	1.145	1.132	1.274	1.134	1.077	**0.888**
6	48×	1.129	1.101	1.084	1.157	1.085	1.040	**0.900**
7	41×	1.081	1.072	1.066	1.076	1.082	1.065	**0.899**

Unlike other methods, with FERF_∞, the error does not decrease as the number of evaluations increases. Interestingly, the best performing model is also the fastest, suggesting that, at least for Oxford$^+$,

testing a pipeline on a single image pair effectively reveals its quality. It is important to note that these results may vary not only with different image pair datasets but also with different pipelines. Figure 4.7 shows the histogram of the target variable values for the generated image matching pipelines with the selected image pair dataset. When the distribution changes, the results are expected to change. However, the same methodology can be applied to any image pair dataset and any set of pipelines.

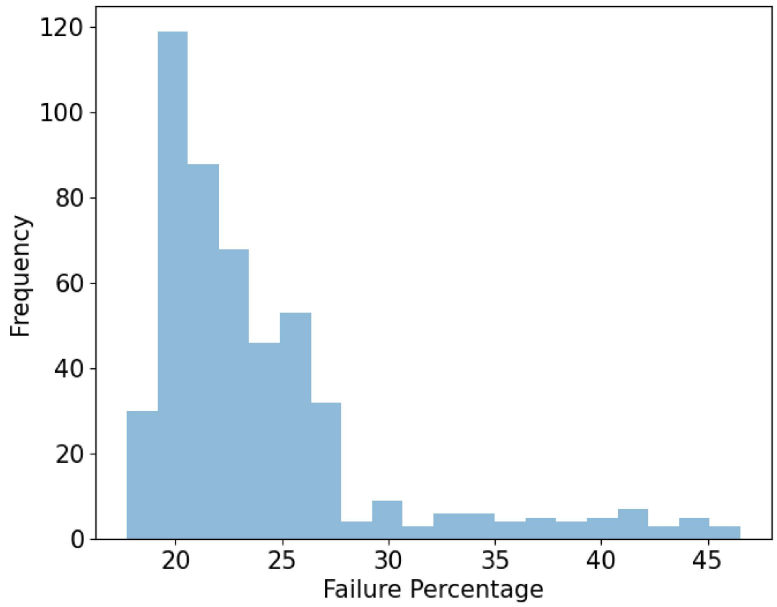


Figure 4.7. Distribution of failure percentages in the quick evaluation dataset: Among the 500 pipelines, the mean failure percentage is 24.1, with a standard deviation of 5.6. The failure percentages range from a minimum of 17.7 to a maximum of 46.5. A simple model that always predicts the mean value of 24.1 would result in an RMSE of 5.6.

In the previous chapter, we demonstrated that GGNN outperformed all other methods, including less efficient ones, achieving a failure rate of 18.8% on the Oxford$^+$ dataset. By employing the quick evaluation model FERF_∞ with single evaluation, we conducted hyperparameter optimization through approximate evaluations for randomly selected parameters. This process was completed in significantly less time than required to evaluate a single pipeline on the full dataset. Using the optimized parameters identified through this approach, we further reduced the failure rate to 17.7%, as confirmed by a full dataset evaluation.

4.4. Accurate Image Warping for Improved Dataset Generation

Image warping functions are essential components of any image processing toolbox. These functions apply geometric transformations, such as projective transformations, to images. In digital images, intensity values are located on discrete coordinates. When a pixel is shifted by a non-integer amount,

interpolation is required, resulting in an approximation. These approximations typically smooth the image, causing high-frequency regions such as edges, corners, and textures to become blurred. The loss of information in the image warping process becomes evident when the resultant image differs from the original after a complete cycle of periodic transformations (e.g., applying a 30-degree rotation 12 times) or the sequential application of forward and inverse transformations (e.g., downscaling by a factor of 2 followed by upscaling by a factor of 2).

Super Resolution (SR) encompasses a class of techniques that generate a high-resolution image from one or more low-resolution images. It is trivial and virtually free to generate large datasets of low- and high-resolution versions of images, simply by downscaling any unsupervised image dataset. Thanks to this, many deep learning-based single-image SR models have been proposed. The first proposed model (Dong et al. 2015) applied image restoration to traditionally upscaled images. Then many specialized neural network architectures (Xintao Wang et al. 2018; Y. Wang et al. 2018; Xintao Wang et al. 2021; Zhang et al. 2021) were specifically designed for upscaling images accurately, and repeatedly pushed the state-of-the-art further.

In this section, we propose a solution to the problem of image warping by using data-driven super resolution (SR) as a black box. As far as we know, this application of SR has not been previously documented in the literature. We employ this image warping method to realistically generate synthetic pairs of single captured images for a homography estimation dataset.

4.4.1. A Method for Accurate Image Warping

Interpolation for intensity values is almost always required for image warping. The exceptions are specific cases such as combinations of integer translations, rotations by multiples of $\pi/2$, and downscaling along an axis by an odd integer factor. Consequently, image warping almost invariably leads to information loss.

No free lunch theorems state that the average performance of optimization algorithms across all possible problems is equivalent (Wolpert and Macready 1997). This implies that among single-image SR techniques (e.g., deep learning-based models (Dong et al. 2015; Xintao Wang et al. 2018; Y. Wang et al. 2018; Xintao Wang et al. 2021; Zhang et al. 2021)), non-adaptive interpolation methods (e.g., bilinear interpolation), and adaptive interpolation methods (e.g., edge-directed interpolation (Allebach and Wong 1996; Li and Orchard 2001; Tam, Kok, and Siu 2010)), none is inherently superior for randomly synthesized images. Even a simple constant function used for interpolation is equally successful on average. Fortunately, the laws of physics impose local consistency and other properties on real-world objects, making

some techniques superior for real images (Lin, Tegmark, and Rolnick 2017). Deep learning-based SR techniques leverage extensive datasets of example images to learn natural priors and develop awareness of both local and global patterns. These techniques generally produce better approximations than other upscaling methods, except in the case of intentionally designed adversarial examples.

Warped images can be used as data augmentation for training neural networks, including SR models. We propose the opposite: using SR models to warp images, resulting in more accurate estimations than traditional methods such as bicubic interpolation. This can be achieved by first applying SR to the image, then warping the super-resolved image using the corresponding high-resolution transformation, and finally downscaling the result.

For any uniform upscaling matrix U and a transformation matrix T (such as a homography matrix), there is another transformation matrix T' (also homography matrix if T is a homography matrix), such that $Tp = U^{-1}T'Up$ for all 2-dimensional points p represented in homogeneous coordinates. In fact, $T' = UTU^{-1}$. This means that applying $U^{-1}T'U$ to an image coordinate is equivalent to directly applying T. However, the former approach can leverage SR algorithms for upscaling the image. The upscaled image essentially corresponds to denser intensity values in the original resolution. Denser values lead to better interpolation, assuming the upscaling is more accurate than the simple interpolation and the downscaling step at the end does not negate the accuracy gains achieved at the high resolution. Figure 4.8 illustrates both the direct approach and the proposed approach to the image warping problem, along with the ideal result.

