8th Place Solution for Image Matching Challenge 2023

Alexander Veicht
ETH Zürich
veichta@student.ethz.ch

Andri Horat
ETH Zürich
horatan@student.ethz.ch

Felix Yang
ETH Zürich
fyang@student.ethz.ch

Deep Desai
ETH Zürich
ddesai@student.ethz.ch

Philipp Lindenberger
ETH Zürich
philipp.lindenberger@inf.ethz.ch

June 14, 2023

Abstract

We present a novel, price-eligible solution for the Image Matching Challenge 2023. Our solution consists of an ensemble using ALIKED, DISK and SIFT together with LightGlue, a novel and light-weight matcher. Using PixSfM as well as localization of unregistered images with Hierarchical-Localization further improved the final score. Our approach placed $8th$ while matching the score of second place when using an ensemble with DISK and SuperPoint.

1 Introduction

First of all, we would like to thank the organizers for hosting this fun and interesting challenge, we have greatly enjoyed it! We also greatly enjoy reading about the solution of other teams, who present captivating and innovative concepts through their fantastic works.

In the last weeks we have been pushing for a prize-eligible version that could compete against solutions using SuperPoint (SP) [4] and SuperGlue (SG) [9]. In order to achieve this, we tried out various replacements for SP such as DISK [10] and ALIKED [12] and moved from SuperGlue to LightGlue (LG), a cheaper and more accurate local feature matcher developed at ETHZ. It will be released in the upcoming days under the APACHE license. While LG provided very promising results (See Section 3), we were unable get a satisfactory score without using SP until the very last submission on the last day. This last submission, using an ensemble of ALIKED, DISK and SIFT, gave us enough confidence to choose it as our final submission. However, our best scoring submission would have been an ensemble using DISK, SIFT with SP which would have matched the score of second place. Additionally, we had another submission that was able to match the score of $5th$ place on the private leaderboard but did not produce convincing train nor public scores. A comparison of different configurations is shown in Table 1 as well as a per-scene comparison in Appendix A.

Features	Matchers	Train	Public	Private
ALIKED*	LG	0.763	0.361	0.407
ALIKED+SIFT*	LG+NN	0.594	0.434	0.480
DISK*	LG	0.761	0.386	0.437
DISK+SIFT*	LG+NN	**0.843**	0.438	0.479
ALIKED2K+DISK*	LG+LG	0.837	0.444	0.488
ALIKED2K+DISK+SIFT†	LG+LG+NN	0.837	**0.475**	0.523
ALIKED2K+DISK+SIFT	LG(h)+LG(h)+NN	0.824	0.450	**0.529**
DISK+SP	LG+LG	0.876	0.484	**0.562**
DISK+SP*	LG+SG	0.880	0.498	0.517
DISK+SIFT+SP*	LG+NN+LG	**0.890**	**0.511**	0.559
DISK+SIFT+SP*	LG+NN+SG	0.867	T/o	T/o

Table 1: Comparison of Configurations. (†) is used for our final submission and (*) where late submissions. LG(h) has an increased matching threshold of LightGlue of 0.2 (default is 0.1). All configurations use PixSfM and shared cameras if applicable as well as LightGlue unless stated otherwise. Train scores are reported with rotations.

2 Method

We developed a modular pipeline that can be called with various arguments, enabling us to try out different configurations and combine methods very easily. In our pipeline, we made heavy use of hloc [8], which we used as a starting point.

2.1 Image Retrieval

To avoid matching all image pairs of a scene in an exhaustive manner, we used NetVLAD [2] to retrieve the top k images to construct our image pairs. We also tried out CosPlace [3] but did not observe any notable improvements over NetVLAD. Depending on the configuration of each run, we either used $k = 20$, 30 or 50 due to run time constraints. For our final submission, we used $k = 30$.

2.2 Feature Extraction

For keypoint extraction, we combined and tried multiple alternatives. For all feature extractions, we experimented with different image sizes but finally settled on resizing the larger edge to 1600 as it provided the most robust scores:

- ALIKED [12]: We played around with a few settings and finally chose to add it to our ensemble as it showed promising results on a few train scenes. We had to limit the number of keypoints to 2048 due to run-time limitations.
- DISK [10]: DISK was the most promising replacement for SP. We tried a few different configurations and finally settled with the default using a max of 5000 keypoints.
- SIFT [6]: Due to its rotation invariance and fast matching, adding sift to our ensemble turned out to boost performance, especially for heritage/dioscuri and heritage/cyprus.
- SP [4]: SuperPoint was the best-performing features extractor in all our experiments, however, we did not choose it for our final submission because of its restrictive license.

