# MASt3R-SfM: a Fully-Integrated Solution for Unconstrained Structure-from-Motion

Bardienus Pieter Duisterhof[1] Lojze Zust[2,3] Philippe Weinzaepfel[3]
Vincent Leroy[3] Yohann Cabon[3] Jerome Revaud[3]

[1]Carnegie Mellon University [2]University of Ljubljana [3]Naver Labs Europe

https://github.com/naver/mast3r/

## Abstract

*Structure-from-Motion (SfM), a task aiming at jointly recovering camera poses and 3D geometry of a scene given a set of images, remains a hard problem with still many open challenges despite decades of significant progress. The traditional solution for SfM consists of a complex pipeline of minimal solvers which tends to propagate errors and fails when images do not sufficiently overlap, have too little motion, etc. Recent methods have attempted to revisit this paradigm, but we empirically show that they fall short of fixing these core issues. In this paper, we propose instead to build upon a recently released foundation model for 3D vision that can robustly produce local 3D reconstructions and accurate matches. We introduce a low-memory approach to accurately align these local reconstructions in a global coordinate system. We further show that such foundation models can serve as efficient image retrievers without any overhead, reducing the overall complexity from quadratic to linear. Overall, our novel SfM pipeline is simple, scalable, fast and truly unconstrained, i.e. it can handle any collection of images, ordered or not. Extensive experiments on multiple benchmarks show that our method provides steady performance across diverse settings, especially outperforming existing methods in small- and medium-scale settings.*

## 1. Introduction

Structure-from-Motion (SfM) is a long-standing problem of computer vision that aims to estimate the 3D geometry of a scene as well as the parameters of the cameras observing it, given the images from each camera [18]. Since it conveniently provides jointly for cameras and map, it constitutes an essential component for many practical computer vision applications, such as navigation (mapping and visual localization [10, 35, 46]), dense multi-view stereo reconstruction (MVS) [37, 47, 60, 67], novel view synthesis [6, 22, 34], auto-calibration [17] or even archaeology [38, 55].

In reality, SfM is a "needle in a haystack" type of problem, typically involving a highly non-convex objective function with many local minima [59]. Since finding the global minimum in such a landscape is too challenging to be done directly, traditional SfM approaches such as COLMAP [46] decomposes the problem as a series (or *pipeline*) of minimal problems, *e.g.* keypoint extraction and matching, relative pose estimation, and incremental reconstruction with triangulation and bundle adjustment. The presence of outliers, *e.g.* wrong pixel matches, poses additional challenges and compels existing methods to resort to hypothesis formulation and verification at multiple occasions in the pipeline, typically with RAndom SAmple Consensus (RANSAC) or its many flavors [4, 5, 16, 25, 58, 65]. This approach has been the standard for several decades, yet it remains brittle and fails when the input images do not sufficiently overlap, or when motion (*i.e.* translation) between viewpoints is insufficient [10, 48].

Recently, a set of innovative methods propose to revisit SfM in order to alleviate the heavy complexity of the traditional pipeline and solve its shortcomings. VGGSfM [62],

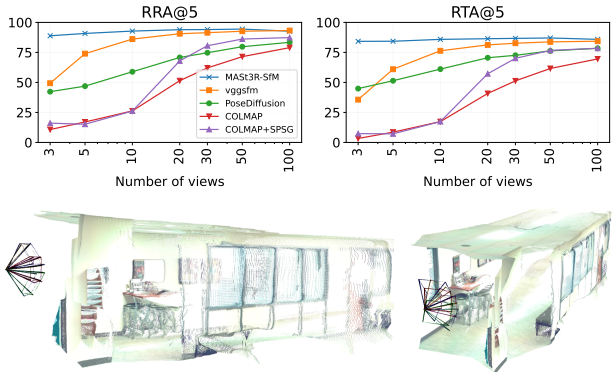


Figure 1. **Top**: Relative rotation (RRA) and translation (RTA) accuracies on the CO3Dv2 dataset when varying the number of input views with random subsampling (the more views, the larger they overlap). In contrast to our competitors, MASt3R-SfM offers nearly constant performance on the full range, even for very few views. **Bottom**: MASt3R-SfM also works *without motion*, *i.e.* in purely rotational settings. We show here a reconstruction from 6 views sharing the same optical center.

for instance, introduces an end-to-end differentiable version of the pipeline, simplifying some of its components. Likewise, detector-free SfM [19] replaces the keypoint extraction and matching step of the classical pipeline with learned components. These changes must, however, be put into perspective, as they do not fundamentally challenge the overall structure of the traditional pipeline. In comparison, FlowMap [50] and Ace-Zero [9] independently propose a radically novel type of approach to solve SfM, which is based on simple first-order gradient descent of a global loss function. Their trick is to train a geometry regressor network during scene optimization as a way to reparameterize and regularize the scene geometry. Unfortunately, this type of approach only works in certain configurations, namely for input images exhibiting high overlap and low illumination variations. Lastly, DUSt3R [26, 64] demonstrates that a single forward pass of a transformer architecture can provide a good estimate of the geometry and cameras parameters of a small two-image scene. These particularly robust estimates can then be stitched together again using simple gradient descent, allowing to relax many of the constraints mentioned earlier. However it yields rather imprecise global SfM reconstructions and does not scale well.

