

# Efficient Feature Matching for Large-scale Images based on Cascade Hash and Local Geometric Constraint

Kan You<sup>1,2</sup>, San Jiang<sup>1,\*</sup>, Yaxin Li<sup>2</sup>, Wanshou Jiang<sup>3</sup>, Xiangxiang Huang<sup>4</sup>

<sup>1</sup> School of Computer Science, China University of Geosciences, Wuhan 430074, China - jiangsan@cug.edu.cn

<sup>2</sup> Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong 999077, China

<sup>3</sup> State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China

<sup>4</sup> School of Computer and Information, Anhui Polytechnic University, Wuhu 241000, China

**Keywords:** Feature Matching, Cascade Hashing, Local Geometric Constraint, Structure from Motion, 3D Reconstruction.

## Abstract

Feature matching plays a crucial role in 3D reconstruction to provide correspondences between overlapped images. The accuracy and efficiency of feature matching significantly impact the performance of 3D reconstruction. The widely used framework with the exhaustive nearest neighbor searching (NNS) between descriptors and RANSAC-based geometric estimation is, however, low-efficient and unreliable for large-scale UAV images. Inspired by indexing-based NNS, this paper implements an efficient feature matching method for large-scale images based on Cascade Hashing and local geometric constraints. Our proposed method improves upon traditional feature matching approaches by introducing a combination of image retrieval, data scheduling, and GPU-accelerated Cascade Hashing. Besides, it utilizes a local geometric constraint to filter matching results within a matching framework. On the one hand, the GPU-accelerated Cascade Hashing technique generates compact and discriminative hash codes based on image features, facilitating the rapid completion of the initial matching process, and significantly reducing the search space and time complexity. On the other hand, after the initial matching is completed, the method employs a local geometric constraint to filter the initial matching results, enhancing the accuracy of the matching results. This forms a three-tier framework based on data scheduling, GPU-accelerated Cascade Hashing, and local geometric constraints. We conducted experiments using two sets of large-scale UAV image data, comparing our method with SIFTGPU to evaluate its performance in initial matching, outlier rejection, and 3D reconstruction. The results demonstrate that our method achieves a feature matching speed 2.0 times that of SIFTGPU while maintaining matching accuracy and producing comparable reconstruction results. This suggests that our method holds promise for efficiently addressing large-scale image matching.

## 1. INTRODUCTION

Feature matching is a fundamental task in the fields of photogrammetry and computer vision for wide-ranging applications, including image retrieval (Li et al., 2021), object recognition, and 3D reconstruction (Jiang et al., 2022b). It involves identifying and matching identical features from spatially overlapped images, a critical aspect of understanding and interpreting visual data. In the literature, extensive research has been conducted for efficient and accurate feature matching, which ranges from the earlier corner detectors to the recent invariant features (Fan et al., 2019; Jiang et al., 2021b).

Traditional feature matching algorithms include methods based on the nearest Euclidean distance and index-based matching. Approaches based on the nearest Euclidean distance include nearest neighbor matching and K-nearest neighbor matching. Nearest neighbor matching, a classical method, involves calculating the Euclidean distance between the target feature and all other features, selecting the nearest one as the match. While simple, it is computationally expensive and susceptible to noise. K-nearest neighbor matching extends this by selecting K-nearest features, improving accuracy but making the choice of K challenging. Muja and Lowe (2014) found that K-nearest neighbor matching is most suitable for fast approximate search in high-dimensional space, and introduced a new algorithm for fast approximate matching of binary features. Its open-source library

FLANN is used in a large number of research and industrial projects and is widely used.

Index-based matching methods include KD-tree matching and hash-based matching. KD-tree matching uses a spatial partitioning tree to reduce computational costs during matching. While fast, constructing KD-trees is relatively complex, and its performance with high-dimensional features is suboptimal (Silpa-Anan and Hartley, 2008). Hash-based matching rapidly finds matching features by mapping features to a hash table (Cheng et al., 2014; Cao et al., 2023). While efficient, hash collisions and sensitivity to noise can be issues. Based on the basic idea of index-based matching, Jiang et al. (2022a) proposed an integrated workflow to achieve simultaneous match pair selection and guided feature matching for image orientation. The core idea of the proposed algorithm is to explore the index structure of both inverted and direct indexes in the context of a vocabulary tree-based image retrieval.