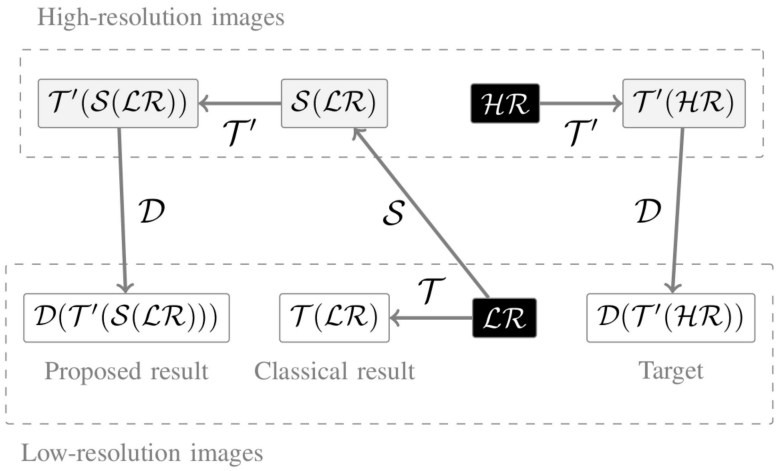


Figure 4.8. Accurate Image Warping: The original warping function ($\mathcal{T}$) required to transform the original image, the calculated function ($\mathcal{T}'$) that mimics $\mathcal{T}$ in high resolution, the super resolution algorithm ($\mathcal{S}$) that accurately upscales images, and the downscaling operation ($\mathcal{D}$) that conventionally downscales images are all represented as functions. $\mathcal{LR}$ represents the original image, and $\mathcal{HR}$ represents the hypothetical perfect high-resolution version of $\mathcal{LR}$. In this context, the target represents the ideal warped image. The classical result is the outcome of direct warping, while the proposed result is a better approximation of the target than the classical result, assuming $\mathcal{S}$ and $\mathcal{D}$ produce accurate results. The target is typically unknown, as it is not possible to calculate $\mathcal{HR}$ from $\mathcal{LR}$. However, it is possible to use a given image as $\mathcal{HR}$ and obtain $\mathcal{LR}$ by downscaling it. This allows us to confirm that the similarity between the proposed result and the target is higher than that between the classical result and the target.

For tasks like homography estimation, it is common practice to train models and evaluate algorithms on large datasets of artificially transformed real images. In the previous chapter, we generated such a dataset using images from the Homogr dataset. The proposed image warping method can be employed to generate these datasets in a more realistic manner. While recent deep warping models (Son and Lee 2021; Xiao et al. 2024) offer an alternative approach, our method has the advantage of requiring each image to be super-resolved only once, regardless of the number of different transformations applied. This makes our approach highly efficient for generating a vast number of image pairs, potentially even an infinite number, from single high-resolution images.

4.4.2. Evaluation

We generated a dataset of 320 image pairs using the Homogr images. As the baseline warping method, we employed bicubic interpolation. For the proposed warping approach, we used BSRGAN (Zhang et al. 2021) for super resolution with a scale factor of 2, followed by bicubic interpolation for warping in high resolution and subsequent downscaling.

Another challenge with image warping for synthetic dataset generation, beyond interpolation issues, is that image resolutions significantly decrease because the warped images must be cropped to the largest inscribed rectangles. These rectangles are considerably smaller than the original images, particularly when the transformations are more extreme. For this reason, it is usually sensible to upscale the warped images and use them as the image pair dataset. We follow this approach as well. We upscale the resultant images for the bicubic warping method and omit the downscaling step for the proposed warping method.

Figure 4.9 shows sample patches from warped images. The proposed method's results appear significantly sharper based on qualitative visual inspection.

Table 4.4 shows the performance of GGNN on warped images using both methods, as well as on the high-resolution versions. It is important to note that the results for the low-resolution and high-resolution datasets are not directly comparable, as high-resolution datasets present easier problems.

Table 4.4. Homography Estimation Results on Warped Images

	Bicubic Warping	**Proposed Warping**	**Bicubic Warping** (2×)	**Proposed Warping** (2×)
Failure Percentage	28.4	**26.9**	7.2	**2.5**

The results suggest that the proposed method is a better simulator of camera movement in the real world compared to the widely used method in the community. This approach can potentially be applied to other image warping applications as well.

