Supplementary Material for: Grounding Image Matching in 3D with MASt3R

Vincent Leroy, Yohann Cabon, and Jerome Revaud

Naver Labs Europe
<firstname>.<lastname>@naverlabs.com

In this supplementary material, we first present additional qualitative examples on various tasks in Sec. 1, followed by a proof of convergence of the fast reciprocal matching algorithm and an in-depth study of the related performance gains in Sec. 2. We finally show an ablative study concerning the impact of *coarse-to-fine* matching in Sec. 3.

Fig. 1: Qualitative MVS results on the DTU dataset [1] simply obtained by triangulating the dense matches from MASt3R.

1 Additional Qualitative Results

We provide here additional qualitative results on the DTU [1], InLoc [7], Aachen Day-Night datasets [9] and the Map-free benchmark [2].

MVS on DTU. We show in Fig. 1 the output point clouds after post-processing, shaded with approximate normals from the tangent planes based on the 50 nearest neighbors. We wish to emphasize again that the point clouds are raw values obtained via triangulation of the *coarse-to-fine* matches of MASt3R. The matching was performed in an one-versus-all strategy, meaning that we did not leverage the epipolar constraints coming from the GT cameras, which is in stark contrast with all existing approaches for MVS. MASt3R is particularly precise and robust, giving sharp and dense details. The reconstructions are complete even in low-contrast homogeneous regions like the surfaces of the vegetables or the sides of the power supply. The matching is also robust to varied textures or materials, and also to violations of the Lambertian assumption, *i.e.* specularities on the vegetables, plastic surfaces or the white sculpture.

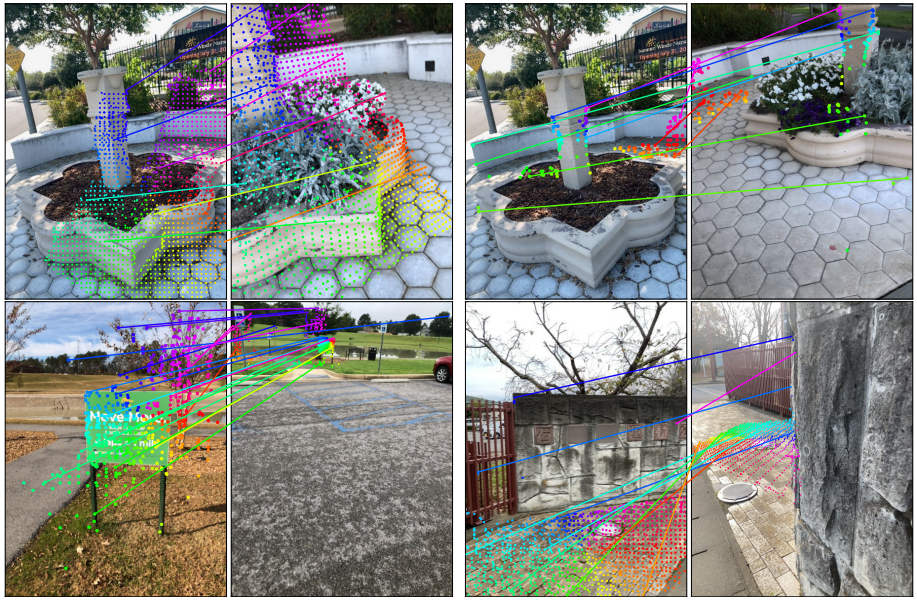

Fig. 2: Qualitative examples of matching on Map-free localization benchmark.

Qualitative matching results. We show a few examples of matches Fig. 2 for the Map-free benchmark [2], in Fig. 3 for the InLoc [7] dataset and in Fig. 4 for the Aachen Day-Night dataset [9]. The proposed MASt3R approach is robust to extreme viewpoint changes, and still provides approximately correct correspondences in such cases (right-hand side pairs of Map-free in Fig. 2), even for views facing each other (coffee tables or corridor pairs of InLoc 3). This is reminiscent of the capabilities of DUSt3R that provided an unprecedented robustness to such cases. Similarly, our approach handles large scale differences (*e.g.* on Map-free in Fig. 2) repetitive and ambiguous patterns, as well as environmental and day/night illuminations changes (Fig. 4). Interestingly, the accuracy of cor-