Additionally, there are some other feature matching methods like Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981), Support Vector Machine (SVM) (Hearst et al., 1998), and deep learning (Jiang et al., 2021a). RANSAC mitigates noise and outliers by randomly selecting a subset of samples for matching and performing consistency checks on all samples. SVM maps features to a high-dimensional space to find optimal matching results. Deep learning automatically learns feature representations and performs matching but requires substantial training data and computational resources (Maltezos et al.,

\* Corresponding author

2017; Liu et al., 2018).

Each algorithm has its pros and cons, requiring selection based on specific application scenarios and needs. As image data scales up, traditional feature matching methods often struggle to maintain efficiency and accuracy due to the associated high computational costs and complexity.

Our proposed method combines the advantages of Euclidean distance-based and index-based feature matching methods, excelling in both speed and accuracy compared to other methods. In terms of speed, our method generates scheduling orders based on image retrieval results, significantly optimizing data transfer between memory and GPU. Simultaneously, our method uses GPU to accelerate Cascade Hashing, ensuring a smooth and efficient matching process. In terms of accuracy, our method exhibits improvements over traditional approaches. In the outlier elimination stage, local geometric constraints are applied to filter initial matching results. This forms a two-tier outlier elimination structure with RANSAC, enhancing both speed and accuracy, representing considerable progress. Our method is well-suited for Cascade Hashing under the framework of image retrieval, data scheduling, GPU acceleration, and local geometric constraints. This opens up new possibilities for improving the efficiency and accuracy of large-scale image feature matching. We believe our proposed method not only contributes to the field of photogrammetry but also holds potential implications for other areas such as image retrieval and monitoring systems.

The paper will introduce the methods involved in the second part. In the third part, we will analyze the issues with existing methods and provide detailed explanations of our improvements to current methods. In the fourth part, we will present our experimental results, indicating that our method performs well in large-scale feature matching.

## 2. THE WORKFLOW OF CASCADE HASHING MATCHING

Feature matching is typically the primary factor contributing to time consumption in 3D reconstruction. Consequently, many researchers in related fields have been continuously exploring ways to optimize the time consumption of the feature matching phase. For example, the SIFTGPU algorithm proposed by Wu (2013) accelerates the traditional SIFT algorithm and stands as a mainstream algorithm in current feature matching. Cao et al. (2023) proposed a feature matching approach based on hash indexing. Cheng et al. (2014) proposed a Cascade Hashing approach to accelerate feature matching. The method they introduced exhibits higher accuracy and speed compared to traditional hash indexing matching. Xu et al. (2017) and others proposed a Cascade Hashing algorithm accelerated by GPU. Zhang et al. (2023) combined data scheduling and GPU-accelerated Cascade Hashing. They also filtered initial matches using rough positional data on image pairs, achieving high-speed feature matching. Building upon this, we integrated various methods, forming a three-tier framework based on data scheduling, GPU-accelerated Cascade Hashing, and local geometric constraints. We propose an accelerated feature matching algorithm tailored for UAV images without positional data. In the following sections, we will introduce the methods involved.

The data scheduling component is the first part of our framework. Due to the substantial number of invalid match pairs resulting from an exhaustive matching strategy, employing an effective algorithm to filter potential matching image pairs before the matching process can significantly reduce the overall

time consumption. In our method, we initiate the process by conducting image retrieval on the dataset to establish adjacency relationships between different images. In the algorithmic part of data scheduling, we draw inspiration from the approach proposed by Zhang et al. (2023). This involves generating an image scheduling order based on the results of image retrieval. In the following sections, we will provide a detailed introduction to each part of the data scheduling process.

### 2.1 Image Retrieval

We have employed an efficient method for retrieving matched pairs in image retrieval (Jiang et al., 2023). This method integrates vector of locally aggregated descriptors (VLAD) (Jgou et al., 2012) and hierarchical navigable small word (HNSW) (Malkov and Yashunin, 2018) and establishes adjacency relationships between images through image retrieval before matching image pairs, thereby reducing the computational cost of global matching. Specifically, the key steps of this method include:

- 1) **Online Training of Individual Codebook:** By considering the redundancy of UAV images and local features, we avoid ambiguity in training codebooks from other datasets. Through online training, an individual codebook is established for the subsequent aggregation of local features.
- 2) **Aggregation of Local Features:** Utilizing the trained codebook, local features of each image are aggregated into a high-dimensional global descriptor. This significantly reduces the number of local features, alleviating the burden of nearest neighbor searching in image retrieval.
- 3) **Graph Indexing of Global Descriptors:** The global descriptors are indexed using a graph structure based on hierarchical-navigable-small-world principles, enabling efficient nearest neighbor searches. This contributes to the acceleration of matched pair retrieval.
- 4) **Match Pair Retrieval and View Graph Construction:** An adaptive threshold selection strategy is employed to retrieve match pairs, which are then used to construct a view graph. This graph facilitates divide-and-conquer-based parallel Structure from Motion (SfM) reconstruction.