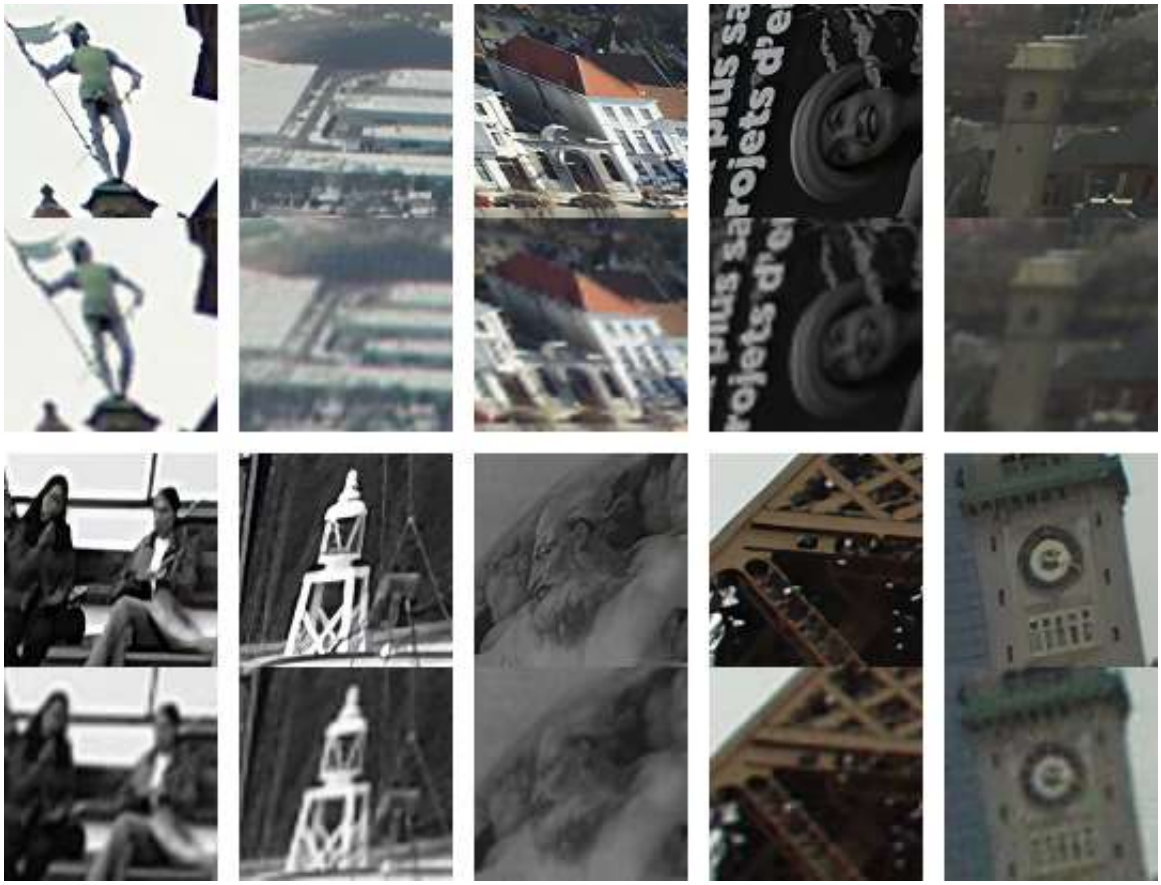


Figure 4.9. Sample Patches from Warped Images: The image patches above demonstrate the results of the proposed warping method, whereas the image patches below show the results of the bicubic warping method.

4.5. Generalization to 3D Scenes

In this section, we extend the application of our hierarchical image matching method from planar scenes to three-dimensional (3D) scenes, leveraging the principles of epipolar geometry. While in planar scenes or when objects are sufficiently far from the camera, estimating a homography matrix remains meaningful, general 3D cases require a more complex approach. By utilizing the concepts of the essential matrix and epipolar geometry, we aim to demonstrate that our method can be effectively generalized to handle 3D scenes, providing accurate pose estimation and robust matching performance.

4.5.1. Pose Estimation with Hierarchical Image Matching

In three-dimensional scenes, pose estimation becomes crucial for understanding the spatial relationships and relative positioning of objects. To achieve this, we employ the concepts of epipolar geometry, which are fundamental to 3D computer vision.

Figure 4.10 illustrates epipolar geometry. If the camera intrinsics are known, the relative pose (position and orientation) of the two cameras can be represented by an essential matrix; otherwise, it is represented by a fundamental matrix. However, there are degenerate cases where the essential or fundamental matrix cannot be accurately estimated, such as when the scene is planar, when there is pure rotation without translation, when the cameras move parallel to each other, or when one camera is directly behind the other.

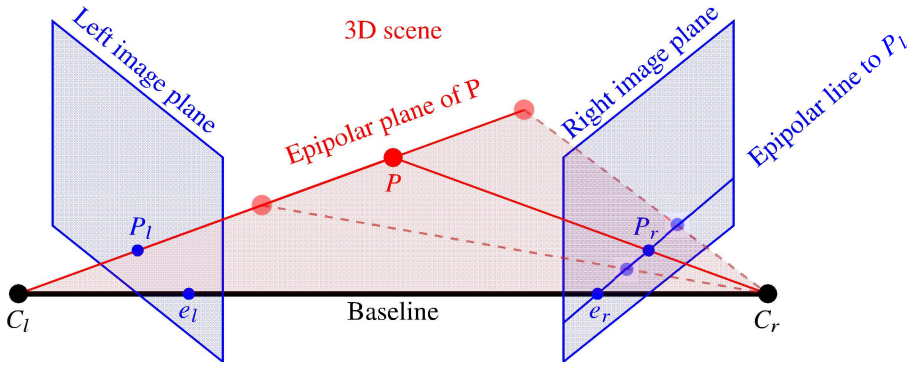


Figure 4.10. Epipolar Geometry: C_l and C_r are the centers of two cameras (or a single camera at two different times). The baseline is the line that connects these two camera centers. Terms like "wide-baseline stereo" refer to the length of the segment $\overline{C_l C_r}$, indicating that the distance between C_l and C_r is relatively long. Image planes are the 2D projections of the scene as seen by the cameras. Although these planes are theoretically infinite, the cameras capture only a finite portion of the scene within their field of view. The epipoles e_l and e_r are the projections of each camera center onto the image plane of the other camera and they lie on the baseline. Depending on the field of view, the epipoles can be outside the actual image. P is an arbitrary 3-dimensional point in the scene. An epipolar plane is defined by a 3D point and the optical centers of the two cameras. The projection of P on the left image plane is P_l and on the right image plane is P_r. If we know the relative pose of the cameras, and that P_l and P_r correspond to the same point, we can locate the 3D point P, as the lines $\overline{C_l P_l}$ and $\overline{C_r P_r}$ intersect at P. In practice, due to localization errors, the lines may not actually intersect, but an approximate intersection can be calculated. Epipolar lines are the lines on an image plane that pass through the epipole and all possible locations of a point. For instance, the epipolar line corresponding to P_l lies on the right image plane and passes through the epipole e_r and the projected point P_r. If we know the relative pose of the cameras and want to locate P_r, the corresponding point to P_l in the right image, it can be anywhere on the corresponding epipolar line. This uncertainty arises because the depth (distance of P to C_l) is not known. If the depth is known, the position of P_r can be determined.

For general 3D scenes, the problem of pose estimation involves determining the essential or fundamental matrix based on whether the camera intrinsics are known. This process is analogous to homography estimation in planar scenes, relying on robust estimation techniques using feature matches. The essential matrix encodes the rotation and translation between two camera views, while the fundamental matrix represents this relationship when camera calibration is unavailable.

To estimate the essential or fundamental matrix, we utilize a robust estimation method such as Random Sample Consensus (RANSAC) to identify a consistent set of feature matches. This step ensures that outliers, which can significantly affect the accuracy of the estimated matrix, are excluded from the calculation. Once the matrix is estimated, the relative pose of the cameras, including their rotation and translation, can be derived.

Guided matching in the context of 3D scenes follows a similar principle to the planar case but is adapted to work with epipolar constraints. Instead of searching for corresponding points around a given point, we search along the epipolar lines. This approach leverages the geometric relationship defined by the essential or fundamental matrix, where each point in one image has a corresponding epipolar line in the other image.

4.5.2. Evaluation

To evaluate our method, we utilize the Tanks and Temples dataset (Knapitsch et al. 2017), a benchmark specifically designed for large-scale scene reconstruction. This dataset provides a comprehensive array of challenging indoor and outdoor scenes captured under realistic conditions using high-resolution video. Ground-truth data, obtained using an industrial laser scanner, ensures high accuracy and enables detailed performance assessment of 3D reconstruction methods.