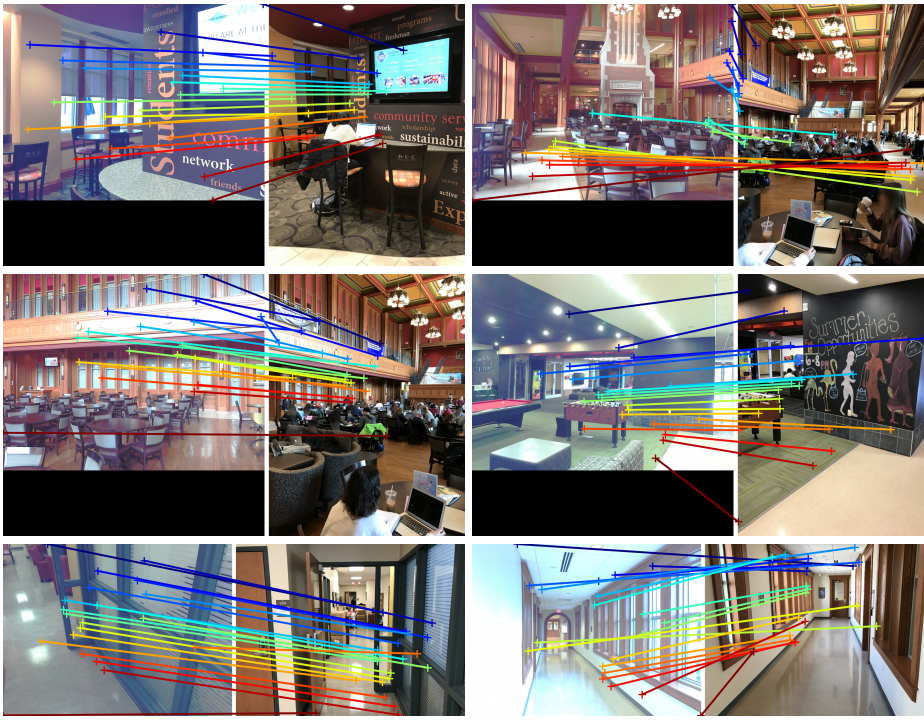

Fig. 3: Qualitative examples of matching on the InLoc localization benchmark.

respondences output by MASt3R gracefully degrades when the viewpoint baseline increases. Even in extreme cases where correspondences get very coarsely estimated, approximately correct relative camera poses can still be recovered. Thanks to these capabilities, MASt3R reach state-of-the-art performance or close to it on several benchmarks in a zero-shot setting. We hope this work will foster research in the direction of pointmap regression for a multitude of vision tasks, where robustness and accuracy are critical.

2 Fast Reciprocal Matching

2.1 Theoretical study

We detail here the theoretical proofs of convergence of the Fast Reciprocal Matching algorithm presented in Sec.3.3 of the main paper. Contrary to the traditional bipartite graph matching formulation [3], where the complete graph is used for the matching, we wish to decrease the computational complexity by calculating only a smaller portion of it. As explained in equation (14) of the main paper, considering the two predicted sets of features D^1, $D^2 \in \mathbb{R}^{H \times W \times d}$, partial reciprocal matching boils down to finding a subset of the reciprocal correspondences, *i.e.* mutual Nearest Neighbors (NN):

Fig. 4: Qualitative examples of matching on the Aachen Day-Night localization benchmark. Pairs from the day subset are on the left column, and pairs from the night subset are on the right column.

$$\mathcal{M} = \{(i,j) \mid j = \mathrm{NN}_2(D_i^1) \text{ and } i = \mathrm{NN}_1(D_j^2)\}, \tag{1}$$

$$\text{with } \mathrm{NN}_A(D_j^B) = \arg\min_i \left\| D_i^A - D_j^B \right\|. \tag{2}$$

We remind here the behavior of the algorithm: an initial set of k pixels of I^1, $U^0 = \{U_n^0\}_{n=1}^k$ with $k \ll WH$, is mapped to their NN in I^2, yielding V^1, that are then mapped to their nearest neighbors back to I^1:

$$U^t \longmapsto [\mathrm{NN}_2(D_u^1)]_{u \in U^t} \equiv V^t \longmapsto [\mathrm{NN}_1(D_v^2)]_{v \in V^t} \equiv U^{t+1} \tag{3}$$

After this back-and-forth mapping, the reciprocal matches (*i.e.* those which form a cycle) are recovered and removed from U^{t+1}. The remaining "active" ones are mapped back to I^2 and reciprocity is checked again. We iterate this process for a few iterations. After enough iterations we discard any active sample remaining.