In this work, we propose MASt3R-SfM, a fully-integrated SfM pipeline that can handle completely unconstrained input image collections, *i.e*. ranging from a single view to large-scale scenes, possibly without any camera motion as illustrated in Fig. 1. We build upon the recently released DUSt3R [64], a foundation model for 3D vision, and more particularly on its recent extension MASt3R that is able to perform local 3D reconstruction and matching in a single forward pass [26]. Since MASt3R is fundamentally limited to processing image pairs, it scales poorly to large image collections. To remedy this, we hijack its frozen encoder to perform fast image retrieval with negligible computational overhead, resulting in a scalable SfM method with quasi-linear complexity in the number of images. Thanks to the robustness of MASt3R to outliers, the proposed method is able to completely get rid of RANSAC. The SfM optimization is carried out in two successive gradient descents based on local reconstructions output by MASt3R: first, using a matching loss in 3D space; then with a 2D reprojection loss to refine the previous estimate. Interestingly, our method goes beyond structure-from-motion, as it works even when there is *no motion* (*i.e*. purely rotational case), as illustrated in Fig. 1.

In summary, we make three main contributions. First, we propose MASt3R-SfM, a full-fledged SfM pipeline able to process unconstrained image collections. To achieve linear complexity in the number of images, we show as second contribution how the encoder from MASt3R can be exploited for large-scale image retrieval. Note that our entire SfM pipeline is training-free, provided an off-the-shelf MASt3R checkpoint. Lastly, we conduct an extensive benchmarking on a diverse set of datasets, showing that existing approaches are still prone to failure in small-scale settings, despite significant progress. In comparison, MASt3R-SfM demonstrates state-of-the-art performance in a wide range of conditions, as illustrated in Fig. 1.

## 2. Related Works

**Traditional SfM.** At the core of Structure-from-Motion (SfM) lies matching and Bundle Adjustment (BA). Matching, *i.e*. the task of finding pixel correspondences across different images observing the same 3D points, has been extensively studied in the past decades, beginning from handcrafted keypoints [7, 31, 42] and more recently being surpassed by data-driven strategies [11–14, 21, 41, 43, 52, 63]. Matching is critical for SfM, since it builds the basis to formulate a loss function to minimize during BA. BA itself aims at minimizing reprojection errors for the correspondences extracted during the matching phase by jointly optimizing the positions of 3D points and camera parameters. It is usually expressed as a non-linear least squares problem [2], known to be brittle in the presence of outliers and prone to fall into suboptimal local minima if not provided with a good initialization [1, 51]. For all these reasons, traditional SfM pipelines like COLMAP are heavily handcrafted in practice [19, 29, 46]. By triangulating 3D points to provide an initial estimate for BA, they incrementally build a scene, adding images one by one by formulating hypothesis and discarding the ones that are not verified by the current scene state. Due to the large number of outliers, and the fact that the structure of the pipeline tends to propagate errors rather than fix them, robust estimators like RANSAC are extensively used for relative pose estimation, keypoint track construction and multi-view triangulation [46].

**SfM revisited.** There has been a recent surge of methods aiming to simplify or even completely revisit the traditional SfM pipeline [9, 19, 50, 62, 64]. The recently proposed FlowMap and Ace-Zero, for instance, both rely on the idea of training a regressor network at test time. In the case of FlowMap [50], this network predicts depthmaps, while for Ace-Zero [9] it regresses dense 3D scene coordinates. While this type of approach is appealing, it raises several problems such as scaling poorly and depending on many off-the-shelf components for FlowMap. Most importantly, both methods only apply to constrained settings where the input image collections offers enough uniformity and continuity in terms of viewpoints and illuminations. This is because the regressor network is only able to propagate information incrementally from one image to other tightly similar images. As a result, they cannot process unordered image collections with large viewpoint and illumination disparities. On the other hand, VGGSfM, Detector-Free SfM (DF-SfM) and DUSt3R cast the SfM problem