Although the dataset is primarily intended for multiview matching, we sample image pairs from the image sequences for our pairwise image matching evaluation. We select three scenes for this purpose: Meeting room, Truck, and Barn. These scenes offer a variety of environments: the Meeting room is an indoor scene with varying lighting conditions, the Truck represents a complex outdoor vehicle scene with reflective surfaces, and the Barn is an outdoor scene with repetitive patterns and a mix of natural and man-made textures. Figure 4.11 provides a visual reference for the scenes.

(a) Meeting Room (b) Truck (c) Barn

Figure 4.11. Selected Scenes: (a) Meeting Room, (b) Truck, (c) Barn.

We generate three image pair datasets by matching every 10th image to 18 images around it, filtering out those with an angular distance greater than 60 degrees. This results in the following number of image pairs: Meeting room: 570, Truck: 416, and Barn: 694.

To evaluate the quality of the estimated poses, we use the mean Average Accuracy (mAA) metric (Yi et al. 2018; Jin et al. 2021). mAA measures the angular accuracy of the estimated poses by computing the

difference between the estimated and ground truth translation vectors and rotation vectors. For simplicity, we only consider the rotation vector in our evaluation. The mAA is calculated by thresholding the angular error at multiple levels and integrating the resulting accuracy values to provide a single scalar score.

We perform experiments using the same methods described in Chapter 3. Figure 4.12 presents the pose estimation results, showing average accuracy versus angular error. Numerical results, specifically mean Average Accuracy at 5 and 10 degrees, are summarized in Table 4.5.

Table 4.5. Pose Estimation Results

Dataset	**Image Pairs**	**Threshold**	**mean Average Accuracy (mAA)**									
			NN	MNN	SNN (2004)	FGINN (2015)	AdaLAM (2020)	SMNN (2021)	HNN	**GGNN (Proposed)**	FLANN (2009)	HNSW (2018)
Meeting room	570	5°	.209	.254	.250	.254	.222	**.271**	.181	.250	.148	.204
		10°	.360	.415	.408	.415	.368	**.437**	.302	.398	.270	.352
Truck	416	5°	.169	.216	.215	.213	.204	**.232**	.139	.209	.121	.168
		10°	.279	.342	.350	.354	.325	**.372**	.226	.332	.204	.279
Barn	694	5°	.271	.332	.325	.306	.299	**.349**	.252	.304	.225	.269
		10°	.402	.465	.460	.443	.432	**.490**	.368	.428	.339	.399
Time Complexity			$\Theta(n^2)$						$\Theta(n\sqrt{n})$		$\Theta(n \log n)$	

The results extend the main conclusions we drew in Chapter 3 to epipolar geometry: While SMNN consistently outperforms all other methods across every dataset, GGNN demonstrates superior performance compared to other methods with the same or greater efficiency. It also surpasses some of the less efficient methods. In our experiments, GGNN consistently outperforms NN and achieves better results than AdaLAM in most cases. These findings highlight the robustness and versatility of GGNN in various challenging scenarios, reinforcing its potential for practical applications in 3D reconstruction and other computer vision tasks.

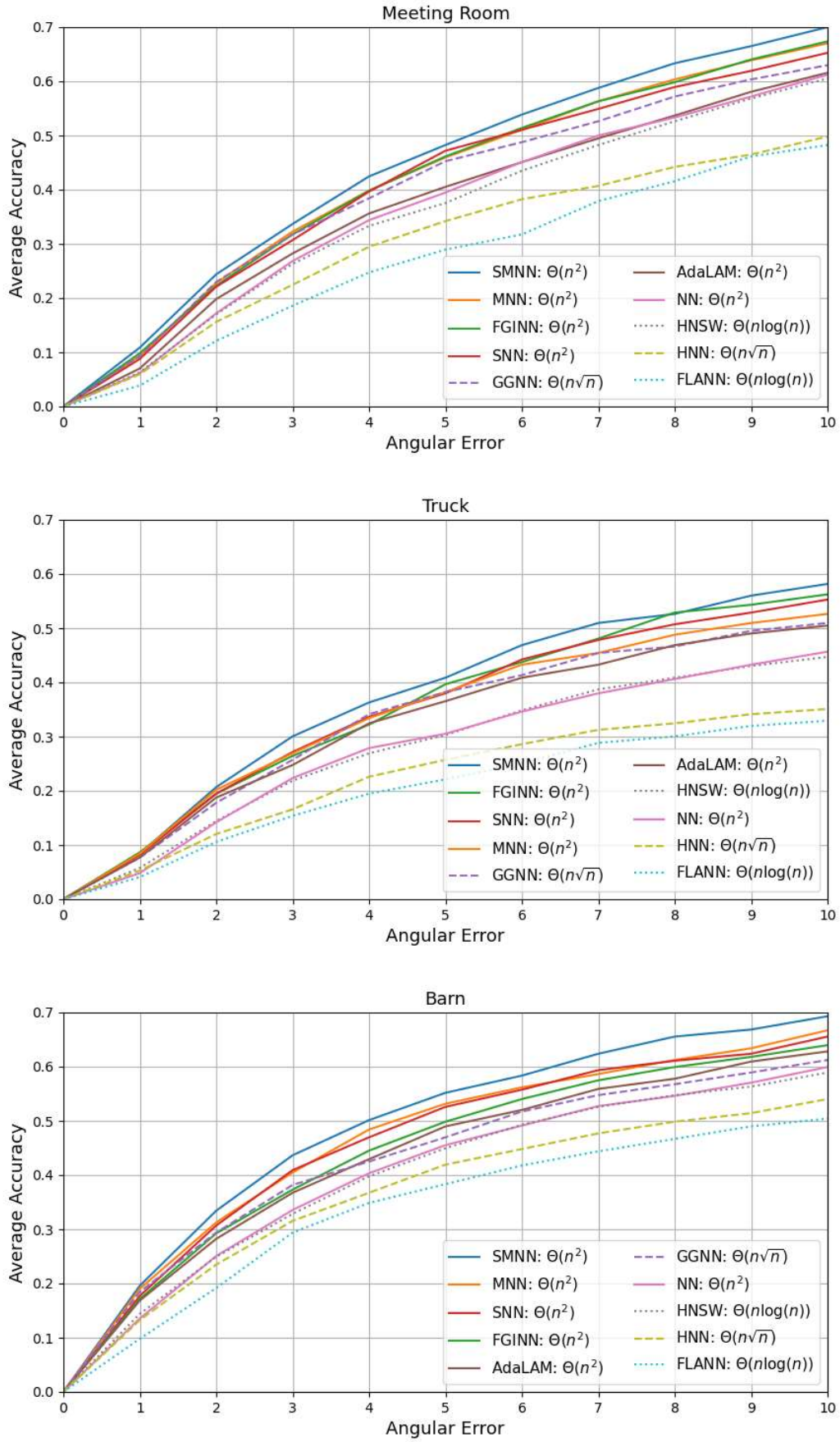


Figure 4.12. Pose Estimation Results: The x-axis represents the angular difference between the estimated and ground truth rotation vectors, while the y-axis shows the average accuracy at various error thresholds. Results are presented for three image pair datasets.