It is important to note that the NN algorithm we use is deterministic and consistently returns the same index in the case where multiple descriptors in the

other image share the same minimal distance (or maximal similarity), although this is very unlikely since descriptors are real-valued.

Proof of Convergence. By design, Fast Reciprocal Matching (FRM) operates on the directed bipartite graph $\mathcal{G}$ of nearest neighbors between I^1 and I^2. $\mathcal{G}$ contains oriented edges $\mathcal{E}$. All nodes, *i.e.* pixels, belong to $\mathcal{G}$ since we add an edge for each pixel's nearest neighbor, but note that all pixels cannot reach all other pixels. For example, two reciprocal pixels in I^1 and I^2 are only connected to each other and to no other pixels. This means $\mathcal{G}$ is composed of possibly multiple disjoint sub-graphs $\mathcal{G}^i, 1 \leq i \leq HW$ with directed edges $\mathcal{E}^i$, as depicted in Fig. 5.

Proposition 1. *There can be only one cycle in each sub-graph $\mathcal{G}^i$.*

Proof. This is a rather trivial fact, since we build $\mathcal{G}$ s.t. only one edge exits each node. If one were to follow the path of a sub-graph $\mathcal{G}^i$, once a node that belongs to a cycle is reached, no edge can exit the cycle, for the only exiting edge is already part of the cycle. A second cycle (or more) thus cannot exist in $\mathcal{G}^i$.

Lemma 1. *Each of the subgraph $\mathcal{G}^i$ is either a single cycle or a special arborescence,* i.e. *a directed graph where, from any node there exist a single path towards a root cycle.*

Proof. The former follows naturally from the previous explanation: since there can only be a single cycle in $\mathcal{G}^i$, it can naturally be a cycle. We now demonstrate the latter, *i.e.* when $\mathcal{G}^i$ is not trivially a cycle. Let us march on $\mathcal{G}^i$ starting from an arbitrary node a, to which is attached a descriptor D^1_a. The only edge exiting this node goes to its nearest neighbor $NN_2(D^1_a) = b$. Now at node b, we do the same and follow the only edge exiting back to I^1: $NN_1(D^2_b) = c$. Alternating between I^1 and I^2, we get $NN_2(D^1_c) = d$, $NN_1(D^2_d) = e$ and so forth. We denote $s(u, v) = D^{1\top}_u D^2_v$ the similarity score of an edge between two nodes u and v, $(u, v) \in \mathcal{E}^i$. Because edges are nearest neighbors, we note that $s(a, b) \leq s(c, b)$. This trivially stems from the fact that if $s(c, b) < s(a, b)$ then the nearest neighbor of b would no longer be c but at least a. Expanding this property to the path along $\mathcal{G}^i$ it follows that:

$$s(a, b) \leq s(c, b) \leq s(c, d) \leq s(e, d)... \tag{4}$$

Meaning that the similarity score monotonously increases as we walk along the graph. There is a finite number of nodes in $\mathcal{G}^i$ so this sequence reaches the upper-bound similarity value $s(u, v)$. Because $s(u, v)$ is the maximal similarity in $\mathcal{G}^i$, this ensures that $NN_2(D^1_u) = v$ and $NN_1(D^2_v) = u$ forming a cycle of at least two nodes. This means there is always a cycle in $\mathcal{G}^i$, between the maximal similarity pair. Following Proposition 1, we can conclude that there is no other cycle in $\mathcal{G}^i$ and that each starting point is thus guaranteed to lead towards the root via a single path, forming an arborescence with a cycle at its root.

Note that the root cycle can be of more than two nodes if more than one greatest similarity of Eq. (4) are perfectly equal and the NN algorithm creates

a greater cycle. Because $\mathcal{G}$ is a bipartite graph, $\mathcal{G}^i$ is also bipartite, meaning the end-cycle is composed of an even number of nodes. In practice however, we work with floating-point descriptors of dimension 24. For greater cycles to exist, *e.g.* cycles of 4 nodes a, b, c, d, the similarities must satisfy increasingly prohibitive constraints, *e.g.* $s(a,b) = s(c,b) = s(c,d) = s(a,d)$. This is extremely unlikely with real-valued distance and we consider it is negligible.