In practical applications of image retrieval, we have successfully reduced data transfer costs by generating a rational scheduling order based on image pairs. This method not only ensures the selection of appropriate image pairs for matching but also enhances the speed of feature matching.

### 2.2 Generation of Scheduling Order

In this study, the scheduling algorithm used is mainly based on the memory capacity and the adjacency matrix generated by the image retrieval part to generate the scheduling order of images. During the scheduling process, we divide the images into two categories, images in the video memory and images not in the video memory, and adopt a specific scheduling strategy. Specifically, when there is no image in the video memory, we prioritize the image with the largest number of neighbors to the video memory. When there are images in the video memory and the number of images does not reach the maximum value that the video memory can accommodate, considering the adjacency relationship between the two types of images and within

each type of image, we adopt the strategy proposed by Zhang et al. (2023). By setting a weight factor  $\omega$ , for images that have not been transferred to the video memory, calculate the weight  $K_{select}$  of each image.

$$K_{select} = \omega * M - K \quad (1)$$

As shown in Equation (1), among them,  $M$  is the number of adjacent images in the GPU memory, and  $K$  is the number of adjacent images not yet transferred into the GPU memory. Images with the maximum  $K_{select}$  are prioritized for scheduling into the GPU memory. When the number of images in the GPU memory reaches its maximum, the image with the minimum  $K$  value is selected from the images not yet transferred into the GPU memory and removed from the GPU memory. This process continues until all images are successfully transferred, generating the scheduling order.

During the experimental process, we found that due to the limited number of images that the GPU memory can accommodate, in the actual scheduling process, some image pairs may be unable to complete matching because certain images have been removed from the GPU memory. To address this situation, we adopted a rematch strategy. For matched pairs, we mark them in the adjacency matrix. Then, based on the adjacency matrix, we can continue to generate the scheduling order of images using the scheduling algorithm introduced above. Through this cyclic matching strategy, we eventually complete the matching of all image pairs with as few schedules as possible.

In summary, the image retrieval-based scheduling algorithm we proposed is an efficient scheduling method that reduces data transfer costs by generating a reasonable scheduling order. We believe that this method will play an important role in future research.

### 2.3 Cascade Hashing

The Cascade Hashing algorithm is an efficient method for feature point matching, combining a series of steps including SIFT-GPU feature extraction, hash code calculation, hash table establishment, and the matching process. The process is shown in Figure 1. Here is a detailed description of each step in the Cascade Hashing algorithm:

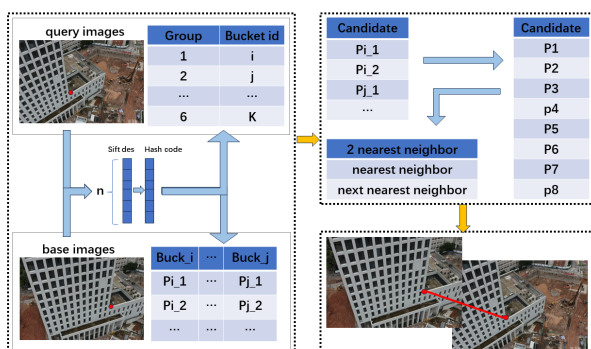


Figure 1. The overall workflow of cascade hashing

- 1) **Feature Extraction:** Utilize the SIFT-GPU algorithm to detect and extract key points (feature points) along with their corresponding descriptors (Wu, 2013). These descriptors typically constitute 128-dimensional vectors with invariance to scale and rotation.

- 2) **Hash Code Calculation:** For each extracted SIFT feature point, calculate its hash code. This process typically involves using a set of random matrices for projection, mapping the high-dimensional SIFT descriptors to low-dimensional binary hash codes. Such hash codes serve as a compact representation of feature points, facilitating efficient storage and retrieval.
- 3) **Hash Table Establishment:** Use the Locality-Sensitive Hashing (LSH) algorithm to map the computed hash codes to multiple hash tables. Each hash table comprises multiple buckets, grouping feature points with similar hash codes into the same bucket. This hash table structure enables quick localization of similar feature points during matching.
- 4) **Hash Table Lookup for Candidate Points during Matching:** For each feature point in the query image, perform a hash table lookup to find its candidate points. Due to the bucket structure inside the hash table, only feature points within the same bucket are potential candidates for matching.
- 5) **Low Ratio Test-Based Filtering of Candidate Points:** For each query point, execute Lowe ratio tests among the candidate points retrieved from the hash table. This test compares the distances between the query point and its two nearest neighbors. By setting a threshold, points that do not meet the similarity requirements are filtered out.
- 6) **Obtaining the Final Matching Results:** After filtering with Lowe ratio tests, the retained candidate points constitute the final matching results. These points exhibit similar features in the hash code space and have undergone similarity checks, enhancing the accuracy of the matching process.