2.3 Feature Matching

We used NN-ratio to match SIFT features. For the deep features such as DISK, ALIKED and SP, we trained LightGlue on the MegaDepth dataset.

2.4 Ensembles

The ensembles gave us the biggest boost in the score. It allowed us to run extraction and matching for different configurations and combine the matches of all configurations. This basically gives us the benefits of all used methods. The only drawback is the increased run-time and we thus had to decrease the number of retrievals. Adding SIFT was always a good option because it did not increase the run-time by much while helping to deal with rotations.

2.5 Structure-from-Motion

For the reconstruction we used PixSfM [5] and forced COLMAP to use shared camera parameters for some scenes.

2.5.1 Pixel-Perfect-SfM

We added PixSfM (after compiling a wheel for manylinux, following the build pipeline of pycolmap) as an additional refinement step to the reconstruction process. During our experiments, we noted that using PixSfM decreased the score on scenes with rotated images as the S2DNet features are not rotation invariant. We thus only used it if no rotations are found in the scene. Due to the large number of keypoints in our ensemble, we had to use the low memory configuration in all scenes, even on the very small ones.

2.5.2 Shared Camera Parameters

We noticed that most scenes have been taken with the same camera and therefore decided to force COLMAP to use the same camera for all images in a scene if all images have the same shape. This turned out to be especially valuable on the haiper scenes where COLMAP assigned multiple cameras.

2.6 Localizing Unregistered Images

Some images were not registered, even with a high number of matches to registered ones, possibly because the assumption of shared intrinsics was not always valid. We, therefore, introduced a post-processing step where we used the hloc toolbox to estimate the pose of unregistered images. Specifically, we checked if the camera of an unregistered image is already in the reconstruction database. If that was not the case, we would infer it from the exif data.

3 Results

3.1 Leaderboard Submission

We noticed that our rotation correction method decreased the performance on the test set and thus opted to submit all runs without rotation prediction.

Features	Matchers	PixSfM	Train	Public	Private
SP	SG		0.790	0.398	0.460
SP	LG		0.790	0.384	0.471
SP	SG	✓	**0.814**	**0.401**	0.467
SP	LG	✓	0.813	**0.401**	**0.476**

Table 2: Ablation Study on SuperGlue, LightGlue and PixSfM

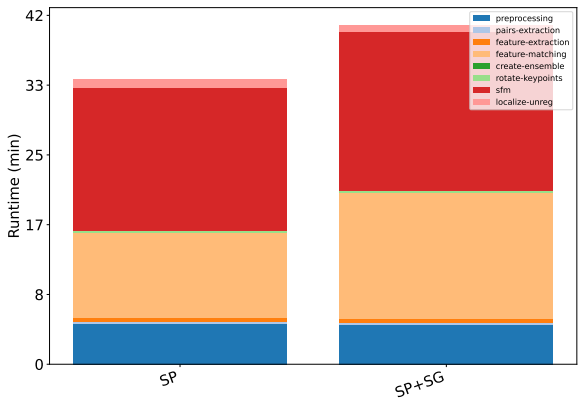

Figure 1: Training run-time comparison between LightGlue (left) and SuperGlue (right).

This is possibly due to wrong rotations as well as PixSfM not being used in the case of rotations. A comparison of different configurations is shown in Table 1.

3.2 Ablation Study on SuperGlue, LightGlue and PixSfM

In order to evaluate the performance impact of LG compared to SG as well as using PixSfM, we evaluated the pipeline in multiple configurations. A comparison is shown in Table 2. We see that LG matches the score on the training and public test set and outperforms SG on the private one by 0.09 when paired with PixSfM. Using PixSfM with its low memory configuration also boosts performance in almost all cases.

Another benefit of using LG over SG is the runtime as shown in Figure 1. LG is much faster in the matching part, offsetting the additional time needed to run PixSfM.

4 Other things we tried

4.1 Image Rotations

Scenes like Cyprus and Dioscuri were very challenging as many deep features are not invariant to rotations. We, therefore, used a pre-trained vision transformer [7] to rotate images by a multiple of 90 degrees when necessary. We then either rotated the keypoints back to the original orientation before starting SfM or we rotated the camera poses in 3D after SfM. As discussed before, the image rotations helped a lot on the train set but not on the test set. We think that it broke on the zoom-in images because on these images it is often impossible to estimate the rotation by eye. On Dioscuri we had at least 9 zoom-in images where the rotation prediction was wrong.

4.2 Smaller things

- **Back-rotating the camera pose:** Rotating back the resulting camera pose at the end of the pipeline worked. However, it did not improve upon back-rotating the features. It also resulted in not being able to use shared camera intrinsics in some scenes.