Corollary 1. *Regardless of the starting point in $\mathcal{G}^i$, the FRM algorithm always converges towards reciprocal matches.*

This follows naturally from the above: we did not make any assumption about the starting point of this walk nor about the sub-graph it belongs to. For any starting point in the graph, *i.e.* for all initial pixels U, the FRM algorithm will by design follow the sub-graph of nearest neighbors that will ultimately lead to the root cycle, which is by definition a reciprocal match.

We illustrate this behavior in Fig. 5. In the upper part (pink) the starting point u_0 directly lies in a cycle containing two nodes u_0 and v_0 and the algorithm stops after the first cycle verification at step $t = 1$. The bottom part shows a more complex case of convergence basin, where several starting points u_1, u_2, u_3, u_4 lead to resp. two nodes v_1 and v_2 in I^2. Following the path to the root of the arborescence, and updating U and V along the way, the algorithm finds a cycle between u_1 and v_1 at timestep $t = 1$. From 5 initial pixel positions, the algorithm returned a unique reciprocal correspondence.

Note that it is possible to artificially build a graph that maximizes the number of NN queries thus impacting the computational efficiency, but these are very unlikely in practice as seen in Figure 2 (center) of the main paper. The number of active samples, *e.g.* samples that did not reach a cycle, quickly drops to 0 after only 6 iterations, leading to a significant speed-up in computation (right).

Proposition 2. *Starting from $k \ll HW$ samples, the FRM algorithm recovers a subset $\mathcal{M}_k$ of all possible reciprocal correspondences of cardinality $|\mathcal{M}_k| = j \leq k$.*

Proof. This fact comes trivially from the k sparse initial samples U. As explained before, $\mathcal{G}$ is composed of at most HW sub-graphs $\mathcal{G}^i$. Because we initialize the algorithm with $k \ll HW$ seeds, these can at most span k sub-graphs each leading to a single reciprocal match. Due to the potential presence of convergence basins, as seen in Fig. 5, samples can merge along the paths to their root cycles, decreasing the final number of reciprocals and explaining the inequality $j \leq k$.

2.2 Performance improvement with fast matching

As observed in Figure 2 of the main paper, FRM significantly improves the performance. In the minimal example we provide in Fig. 5, it is clearly visible that the FRM provides a sampling biased towards finding reciprocal matches with large basins (bottom), since a greater number of initial samples can fall onto them compared to small basins (top). Note that the size of the basin is inversely proportional to the maximal density of reciprocal matches. Interestingly with

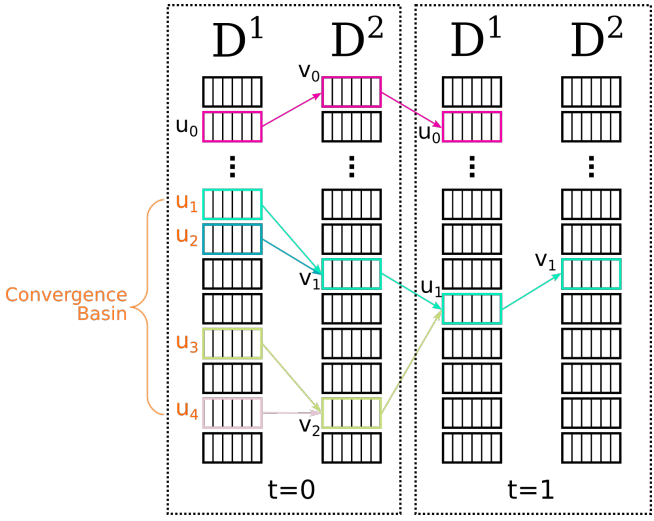

Fig. 5: Illustration of the iterative FRM algorithm. Starting from 5 pixels in I^1 at $t = 0$, the FRM connects them to their Nearest Neighbors (NN) in I^2, and maps them back to their NN in I^1. If they go back to their starting point (top pink), a cycle (reciprocal match) is detected and returned. Otherwise (bottom) the algorithm continues iterating until a cycle is detected for all starting samples, or until the maximal number of iterations is reached. We show in orange the starting points of a *convergence basin*, *i.e.* nodes of a sub-graph for which the algorithm will converge towards the same cycle. For clarity, all edges of $\mathcal{G}$ were not drawn.

the FRM, this results in a more homogeneous distribution (*i.e.* spatial coverage) of reciprocal matches than the full matching, as depicted in Fig. 6. As a direct consequence of a more homogeneous spatial coverage, RANSAC is able to better estimate epipolar lines than when lots of points are packed together in a small image region, which in turn provides better and more stable pose estimates.