The key advantage of the Cascade Hashing algorithm lies in its ability to map the high-dimensional descriptors of SIFT features to a low-dimensional hash code space, enabling efficient matching of feature points through hash tables. During matching, the Lowe ratio test further enhances the reliability of correspondences. The efficiency of this process makes the algorithm perform exceptionally well when handling large-scale image datasets.

## 3. OUR IMPROVEMENTS

Despite the excellent performance of Cascade Hashing in matching, certain issues persist. Through practical usage, we conducted an analysis of these problems and proposed our solutions based on the identified issues.

### 3.1 Problems

Currently, although Cascade Hashing exhibits fast matching speed, its initial matching phase involves a significant number of errors, potentially resulting in extended time consumption during gross error removal. To address this issue, we have employed a series of methods.

During the initial matching phase, we have fully utilized GPU for accelerated Cascade Hashing. However, despite the superior performance of Cascade Hashing in certain aspects, it still has some limitations. Hash encoding, being a form of compression

representation, may lead to information loss, affecting the accuracy of matching and ultimately resulting in suboptimal precision in the matching results. During program execution, the GPU-based Cascade Hashing feature matching runs in parallel with the CPU-based gross error removal. As a result, the overall speed of our method is influenced not only by the speed of Cascade Hashing but also by the completion time of the gross error removal part. Experiments have shown that, due to precision considerations, in cases where the initial matching speed is fast, the final time of feature matching is primarily determined by the speed of gross error removal. Therefore, we need to adopt corresponding strategies to further optimize the gross error removal part to enhance the overall matching performance.

### 3.2 Improvements

In order to solve the above problems, we chose to adopt a two-layer filtering algorithm, combining the algorithm proposed by Jiang et al. (2020) with our method, using local geometric constraints to filter the initial matching results, and using RANSAC to eliminate gross errors in the filtered data. The workflow of the local geometric constraint is illustrated in Figure 2.

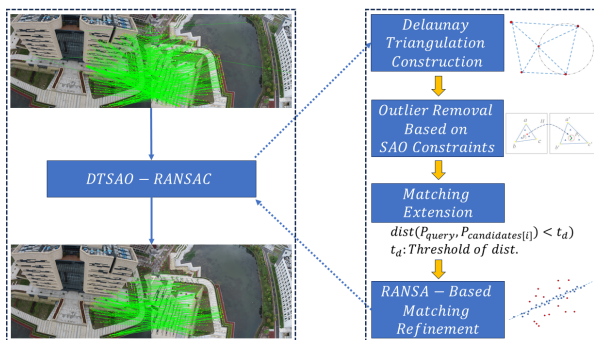


Figure 2. The overall workflow of local geometric constraint.

The core idea is to design local geometric constraints within adjacent structures using Delaunay triangulation and a two-stage approach for outlier removal and matching refinement. The proposed algorithm is referred to as DTSAO-RANSAC. The algorithm's workflow is summarized as follows:

- 1) **Initial Matching Generation:** Detect and describe feature points of images using the SIFT algorithm. Obtain initial matches by Cascade Hashing between SIFT descriptors.
- 2) **Delaunay Triangulation Construction:** Build Delaunay triangulation (G1) using the initial matches and its corresponding graph G2.
- 3) **Outlier Removal Based on SAO Constraints:** Remove outliers with the affine-invariant spatial angular order (SAO) constraints on the target vertices in G1. Iteratively execute a layered elimination strategy until the dissimilarity score is below a specified threshold.
- 4) **Matching Extension:** Use triangulation constraints for matching extension to recover potentially missed true matches.
- 5) **RANSAC-Based Matching Refinement:** Utilize the RANSAC algorithm to estimate the fundamental matrix and refine global geometric constraints for the retained matches.

Overall, the DTSAO-RANSAC algorithm filters, extends, and refines initial matches by combining local geometric (SAO constraints) and global geometric (RANSAC) strategies, enhancing the reliability and efficiency of the matching process. Experimental results demonstrate that DTSAO-RANSAC achieves efficient outlier removal, providing reliable matching results.