CHAPTER 5

CONCLUSIONS

The primary goal of this thesis is to reduce the computational cost of solving the common problem of image matching. Many people around the world cannot afford high-end CPUs, GPUs, and other processing units on their computers, and numerous applications need to run on mobile or embedded devices. This work is designed not just for those low-resource devices but also reflects the limitations of our own relatively modest computer, which constrained our focus to efficiency over accuracy. The motivation is practical and immediate.

This thesis presents advancements in the field of image matching through the introduction of efficient algorithms leveraging hyperdimensional computing and group testing principles. The proposed methods, Group-Guided Nearest Neighbors (GGNN) and Group-Tested Nearest Neighbors (GTNN), offer practical approaches to enhancing the efficiency of image matching without compromising accuracy.

The hierarchical approach introduced in GGNN reduces the computational complexity of feature matching from $\Theta(n^2)$ to $\Theta(n\sqrt{n})$. By grouping features spatially and matching these groups first, followed by individual feature matching within the matched groups, this method identifies sufficiently similar, geometrically meaningful matches efficiently.

The GTNN algorithm further optimizes the process by achieving $\Theta(n)$ time complexity. This is accomplished by initially matching the most distinct features to feature groups of the other image, followed by matching these distinct features only with the members of the matched groups. This linear-time matching algorithm shows promise in improving performance compared to linear-time adaptations of quadratic-time algorithms.

Empirical results on homography and pose estimation tasks indicate that the proposed GGNN and GTNN methods outperform traditional nearest neighbors algorithms and achieve performance levels comparable to other, less efficient methods. These results support the hypothesis that hierarchical matching of geometrically meaningful feature correspondences can improve or maintain matching performance with lower computational complexity.

We also proposed a technique for generating better synthetic image pair datasets for homography estimation. This technique contributes to the development of more realistic datasets, which are essential for training and evaluating image matching algorithms. Additionally, we introduced methods to facilitate faster evaluation of image matching pipelines. These methods allow for more efficient assessment of image

matching performance, speeding up the development cycle and enabling quicker iterations during algorithm optimization.

The efficiency of solutions in computer science is often a temporary concern. Historically, computing algorithms have become simpler over time when the problem size remained constant, and this trend persists. We have seen our specialized algorithms, once designed with pride, outperformed by simpler ones with fewer steps and branches, executed in parallel. We believe that even artificial general intelligence will eventually be implemented with a straightforward algorithm. Though that day has not yet arrived. Additionally, while processing units are getting more and more powerful, the number of images to process and the resolution of these images are also increasing, potentially requiring efficient algorithms in the future.

We intend to open-source the project. This release will encompass not only the implementations but also a comprehensive image matching framework that includes all discussed pipelines, various alternative modules, several benchmarks, a quick evaluation system, a synthetic dataset generation system, and a playground with numerous automated visualizations and interactive tools. It is our aspiration that this will significantly facilitate the work of others in the field. Practitioners will have the capability to experiment with pipelines by freely selecting alternative modules and parameters, thereby obtaining numerical and insightful results. Researchers can enhance the existing work by defining their own modules and automatically obtaining all intermediate and final results. Notably, our experience indicates that establishing these systems require more effort than the implementation of the proposed methods as modules.

For future research, several directions hold significant potential to further enhance the efficiency and accuracy of the proposed methods. Key areas of focus include:

- **Optimized Feature Extraction:** Learning a single high-dimensional binary descriptor instead of concatenating descriptors can enhance both efficiency and robustness in feature matching by minimizing correlation and improving speed. Integrating non-maximum suppression directly into the keypoint detection process, rather than using it as a post-processing step, can further reduce computational overhead.

- **Automated Parameter Selection for GGNN:** Developing methods for automatic parameter selection can enhance the performance of a single run and reduce the overall time required for optimizing parameters on a dataset of image pairs. For example, determining the rate at which to discard the lowest-quality group matches based on distance statistics can streamline the process.

- **Solving Easy Problems Faster with GGNN:** Enhancing the GGNN algorithm to adapt to varying levels of difficulty in image matching tasks is a promising avenue. In scenarios with small viewpoint changes, a single group match could be sufficient to generate and verify a reliable hypothesis for transformation parameters, thus speeding up the process.

- **Solving Difficult Problems Better with GGNN:** Large viewpoint and scale changes complicate matching. Grouping keypoints with similar sizes, proportional to the region size, can address large scale differences. Although our initial attempts did not improve effectiveness, there are many alternative approaches to explore. If images are reused for matching with different images, we can afford to spend more time processing them individually. One approach is performing affine simulation by warping the image or patches and allocating part of the group budget for the warped features. We experimented with bundling warped features of a region to obtain a single affine-invariant group descriptor, but we did not try bundling features of warped regions independently from other simulations to create multiple affine-covariant group descriptors for each region.

- **Maximizing Intra-Group Orthogonality for GTNN:** In GGNN, spatial grouping of features is crucial. For the GTNN method, grouping features based on descriptor distances rather than spatial proximity, using a greedy algorithm to maximize pairwise orthogonality within groups, could enhance performance. This is performed once for each image, not for each image pair. If a linear-time algorithm is needed, random grouping or a hash-based grouping technique could also work. These approaches might eliminate the need for non-maximum suppression.

- **Improving GTNN to Potentially Outperform GGNN:** The GTNN algorithm with a time complexity of $\Theta(n\sqrt{n})$, which matches all individual features rather than just the top $\sqrt{n}$ can be enhanced to potentially outperform GGNN. The key to this improvement might lie in filtering out feature-group matches with descriptor distances explainable by pure chance and ensuring local geometric consistency before performing global verification.

- **Multiview Matching**: Extending the proposed algorithms to handle multiview matching scenarios is a rewarding direction. For instance, image pairs can be selected based on similarities of feature groups could enhance the process.

In conclusion, while the proposed GGNN and GTNN methods are not revolutionary, they offer meaningful improvements in the efficiency of image matching tasks. Further research and development are encouraged to fully realize their potential and continue advancing image matching methodologies.

BIBLIOGRAPHY

Adel, Ebtsam, Mohammed Elmogy, and Hazem Elbakry. 2014. “Image stitching based on feature extraction techniques: a survey.” *International Journal of Computer Applications* 99 (6): 1–8.

Aldridge, Matthew, Oliver Johnson, and Jonathan Scarlett. 2019. “Group testing: an information theory perspective.” ISBN: 1567-2190 Publisher: Now Publishers, Inc. *Foundations and Trends® in Communications and Information Theory* 15 (3): 196–392.

Allebach, Jan, and Ping Wah Wong. 1996. “Edge-directed interpolation.” In *Proceedings of 3rd IEEE international conference on image processing,* 3:707–710. IEEE.