In order to demonstrate the effect of basin-biased sampling, we propose to compute the full correspondence set $\mathcal{M}$ (Eq. (1)) and to subsample it in two ways: first, we naively subsample it randomly to reach the same number of reciprocals as the FRM. Second, we compute the size of each basin (as shown in Fig. 7) and we bias the subsampling using the sizes. We report the results of this experiment in Fig. 8. While random subsampling results in catastrophic performance drops, basin-biased sampling actually increases the performance compared to using the full graph (rightmost datapoint). As expected, the FRM algorithm provides a performance that closely follows biased subsampling, yet by only a fraction of the compute compared to basin-biased sampling which requires to compute all reciprocal matches in order to measure basin sizes. Importantly, these observations hold for both reprojection error and pose accuracy, regardless of the variant of RANSAC used to estimate relative poses.

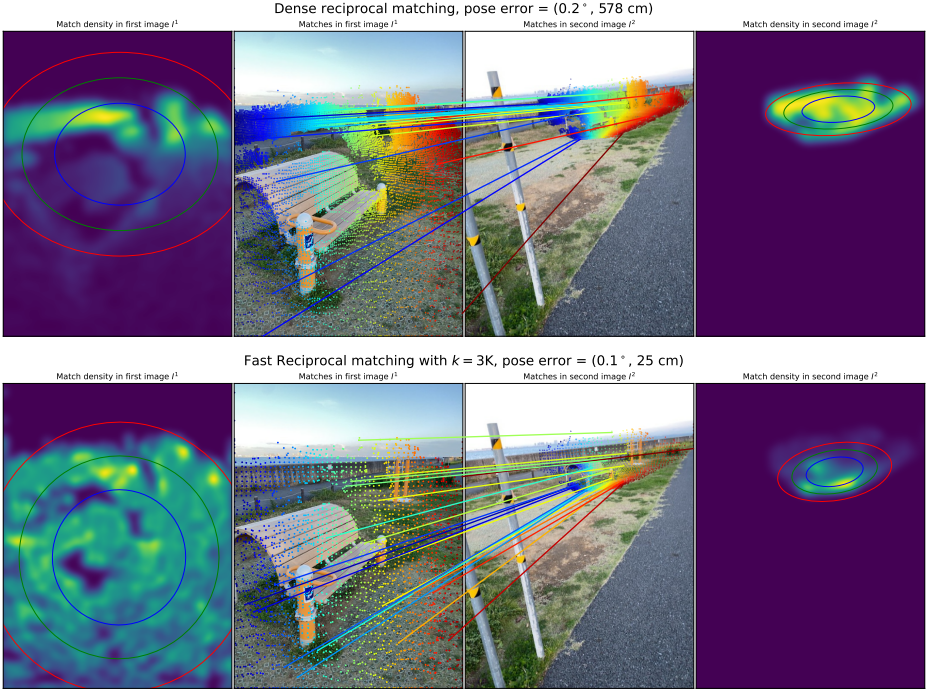

Fig. 6: Illustration of the difference in matching density when using dense reciprocal matching (baseline) and fast reciprocal matching with $k = 3000$. Fast reciprocal matching samples correspondences with a bias for large convergence basins, resulting in a more uniform coverage of the images. Coverage can be measured in terms of the mean and standard deviation σ of the point matches in each density map, plotted as colored ellipses (red, green and blue correspond respectively to $1\sigma, 1.5\sigma$ and 2σ).

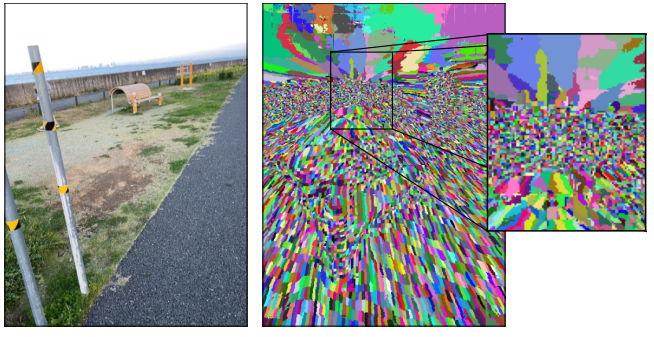

Fig. 7: Illustration of convergence basins for one of the image in Fig. 6. Each basin is filled with the same (random) color. A convergence basin is an area for which any of its point will converge to the same correspondence when applying the fast reciprocal matching algorithm.