In summary, our approach begins with image retrieval to obtain the adjacency matrix between images. Based on this matrix, we generate the scheduling order for images. Utilizing GPU-accelerated Cascade Hashing expedites the feature matching process, significantly reducing data transfer during matching. Subsequently, we employ local geometric constraints to filter initial matches, avoiding significant time consumption in the RANSAC phase due to low precision, and greatly improving matching accuracy. This ultimately realizes accelerated feature matching for large-scale UAV images. We summarize our approach, and the matching process is illustrated in Figure 3. Among them,  $Q_{in}$  is the queue for image scheduling into the video memory;  $Q_{out}$  is the queue for deleting images from the video memory;  $Q_{init\_match}$  is the initial matching result queue;  $Q_{result}$  is the result queue after gross error elimination;  $Num_{max}$  is the maximum number of images that the video memory can accommodate;  $Num_{cur}$  is the current number of images in the video memory;  $S_{pairs}$  is the total number of image pairs.

## 4. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we evaluate our method using two datasets and compare it with other state-of-the-art approaches. The experiments were conducted on a Lenovo Y9000P laptop equipped with an Intel Core i9-13900HX processor, NVIDIA GeForce RTX 4050 graphics processor, and 16 GB DDR5 memory. The operating system used was Windows 11, and CUDA 11.8 served as our development environment. The experimental platform employed the open-source software COLMAP (Schonberger and Frahm, 2016), taking full advantage of its features such as feature matching, sparse reconstruction, and 3D model evaluation. Subsequently, we will introduce our datasets and present detailed experimental results for the three steps: feature matching, outlier removal, and 3D reconstruction.

### 4.1 Datasets

The datasets used to evaluate our method were captured from two distinct Unmanned Aerial Vehicle (UAV) datasets, namely Campus and SZU. Table 1 provides detailed information about the datasets. In addition to the information provided by the table, it should be noted that the Campus dataset images are captured from university campuses covered with densely packed low-rise buildings, while the SZU dataset images are taken from complex university structures and constitute a set of wide-angle drone image sequences with significant perspective variations. Figure 4 displays sample images from each dataset.

### 4.2 Initial Match and Gross Error Removal

The matching phase is a crucial step in the feature matching process. We showcase the initial matching quantities, time consumption, and outlier removal statistics of our method compared to SIFTGPU. By comparing with existing methods, we aim to highlight the superiority of our approach.

As shown in Table 2, in the Campus dataset, our method demonstrates outstanding performance in the initial matching

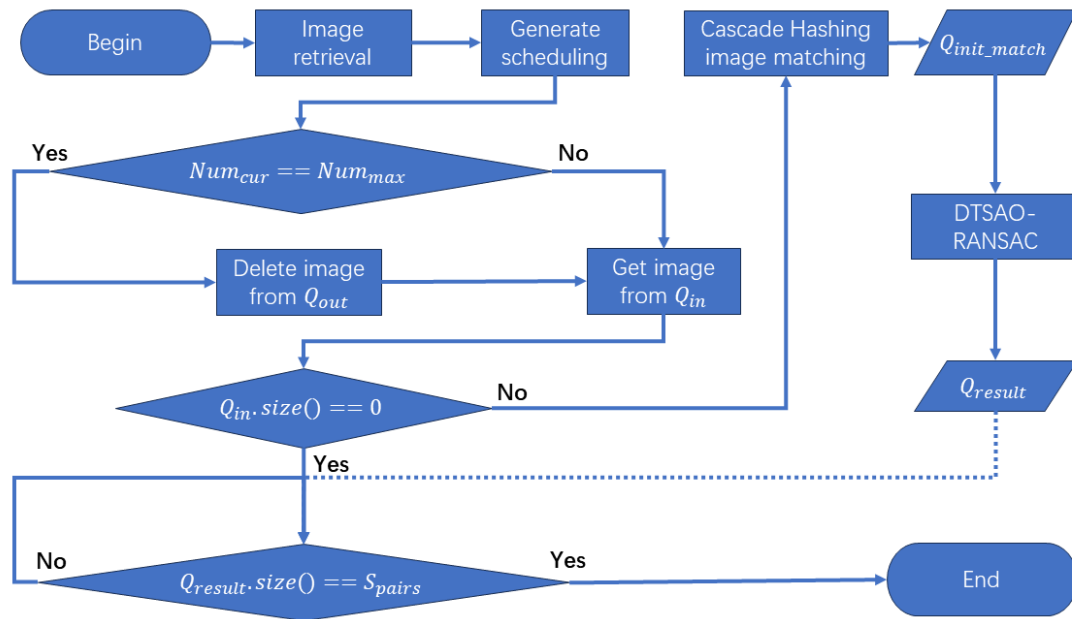


Figure 3. The overall workflow of matching process.