Arandjelović, Relja, and Andrew Zisserman. 2012. “Three things everyone should know to improve object retrieval.” In *2012 IEEE conference on computer vision and pattern recognition,* 2911–2918. IEEE.

Bailey, Tim, and Hugh Durrant-Whyte. 2006. “Simultaneous localization and mapping (SLAM): Part II.” *IEEE robotics & automation magazine* 13 (3): 108–117.

Balntas, Vassileios, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. 2017. “HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 3852–3861.

Barath, Daniel, Luca Cavalli, and Marc Pollefeys. 2022. “Learning to find good models in RANSAC.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 15744–15753.

Barath, Daniel, and Jiří Matas. 2018. “Graph-cut RANSAC.” In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 6733–6741.

Barath, Daniel, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. 2020. “MAGSAC++, a fast, reliable and accurate robust estimator.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 1304–1312.

Barroso-Laguna, Axel, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2019. “Key.Net: Keypoint detection by handcrafted and learned cnn filters.” In *Proceedings of the IEEE/CVF international conference on computer vision,* 5836–5844.

Bastanlar, Yalin, Alptekin Temizel, and Yasemin Yardimci. 2010. “Improved SIFT matching for image pairs with scale difference.” Publisher: IET, *Electronics Letters* 46 (5).

Bastanlar, Yalin, Alptekin Temizel, Yasemin Yardimci, and Peter Sturm. 2010. "Effective structure-from-motion for hybrid camera systems." In *Proceedings of the International Conference on Pattern Recognition.*

Biau, Gérard, and Erwan Scornet. 2016. "A random forest guided tour." *Test* 25:197–227.

Brachmann, Eric, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. 2017. "Dsac-differentiable ransac for camera localization." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 6684–6692.

Brachmann, Eric, and Carsten Rother. 2019. "Neural-guided RANSAC: Learning where to sample model hypotheses." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 4322–4331.

Bradski, Gary. 2000. "The opencv library." ISBN: 1044-789X Publisher: Miller Freeman Inc. *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25 (11): 120–123.

Breiman, Leo. 2017. *Classification and regression trees.* Routledge.

Breimann, Leo, Jerome H Friedman, Richard A Olshen, and Charles J Stone. 1984. "Classification and regression trees." *Pacific Grove, Wadsworth.*

Brown, Matthew, and David G. Lowe. 2007. "Automatic panoramic image stitching using invariant features." ISBN: 0920-5691 Publisher: Springer, *International journal of computer vision* 74:59–73.

Brown, Matthew, and Sabine Süsstrunk. 2011. "Multi-spectral SIFT for scene category recognition." In *CVPR 2011,* 177–184. IEEE.

Calonder, Michael, Vincent Lepetit, Mustafa Özuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal V. Fua. 2012. "BRIEF: Computing a Local Binary Descriptor Very Fast." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34:1281–1298.

Calonder, Michael, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 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 11,* 778–792. Springer.

Cao, Sixi, Runmin Zhang, Lun Luo, Beinan Yu, Zehua Sheng, Junwei Li, and Hui Shen. 2023. "Recurrent Homography Estimation Using Homography-Guided Image Warping and Focus Transformer." *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 9833–9842.

Cao, Si-Yuan, Jianxin Hu, Zehua Sheng, and Hui-Liang Shen. 2022. "Iterative deep homography estimation." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 1879–1888.

Cavalli, Luca, Daniel Barath, Marc Pollefeys, and Viktor Larsson. 2023. "Consensus-adaptive ransac." *arXiv preprint arXiv:2307.14030.*

Cavalli, Luca, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. 2020. "Handcrafted outlier detection revisited." In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16,* 770–787. Springer. ISBN: 3-030-58528-X.

Cheng, Jian, Cong Leng, Jiaxiang Wu, Hainan Cui, and Hanqing Lu. 2014. "Fast and accurate image matching with cascade hashing for 3d reconstruction." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 1–8.

Chum, Ondrej, and Jiri Matas. 2005. "Matching with PROSAC-progressive sample consensus." In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05),* 1:220–226. IEEE.

Chum, Ondřej, Jiří Matas, and Josef Kittler. 2003. "Locally optimized RANSAC." In *Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25,* 236–243. Springer.

DeTone, Daniel, Tomasz Malisiewicz, and Andrew Rabinovich. 2016. "Deep image homography estimation." *arXiv preprint arXiv:1606.03798.*

———. 2018. "Superpoint: Self-supervised interest point detection and description." In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops,* 224–236.

Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2015. "Image super-resolution using deep convolutional networks." *IEEE transactions on pattern analysis and machine intelligence* 38 (2): 295–307.

Du, Dingzhu, Frank K Hwang, and Frank Hwang. 2000. *Combinatorial group testing and its applications.* Vol. 12. World Scientific.

Durrant-Whyte, Hugh, and Tim Bailey. 2006. "Simultaneous localization and mapping: part I." *IEEE robotics & automation magazine* 13 (2): 99–110.

Erion, Gabriel, Joseph D Janizek, Carly Hudelson, Richard B Utarnachitt, Andrew M McCoy, Michael R Sayre, Nathan J White, and Su-In Lee. 2022. “A cost-aware framework for the development of AI models for healthcare applications.” *Nature Biomedical Engineering* 6 (12): 1384–1398.

Fischler, Martin A, and Robert C Bolles. 1981. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.” *Communications of the ACM* 24 (6): 381–395.

Fu, Cong, Chao Xiang, Changxu Wang, and Deng Cai. 2017. “Fast approximate nearest neighbor search with the navigating spreading-out graph.” *arXiv preprint arXiv:1707.00143.*

Fuentes-Pacheco, Jorge, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. 2015. “Visual simultaneous localization and mapping: a survey.” *Artificial intelligence review* 43:55–81.

Gionis, Aristides, Piotr Indyk, Rajeev Motwani, et al. 1999. “Similarity search in high dimensions via hashing.” In *Vldb,* 99:518–529. 6.

Gleize, Pierre, Weiyao Wang, and Matt Feiszli. 2023. “SiLK: Simple Learned Keypoints.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 22499–22508.

Hong, Ming, Yuhang Lu, Nianjin Ye, Chunyu Lin, Qijun Zhao, and Shuaicheng Liu. 2022. “Unsupervised Homography Estimation with Coplanarity-Aware GAN.” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 17642–17651.

Indyk, Piotr, and Rajeev Motwani. 1998. “Approximate nearest neighbors: towards removing the curse of dimensionality.” In *Proceedings of the thirtieth annual ACM symposium on Theory of computing,* 604–613.

Iscen, Ahmet, and Ondrej Chum. 2018. “Local orthogonal-group testing.” In *Proceedings of the European Conference on Computer Vision (ECCV),* 449–465.