- **Cropping:** Inspired by last year's solutions, we used cropping to focus matching on important regions between image pairs. The idea was to provide more and stronger matches across image pairs that have a large scale difference. Unfortunately, it became infeasible in terms of run-time and disk space as we ended up with a different set of keypoints and matches for each image pair.

- **Orientation from SIFT matches:** Since SIFT features are rotation-invariant, fast to extract and match (even exhaustively), we tried to recover the image rotations from SIFT matches. To this end, we estimated the relative in-plane rotation pairwise from sift matches and then estimated the rotation for each image by propagating the rotation through the maximum spanning tree of pairwise matches. This worked on Dioscuri but showed its limitations on cyprus.

- **Other feature extractors and matchers:** We tried other dense matchers such as LoFTR, DKM with the hloc integration for SfM. They did not improve the results and/or were too slow.

5 Acknowledgments

We would like to thank Philipp Lindenberger for his awesome guidance, tips, and support. We also want to give a huge credit to his novel matcher LightGlue [11]. We also want to thank the Computer Vision and Geometry Group, ETH Zurich [1] for the awesome project that started all this.

References

[1] https://cvg.ethz.ch. 4

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 2

[3] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4878–4888, 2022. 2

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 1, 2

[5] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5987–5997, 2021. 2

[6] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 2

[7] Subhadip Maji and Smarajit Bose. Deep image orientation angle detection. *arXiv preprint arXiv:2007.06709*, 2020. 3

[8] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 1

[9] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 1

[10] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 1, 2

[11] XX. Xx. In *XX*, page XX, 2023. 4

[12] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 2023. 1, 2

A Per Scene Train Scores

Features	Matches	Cyprus	Dioscuri	Wall	Overall
ALIKED	LG	0.850	0.684	0.967	0.833
DISK	LG	0.314	0.592	0.843	0.583
ALIKED+SIFT	LG+NN	0.991	0.772	0.436	0.733
DISK+SIFT	LG+NN	0.993	0.624	0.756	0.791
ALIKED2K+DISK	LG+LG	0.792	0.712	0.930	0.811
ALIKED2K+DISK+SIFT†	LG+LG+NN	**0.993**	0.802	0.595	0.796
ALIKED2K+DISK+SIFT	LG(h)+LG(h)+NN	**0.993**	0.802	0.595	0.796
DISK+SP	LG+LG	0.862	0.721	0.865	0.816
DISK+SP	LG+SG	0.806	0.735	**0.979**	0.840
DISK+SIFT+SP	LG+NN+LG	0.967	0.815	0.806	**0.862**
DISK+SIFT+SP	LG+NN+SG	0.978	**0.820**	0.777	0.858

Table 3: Scores per scene for Heritage.

Features	Matchers	bike	chairs	fountain	Overall
ALIKED	LG	0.431	0.735	**0.998**	0.721
DISK	LG	0.926	0.799	**0.998**	0.908
ALIKED+SIFT	LG+NN	0.579	0.931	**0.998**	0.836
DISK+SIFT	LG+NN	0.917	0.929	**0.998**	0.948
ALIKED2K+DISK	LG+LG	0.918	0.812	**0.998**	0.909
ALIKED2K+DISK+SIFT†	LG+LG+NN	0.922	0.801	**0.998**	0.907
ALIKED2K+DISK+SIFT	LG(h)+LG(h)+NN	0.920	0.934	**0.998**	0.951
DISK+SP	LG+LG	**0.928**	**0.973**	**0.998**	**0.966**
DISK+SP+SG	LG+SG	**0.928**	0.968	**0.998**	0.965
DISK+SIFT+SP	LG+NN+LG	0.922	0.968	**0.998**	0.962
DISK+SIFT+SP+SG	LG+NN+SG	0.924	0.972	**0.998**	0.964

Table 4: Scores per scene for Haiper.

Features	Matchers	kyiv-puppet-theater	Overall
ALIKED	LG	0.735	0.735
DISK	LG	0.793	0.793
ALIKED+SIFT	LG+NN	0.215	0.215
DISK+SIFT	LG+NN	0.789	0.789
ALIKED2K+DISK	LG+LG	0.742	0.742
ALIKED2K+DISK+SIFT†	LG+LG+NN	0.806	0.806
ALIKED2K+DISK+SIFT	LG(h)+LG(h)+NN	0.824	0.824
DISK+SP	LG+LG	**0.846**	**0.846**
DISK+SP+SG	LG+SG	0.836	0.836
DISK+SIFT+SP	LG+LG	0.846	0.846
DISK+SIFT+SP+SG	LG+SG	0.778	0.778

Table 5: Scores per scene for Urban.