Item	Campus	SZU
UAV type	multi-rotor	multi-rotor
Flight height(m)	80	-
Camera mode	DJI FC6310R	DJI Zenmuse P1
Number of cameras	1	1
Focal length(mm)	24	35
Camera angle(°)	0	-
Number of images	3,743	4,030
Image size(pixel)	5472×3648	8192×5460
GSD(cm)	2.6	1.2

Table 1. Details of the two datasets.

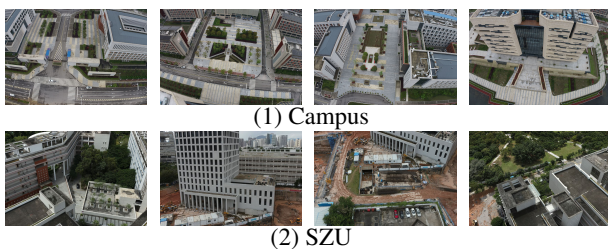


Figure 4. Sample images of the two datasets.

phase, successfully matching approximately 24,770,216 feature points in a mere 3.7 minutes. After coarse outlier removal, around 14,226,818 matching points are retained, with a removal process taking 12.75 minutes, resulting in a final matching precision of 0.57. In comparison, SIFTGPU exhibits a slightly higher initial matching quantity of 26,933,624 points, with a matching time of 8.5 minutes. However, the removal process leads to a reduction in retained matching points to about 11,691,251, taking 21.17 minutes, and a final matching precision of 0.43, which is slightly inferior.

In the SZU scene, our method also performs remarkably well. The initial matching quantity is 41,249,886 points, accomplished in 3.8 minutes. After coarse outlier removal, approximately 28,152,752 matching points are retained, with the removal

(1) Campus			
Item	-	Ours	SIFTGPU
Initial match	Num	24,770,216	26,933,624
	Time(min)	3.7	8.5
After gross error removal	Num	14,226,818	11,691,251
	Time(min)	12.75	21.17
Precision	-	0.57	0.43
(2) SZU			
Item	-	Ours	SIFTGPU
Initial match	Num	41,249,886	43,063,418
	Time(min)	3.9	8.86
After gross error removal	Num	28,152,752	29,471,618
	Time(min)	10.57	22.38
Precision	-	0.68	0.68

Table 2. Result of match.

process taking 10.57 minutes, resulting in a final matching precision of 0.68. In comparison, SIFTGPU exhibits a slightly higher initial matching quantity of 43,063,418 points, with a matching time of 8.86 minutes. However, the removal process leads to a reduction in retained matching points to about 29,471,618, taking 22.38 minutes, and a final matching precision of 0.68, which is comparable to our method.

Overall, our method shows superior matching performance in various scenarios, especially in campus scenarios, highlighting its obvious performance advantages. In the SZU scene, both methods show comparable matching accuracy, but our method excels in error elimination, emphasizing its robustness and adaptability, further highlighting its superior performance in 3D reconstruction. As shown in Figure 5 and 6, we show the results of feature matching on images using our method. (1) is the result of the initial matching, and (2) is the result of the initial matching after DTSAO-RANSAC. It can be seen that after using DTSAO-RANSAC to eliminate gross errors in the matching



results, most of the unmatched pairs have been eliminated. The filtering is successful, which indicates that our method has good robustness.

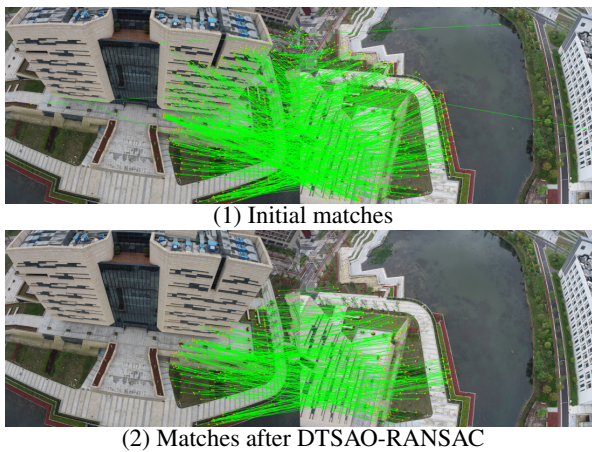


Figure 5. Matches by Ours (Campus).