Ivashechkin, Maksym, Daniel Barath, and Jiří Matas. 2021. “Vsac: Efficient and accurate estimator for h and f.” In *Proceedings of the IEEE/CVF international conference on computer vision,* 15243–15252.

Ivashechkin, Maksym, Dániel Baráth, and Jiri Matas. 2021. “USACv20: robust essential, fundamental and homography matrix estimation.” *ArXiv* abs/2104.05044.

Jafari, Omid, Preeti Maurya, Parth Nagarkar, Khandker Mushfiqul Islam, and Chidambaram Crushev. 2021. “A survey on locality sensitive hashing algorithms and their applications.” *arXiv preprint arXiv:2102.08942.*

Janisch, Jaromír, Tomáš Pevnỳ, and Viliam Lisỳ. 2019. “Classification with costly features using deep reinforcement learning.” In *Proceedings of the AAAI Conference on Artificial Intelligence,* 33:3959–3966. 01.

Jin, Yuhe, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. 2021. “Image matching across wide baselines: From paper to practice.” ISBN: 0920-5691 Publisher: Springer, *International Journal of Computer Vision* 129 (2): 517–547.

Kanerva, Pentti. 2009. “Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors.” *Cognitive computation* 1:139–159.

———. 2022. “Hyperdimensional Computing: An Algebra for Computing with Vectors.” Publisher: Wiley Online Library, *Advances in Semiconductor Technologies: Selected Topics Beyond Conventional CMOS,* 25–42.

Ke, Yan, and Rahul Sukthankar. 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.* 2:II–II. IEEE.

Kingsford, Carl, and Steven L Salzberg. 2008. “What are decision trees?” *Nature biotechnology* 26 (9): 1011–1013.

Knapitsch, Arno, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. “Tanks and temples: Benchmarking large-scale scene reconstruction.” *ACM Transactions on Graphics (ToG)* 36 (4): 1–13.

Le, Hoang, Feng Liu, Shu Zhang, and Aseem Agarwala. 2020. “Deep Homography Estimation for Dynamic Scenes.” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 7649–7658.

Lebeda, Karel, Jırı Matas, and Ondrej Chum. 2012. “Fixing the locally optimized ransac–full experimental evaluation.” In *British machine vision conference,* vol. 2. Citeseer.

Levi, Gil, and Tal Hassner. 2016. “LATCH: learned arrangements of three patch codes.” In *2016 IEEE winter conference on applications of computer vision (WACV),* 1–9. IEEE. ISBN: 1-5090-0641-9.

Levin, Anat, Assaf Zomet, Shmuel Peleg, and Yair Weiss. 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 8,* 377–389. Springer.

Li, Xin, and Michael T Orchard. 2001. “New edge-directed interpolation.” *IEEE transactions on image processing* 10 (10): 1521–1527.

Li, Yunyao, Kehang Chen, Shilei Sun, and Chu He. 2022. “Multi-scale homography estimation based on dual feature aggregation transformer.” *IET Image Process.* 17:1403–1416.

Liao, Yanhao, Yinhui Luo, and Xingyi Wang. 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.*

Lin, Chung-Ching, Sharathchandra U Pankanti, Karthikeyan Natesan Ramamurthy, and Aleksandr Y Aravkin. 2015. “Adaptive as-natural-as-possible image stitching.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 1155–1163.

Lin, Henry W, Max Tegmark, and David Rolnick. 2017. “Why does deep and cheap learning work so well?” *Journal of Statistical Physics* 168:1223–1247.

Lindenberger, Philipp, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. 2021. “Pixel-perfect structure-from-motion with featuremetric refinement.” In *Proceedings of the IEEE/CVF international conference on computer vision,* 5987–5997.

Loh, Wei-Yin. 2011. “Classification and regression trees.” *Wiley interdisciplinary reviews: data mining and knowledge discovery* 1 (1): 14–23.

———. 2014. “Fifty years of classification and regression trees.” *International Statistical Review* 82 (3): 329–348.

Lowe, David G. 1999. “Object recognition from local scale-invariant features.” In *Proceedings of the seventh IEEE international conference on computer vision,* 2:1150–1157. Ieee.

———. 2004. “Distinctive image features from scale-invariant keypoints.” ISBN: 0920-5691 Publisher: Springer, *International journal of computer vision* 60:91–110.

Luo, Yinhui, Xingyi Wang, Yanhao Liao, Qiang Fu, Chang Shu, Yuezhou Wu, and Yuanqing He. 2023. “A Review of Homography Estimation: Advances and Challenges.” *Electronics.*

Luo, Yinhui, Xingyi Wang, Yuezhou Wu, and Chang Shu. 2022. “Detail-Aware Deep Homography Estimation for Infrared and Visible Image.” *Electronics.*

———. 2023. “Infrared and Visible Image Homography Estimation Using Multiscale Generative Adversarial Network.” *Electronics.*

Luo, Zixin, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. 2019. “Contextdesc: Local descriptor augmentation with cross-modality context.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 2527–2536.

Malkov, Yu A, and Dmitry A Yashunin. 2018. “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs.” *IEEE transactions on pattern analysis and machine intelligence* 42 (4): 824–836.

Mikolajczyk, Krystian, and Cordelia Schmid. 2005. “A performance evaluation of local descriptors.” ISBN: 0162-8828 Publisher: IEEE, *IEEE transactions on pattern analysis and machine intelligence* 27 (10): 1615–1630.

Mishchuk, Anastasiia, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. 2017. “Working hard to know your neighbor’s margins: Local descriptor learning loss.” *Advances in neural information processing systems* 30.

Mishkin, Dmytro, Jiri Matas, and Michal Perdoch. 2015. “MODS: Fast and robust method for two-view matching.” *Computer vision and image understanding* 141:81–93.

Montemerlo, Michael, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. 2002. “FastSLAM: A factored solution to the simultaneous localization and mapping problem.” *Aaai/iaai* 593598:593–598.

Moulon, Pierre, Pascal Monasse, Romuald Perrot, and Renaud Marlet. 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,* 60–74. Springer.

Muja, Marius, and David G. Lowe. 2009. “Fast approximate nearest neighbors with automatic algorithm configuration.” *VISAPP (1)* 2 (331): 2.

Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D. Tardos. 2015. “ORB-SLAM: a versatile and accurate monocular SLAM system.” ISBN: 1552-3098 Publisher: IEEE, *IEEE transactions on robotics* 31 (5): 1147–1163.

Neubert, Peer, and Stefan Schubert. 2021. “Hyperdimensional computing as a framework for systematic aggregation of image descriptors.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 16938–16947.

Parmar, Aakash, Rakesh Katariya, and Vatsal Patel. 2019. “A review on random forest: An ensemble classifier.” In *International conference on intelligent data communication technologies and internet of things (ICICI) 2018,* 758–763. Springer.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12:2825–2830.