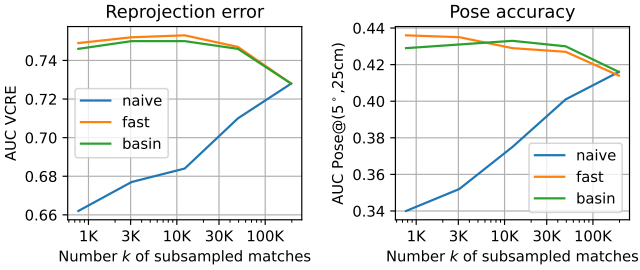

Fig. 8: Comparison of the performance on the Map-free benchmark (validation set) for different subsampling approaches: 'naive' denotes the random uniform subsampling of the original full set of reciprocal matches; 'fast' denotes the proposed fast reciprocal matching; and 'basin' denotes random subsampling weighted by the size of the convergence basin. The 'fast' and 'basin' strategies perform similarly whereas naive subsampling leads to catastrophic results.

Table 1: Coarse matching compared to Coarse-to-Fine for the tasks of visual localization on Aachen Day-Night (left) and MVS reconstruction on the DTU dataset (right).

Methods	Coarse-to-Fine	Day	Night
MASt3R top1	×	74.9/90.3/98.5	55.5/82.2/95.8
MASt3R top1	✓	79.6/93.5/98.7	70.2/88.0/97.4
MASt3R top20	×	80.8/93.8/99.5	74.3/92.1/100
MASt3R top20	✓	83.4/95.3/99.4	76.4/91.6/100

Methods	Acc.↓	Comp.↓	Overall↓
DUSt3R [8]	2.677	0.805	1.741
MASt3R Coarse	0.652	0.592	0.622
MASt3R	0.403	0.344	0.374

3 Coarse-to-Fine

In this section, we showcase the important benefits of the *coarse-to-fine* strategy. We compare it to *coarse-only* matching, that simply computes correspondences on input images down-scaled to the resolution of the network.

Visual localization on Aachen Day-Night [9]. For this task, the input images are of resolution 1600×1200 and 1024×768, in both landscape and portrait are downscaled to $512 \times 384 / 384 \times 512$. We report the percentage of successfully localized images within three thresholds: (0.25m, 2°), (0.5m, 5°) and (5m, 10°) in Tab. 1 (left). We observe significant performance drops when using coarse matching only, by up to 15% in top1 on the Night split.

MVS. The input images of the DTU dataset [1] are of resolution 1200×1600 downscaled to 384×512. As in the main paper, we report here the accuracy, completeness and Chamfer distance of triangulated matches obtained with MASt3R, in the *coarse-only* and *coarse-to-fine* settings in Tab. 1 (right). While coarse matching still outperforms the direct regression of DUSt3R, we see a clear drop in reconstruction quality in all metrics, nearly doubling the reconstruction errors.

Table 2: Absolute camera pose on 7Scenes [6] and Cambridge-Landmarks [4] dataset. We report the median translation and rotation errors $cm/^\circ$.

	7Scenes (Indoor)							Cambridge (Outdoor)				
Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	S. Facade	O. Hospital	K. College	St.Mary's	G. Court
DUSt3R	3/0.97	3/0.95	2/1.37	**3**/1.01	**4**/1.14	**4**/1.34	11/2.84	6/0.26	**17**/0.33	11/0.20	7/0.24	38/0.16
MASt3R	**2/0.81**	**2/0.88**	**1/0.88**	**3/0.95**	**4/1.07**	**4/1.29**	**3/0.95**	**4/0.15**	**17/0.29**	**8/0.14**	**5/0.16**	**13/0.07**

4 Additional Visual localization experiments

In table 2, we provide additional visual localization results on 7Scenes [6] and Cambridge-Landmarks [4]. For these experiments, we use the same parameters as in DUSt3R: we use DUSt3R as a 2D-2D pixel matcher, we leverage the known query intrinsics. We keep the top 20 retrieved images for Cambridge-Landmarks and top 1 for 7Scenes and do not use Coarse-to-Fine for neither method.