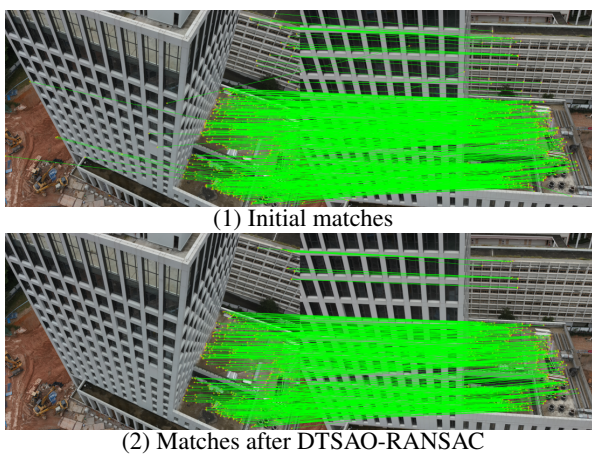


Figure 6. Matches by Ours (SZU).

### 4.3 3D Reconstruction

In the final stage, we conducted parallelized reconstruction on the completed matches to obtain the reconstructed results. In this section, we specifically compare our method with alternative approaches in terms of the successfully fused point cloud quantity and image quantity during parallelized reconstruction. This comparative analysis serves to evaluate the superiority of our method in producing high-quality 3D models.

(1) Campus		
Item	Ours	SIFTGPU
Registered images	3,732	3,737
Points	1,071,340	1,136,424
(2) SZU		
Item	Ours	SIFTGPU
Registered images	4,029	4,028
Points	1,472,216	1,463,596

Table 3. Result of 3D reconstruction.

As shown in Table 3, In both scenarios, the number of registered images and the quantity of generated point clouds by our method are comparable to SIFTGPU. Specifically, in terms of feature matching, our method achieves nearly twice

the matching speed of SIFTGPU, and the three-dimensional reconstruction results exhibit equivalent point cloud quality compared to SIFTGPU. This indicates that our method performs exceptionally well in terms of speed while ensuring high-quality 3D reconstruction.

Overall, compared to alternative methods in both scenarios, our method demonstrates outstanding performance in feature point matching and parallelized reconstruction. This highlights the significant advantage of our method in generating high-quality 3D models. Figure 7 showcases the 3D models reconstructed based on our method. With faster matching speeds, our method surpasses traditional approaches in terms of fused image quantity and key point quantity across different datasets, emphasizing the superiority of our proposed method in reconstruction outcomes.

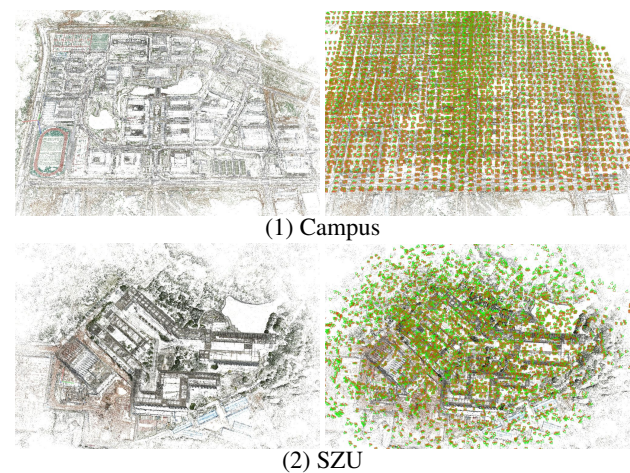


Figure 7. 3D reconstruction model: (left) Point cloud model; (right) Position of the camera.

## 5. CONCLUSIONS AND FUTURE STUDIES

This study introduces an efficient feature matching algorithm for large-scale images, enhancing the efficiency and accuracy of image feature matching through three key steps: data scheduling, GPU-accelerated Cascade Hashing, and local geometric constraints. Firstly, in the data scheduling phase, we successfully reduced the number of ineffective match pairs and minimized data transfer costs through image retrieval and scheduling order generation strategies. The adoption of a cyclic matching strategy effectively addressed issues related to insufficient memory, ensuring the completion of matches for all image pairs. Secondly, in the Cascade Hashing phase, we fully utilized the parallel computing capabilities of the GPU to accelerate image feature matching using the Cascade Hashing algorithm. This not only improved matching speed but also enhanced accuracy and robustness through judicious strategy and parameter settings. Lastly, the introduction of the concept of local geometric constraints led to the development of the DTSO-RANSAC algorithm. By employing Delaunay triangulation and SAO constraints, we successfully filtered out outliers in the initial matches, further increasing the reliability of matching results. Matching extension and RANSAC-based refinement ensured recovery from missed matches and global geometric constraints on matching results. In summary, our algorithm outperforms existing methods in terms of both speed and accuracy, providing a promising solution for large-scale image feature matching. Experimental results demonstrate satisfactory performance at each step, showcasing the synergistic effects of image retrieval,

data scheduling, GPU acceleration, and local geometric constraints. This study not only contributes to the field of computer vision but also holds potential implications for various domains such as image retrieval and surveillance systems.