Peter, Sven, Ferran Diego, Fred A Hamprecht, and Boaz Nadler. 2017. “Cost efficient gradient boosting.” *Advances in neural information processing systems* 30.

Placed, Julio A, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A Castellanos. 2023. “A survey on active simultaneous localization and mapping: State of the art and new frontiers.” *IEEE Transactions on Robotics* 39 (3): 1686–1705.

Potharst, Rob, and Adrianus Johannes Feelders. 2002. “Classification trees for problems with monotonicity constraints.” *ACM SIGKDD Explorations Newsletter* 4 (1): 1–10.

Reyzin, Lev. 2011. “Boosting on a budget: Sampling for feature-efficient prediction.” In *Proceedings of the 28th International Conference on Machine Learning (ICML-11),* 529–536.

Riba, Edgar, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. 2020. “Kornia: an open source differentiable computer vision library for pytorch.” In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision,* 3674–3683.

Rokach, Lior. 2016. “Decision forest: Twenty years of research.” *Information Fusion* 27:111–125.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. “ORB: An efficient alternative to SIFT or SURF.” In *2011 International conference on computer vision,* 2564–2571. Ieee. ISBN: 1-4577-1102-8.

Santellani, Emanuele, Christian Sormann, Mattia Rossi, Andreas Kuhn, and Friedrich Fraundorfer. 2022. “Md-net: Multi-detector for local feature extraction.” In *2022 26th International conference on pattern recognition (ICPR),* 3944–3951. IEEE.

Schonberger, Johannes L., and Jan-Michael Frahm. 2016. “Structure-from-motion revisited.” In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 4104–4113.

Schönberger, Johannes L, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. 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 14,* 501–518. Springer.

Silpa-Anan, Chanop, and Richard Hartley. 2008. “Optimised KD-trees for fast image descriptor matching.” In *2008 IEEE Conference on Computer Vision and Pattern Recognition,* 1–8. IEEE.

Son, Sanghyun, and Kyoung Mu Lee. 2021. “Srwarp: Generalized image super-resolution under arbitrary transformation.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 7782–7791.

Stachniss, Cyrill, John J Leonard, and Sebastian Thrun. 2016. “Simultaneous localization and mapping.” *Springer Handbook of Robotics,* 1153–1176.

Strecha, Christoph, Alex Bronstein, Michael Bronstein, and Pascal Fua. 2011. “LDAHash: Improved matching with smaller descriptors.” *IEEE transactions on pattern analysis and machine intelligence* 34 (1): 66–78.

Suárez, Iago, Ghesn Sfeir, José M. Buenaposada, and Luis Baumela. 2020. “BEBLID: Boosted efficient binary local image descriptor.” ISBN: 0167-8655 Publisher: Elsevier, *Pattern recognition letters* 133:366–372.

Suwanwimolkul, Suwichaya, Satoshi Komorita, and Kazuyuki Tasaka. 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,* 2262–2271.

Szeliski, Richard, et al. 2007. “Image alignment and stitching: A tutorial.” *Foundations and Trends® in Computer Graphics and Vision* 2 (1): 1–104.

Tam, Wing-Shan, Chi-Wah Kok, and Wan-Chi Siu. 2010. “Modified edge-directed interpolation for images.” *Journal of Electronic imaging* 19 (1): 013011–013011.

Thrun, Sebastian. 2008. “Simultaneous localization and mapping.” In *Robotics and cognitive approaches to spatial mapping,* 13–41. Springer.

Tian, Yurun, Axel Barroso Laguna, Tony Ng, Vassileios Balntas, and Krystian Mikolajczyk. 2020. "HyNet: Learning local descriptor with hybrid similarity measure and triplet loss." *Advances in neural information processing systems* 33:7401–7412.

Tyszkiewicz, Micha l, Pascal Fua, and Eduard Trulls. 2020. "DISK: Learning local features with policy gradient." *Advances in Neural Information Processing Systems* 33:14254–14265.

Ullman, Shimon. 1979. "The interpretation of structure from motion." *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203 (1153): 405–426.

Wang, Mengzhao, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. 2021. "A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search." *arXiv preprint arXiv:2101.12631.*

Wang, Xiang, Chen Wang, Bing Liu, Xiaoqing Zhou, Liang Zhang, Jin Zheng, and Xiao Bai. 2021. "Multi-view stereo in the deep learning era: A comprehensive review." *Displays* 70:102102.

Wang, Xingyi, Yinhui Luo, Qiang Fu, Yun Rui, Chang Shu, Yuezhou Wu, Zhige He, and Yuanqing He. 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.

Wang, Xintao, Liangbin Xie, Chao Dong, and Ying Shan. 2021. "Real-esrgan: Training real-world blind super-resolution with pure synthetic data." In *Proceedings of the IEEE/CVF international conference on computer vision,* 1905–1914.

Wang, Xintao, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. "Esrgan: Enhanced super-resolution generative adversarial networks." In *Proceedings of the European conference on computer vision (ECCV) workshops.*

Wang, Yifan, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. 2018. "A fully progressive approach to single-image super-resolution." In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops,* 864–873.

Wang, Zhaobin, and Zekun Yang. 2020. "Review on image-stitching techniques." *Multimedia Systems* 26 (4): 413–430.

Wolpert, David H, and William G Macready. 1997. "No free lunch theorems for optimization." *IEEE transactions on evolutionary computation* 1 (1): 67–82.

Wu, Changchang. 2011. “VisualSFM: A visual structure from motion system.” *http://www.cs.washington.edu /homes/ccwu/vsfm.*

———. 2013. “Towards linear-time incremental structure from motion.” In *2013 International Conference on 3D Vision-3DV 2013,* 127–134. IEEE.

Xiao, Jun, Zihang Lyu, Cong Zhang, Yakun Ju, Changjian Shui, and Kin-Man Lam. 2024. “Towards Progressive Multi-Frequency Representation for Image Warping.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2995–3004.

Xu, Zhixiang, Matt J Kusner, Kilian Q Weinberger, Minmin Chen, and Olivier Chapelle. 2014. “Classifier cascades and trees for minimizing feature evaluation cost.” *The Journal of Machine Learning Research* 15 (1): 2113–2144.

Yi, Kwang Moo, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. 2018. “Learning to find good correspondences.” In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2666–2674.

Yu, Guoshen, and Jean-Michel Morel. 2011. “ASIFT: An algorithm for fully affine invariant comparison.” *Image Processing On Line* 1:11–38.

Zaragoza, Julio, Tat-Jun Chin, Michael S Brown, and David Suter. 2013. “As-projective-as-possible image stitching with moving DLT.” In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2339–2346.

Zhang, Kai, Jingyun Liang, Luc Van Gool, and Radu Timofte. 2021. “Designing a practical degradation model for deep blind image super-resolution.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 4791–4800.