5 Limitations

Even though MASt3R yields state-of-the-art performance on multiple benchmarks, it is of course subject to several limitations that we review below.

Reliability of metric depth estimates. To the best of our knowledge, MASt3R is the first method that performs metric depth prediction in binocular settings (*i.e.* existing metric depth estimation methods are monocular). In the binocular setting, depth estimation is arguably a much less ambiguous task, yet it fundamentally remains data-driven and relies on priors, which provides little guarantee, similar to any monocular metric-depth methods.

Robustness to changes. MASt3R is moderately robust to illumination changes, seasonal changes, dynamic/transient objects and/or long-term changes. It can typically accommodate moderate changes (say day/night), but may dramatically fail when confronted with more significant changes (such as summer/winter as in Fig. 9). The reason is that, MASt3R being a purely data-driven approach, it can only handle similar types of changes as to what was seen during training. Logically, the fact that we trained MASt3R with mostly static scenes that are perfectly 3D consistent makes the model learning to reject pairs with small changes, that would look otherwise extremely similar to human observers (*e.g.* the temple pairs in Fig. 9, where the only changes are the illumination and the presence/absence of scaffholds). Among all datasets used for training, perhaps MegaDepth [5] and Niantic's Map-free dataset [2] are the only ones comprising any significant long-term changes. We hypothesize that MASt3R could become significantly more robust if trained with more of such adequate data.

Fig. 9: Example of failure cases due to difficult seasonal changes (top row, cameras are incorrectly estimated) or scene changes (presence of scaffolds in the bottom row).

Problems with coarse-to-fine matching. We observe consistent errors during the fine matching in the presence of repetitive patterns. While fine matching significantly improves over coarse matching in term of precision, it has the downside of losing the global context. Thus, in the presence of repetitive patterns, matching becomes an ill-posed problem in the absence of further clue. An example of such failure is illustrated in Fig. 10.

Scalability. DUSt3R [8] introduced a procedure for globally aligning all pointmaps in the same world coordinate system. Even though this paper only focuses on pairwise matching, the same procedure could be applied, since MASt3R also outputs pointmaps and confidence maps. However, and much like DUSt3R, MASt3R scales poorly to large image collections due to its pairwise nature, making the cost quadratic in the total number of images unless some pruning is used (*e.g.* image retrieval).

6 Detailed experimental settings

In our experiments, we set the confidence loss weight $\alpha = 0.2$ as in [8], the matching loss weight $\beta = 1$, local feature dimension $d = 24$ and the temperature in the InfoNCE loss to $\tau = 0.07$. We report the detailed hyper-parameter settings we use for training MASt3R in Table 3.

Fig. 10: Example of failure cases due to coarse-to-fine matching losing context when matching crops of the two images.

Table 3: Detailed hyper-parameters for the training

Hyper-parameters	fine-tuning
Optimizer	AdamW
Base learning rate	1e-4
Weight decay	0.05
Adam β	(0.9, 0.95)
Pairs per Epoch	650k
Batch size	64
Epochs	35
Warmup epochs	7
Learning rate scheduler	Cosine decay
Input resolutions	512×384, 512×336 512×288, 512×256 512×160
Image Augmentations	Random crop, color jitter
Initialization	DUSt3R [8]

References

1. Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. IJCV (2016)
2. Arnold, E., Wynn, J., Vicente, S., Garcia-Hernando, G., Monszpart, Á., Prisacariu, V.A., Turmukhambetov, D., Brachmann, E.: Map-free visual relocalization: Metric pose relative to a single image. In: ECCV (2022)
3. Cho, M., Lee, J., Lee, K.M.: Reweighted random walks for graph matching. In: ECCV (2010)
4. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a Convolutional Network for Real-Time 6-DOF Camera Relocalization. In: ICCV (2015)
5. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR. pp. 2041–2050 (2018)
6. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.W.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: CVPR (2013)
7. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. PAMI (2019)
8. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy (2023)
9. Zhang, Z., Sattler, T., Scaramuzza, D.: Reference pose generation for long-term visual localization via learned features and view synthesis. IJCV (2021)