For future research, we aim to further optimize the algorithm, especially in adapting to different data characteristics and scenes to enhance its universality. Additionally, we plan to expand our algorithm for more complex matching scenarios and larger datasets to meet the demands of practical applications.

## 6. Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant No. 42371442), the Hubei Provincial Natural Science Foundation of China (Grant No. 2023AFB568), the Open Research Fund from the Hubei LuoJia Laboratory (Grant No. 230100013), and the Research Foundation for Talented Scholars of Anhui Polytechnic University (Grant No. 2021YQQ013).

## References

- Cao, M., Jiang, H., Zhao, H., 2023. Hash Indexing-Based Image Matching for 3D Reconstruction. *Applied Sciences*, 13(7), 4518.
- Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H., 2014. Fast and accurate image matching with cascade hashing for 3d reconstruction. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–8.
- Fan, B., Kong, Q., Wang, X., Wang, Z., Xiang, S., Pan, C., Fua, P., 2019. A performance evaluation of local features for image-based 3D reconstruction. *IEEE Transactions on Image Processing*, 28(10), 4774–4789.
- Fischler, M. A., Bolles, R. C., 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.
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., Scholkopf, B., 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4), 18–28.
- Jgou, H., Perronnin, F., Douze, M., Snchez, J., Prez, P., Schmid, C., 2012. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9), 1704–1716.
- Jiang, S., Jiang, W., Guo, B., 2022a. Leveraging vocabulary tree for simultaneous match pair selection and guided feature matching of UAV images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187, 273–293.
- Jiang, S., Jiang, W., Guo, B., Li, L., Wang, L., 2021a. Learned Local Features for Structure From Motion of UAV Images: A Comparative Evaluation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 10583–10597.
- Jiang, S., Jiang, W., Li, L., Wang, L., Huang, W., 2020. Reliable and efficient UAV image matching via geometric constraints structured by Delaunay triangulation. *Remote Sensing*, 12(20), 3390.
- Jiang, S., Jiang, W., Wang, L., 2021b. Unmanned aerial vehicle-based photogrammetric 3d mapping: A survey of techniques, applications, and challenges. *IEEE Geoscience and Remote Sensing Magazine*, 10(2), 135–171.
- Jiang, S., Li, Q., Jiang, W., Chen, W., 2022b. Parallel Structure From Motion for UAV Images via Weighted Connected Dominating Set. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–13.
- Jiang, S., Ma, Y., Liu, J., Li, Q., Jiang, W., Guo, B., Li, L., Wang, L., 2023. Efficient Match Pair Retrieval for Large-scale UAV Images via Graph Indexed Global Descriptor. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.
- Li, X., Yang, J., Ma, J., 2021. Recent developments of content-based image retrieval (CBIR). *Neurocomputing*, 452, 675–689.
- Liu, Y., Xu, X., Li, F., 2018. Image feature matching based on deep learning. *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, IEEE, 1752–1756.
- Malkov, Y. A., Yashunin, D. A., 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.
- Maltezos, E., Doulamis, N., Doulamis, A., Ioannidis, C., 2017. Deep convolutional neural networks for building extraction from orthoimages and dense image matching point clouds. *Journal of Applied Remote Sensing*, 11(4), 042620–042620.
- Muja, M., Lowe, D. G., 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11), 2227–2240.
- Schonberger, J. L., Frahm, J.-M., 2016. Structure-from-motion revisited. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.
- Silpa-Anan, C., Hartley, R., 2008. Optimised kd-trees for fast image descriptor matching. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1–8.
- Wu, C., 2013. SiftGPU: A GPU Implementation of Scale Invariant Feature Transform SIFT.
- Xu, T., Sun, K., Tao, W., 2017. Gpu accelerated image matching with cascade hashing. *Computer Vision: Second CCF Chinese Conference, CCCV 2017, Tianjin, China, October 11–14, 2017, Proceedings, Part I*, Springer, 91–101.
- Zhang, Q., Zheng, S., Zhang, C., Wang, X., Li, R., 2023. Efficient large-scale oblique image matching based on cascade hashing and match data scheduling. *Pattern Recognition*, 138, 109442.