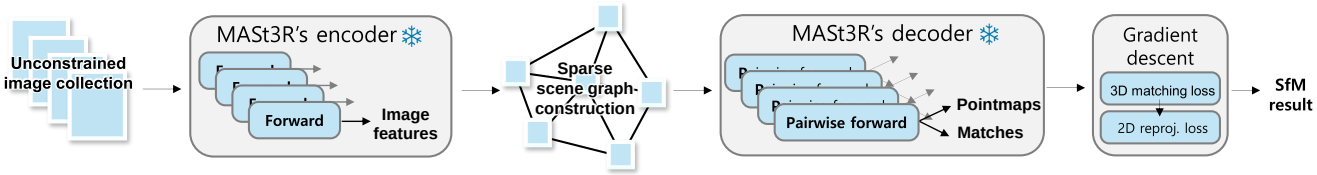


Figure 2. **Overview of the proposed MASt3R-SfM method**. Given an unconstrained image collections, possibly small (1 image) or large (over 1000 images), we start by computing a sparse scene graph using efficient image retrieval techniques given a frozen MASt3R's per-image features. We then compute local 3D reconstruction and matches for each edge using again a frozen MASt3R's decoder. Global optimization proceeds with gradient descent of a matching loss in 3D space, followed by refinement in terms of 2D reprojection error.

in a more traditional manner by relying on trained neural components that are kept frozen at optimization time. VG-GSfM [62], for its part, essentially manages to train end-to-end all components of the traditional SfM pipeline but still piggybacks itself onto handcrafted solvers for initializing keypoints, cameras and to triangulate 3D points. As a result, it suffers from the same fundamental issues than traditional SfM, *e.g.* it struggles when there are few views or little camera motion. Likewise, DF-SfM [19] improves for texture-less scenes thanks to relying on trainable dense pairwise matchers, but sticks to the overall COLMAP pipeline. Finally, DUSt3R [64] is a foundation model for 3D vision that essentially decomposes SfM into two steps: local reconstruction for every image pair in the form of pointmaps, and global alignment of all pointmaps in world coordinates. While the optimization appears considerably simpler than for previous approaches (*i.e.* not relying on external modules, and carried out by minimizing a global loss with first-order gradient descent), it yields rather imprecise estimates and does not scale well. Its recent extension MASt3R [26] adds pixel matching capabilities and improved pointmap regression, but does not address the SfM problem. In this work, we fill this gap and present a fully-integrated SfM pipeline based on MASt3R that is both precise and scalable.

**Image Retrieval for SfM.** Since matching is essentially considering pairs in traditional SfM, it has a quadratic complexity which becomes prohibitive for large image collections. Several SfM approaches have proposed to leverage faster, although less precise, image comparison techniques relying on comparing global image descriptors, *e.g.* APGeM [40] for Kapture [20] or by distilling NetVLAD [3] for HLoc [44]. The idea is to cascade image matching in two steps: first, a coarse but fast comparison is carried out between all pairs (usually by computing the similarity between global image descriptors), and for image pairs that are similar enough, a second stage of costly keypoint matching is then carried out. This is arguably much faster and scalable. In this paper, we adopt the same strategy, but instead of relying on an external off-the-shelf module, we show that we can simply exploit the frozen MASt3R's encoder for this purpose, considering the token features as local features and directly performing efficient retrieval with Aggregated Selective Match Kernels (ASMK) [56].

## 3. Preliminaries

The proposed method builds on the recently introduced MASt3R model which, given two input images $I^n, I^m \in \mathbb{R}^{H\times W\times 3}$, performs joint *local 3D reconstruction* and *pixel-wise matching* [26]. We assume here for simplicity that all images have the same pixel resolution $H \times W$, but of course they can differ in practice. In the next section, we show how to leverage this powerful *local* predictor for achieving large-scale *global* 3D reconstruction.

At a high level, MASt3R can be viewed as a function $f(I^n, I^m) \equiv \text{Dec}(\text{Enc}(I^n), \text{Enc}(I^m))$, where $\text{Enc}(I) \to F$ denotes the Siamese ViT encoder that represents image $I$ as a feature map of dimension $d$, width $w$ and height $h$, $F \in \mathbb{R}^{h\times w\times d}$, and $\text{Dec}(F^n, F^m)$ denotes twin ViT decoders that regresses pixel-wise pointmaps $X$ and local features $D$ for each image, as well as their respective corresponding confidence maps. These outputs intrinsically contain rich geometric information from the scene, to the extent that camera intrinsics and (metric) depthmaps can straightforwardly be recovered from the pointmap, see [64] for details. Likewise, we can recover sparse correspondences (or *matches*) by application of the fastNN algorithm described in [26] with the regressed local feature maps $D^n, D^m$. More specifically, the fast NN searches for a subset of reciprocal correspondences from two feature maps $D^n$ and $D^m$ by initializing seeds on a regular pixel grid and iteratively converging to mutual correspondences. We denote these correspondences between $I^n$ and $I^m$ as $\mathcal{M}^{n,m} = \{y^n_c \leftrightarrow y^m_c\}_{c=1..|\mathcal{M}^{n,m}|}$, where $y^n_c, y^m_c \in \mathbb{N}^2$ denotes a pair of matching pixels.

## 4. Proposed Method

Given an unordered collection of $N$ images $\mathcal{V} = \{I^n\}_{1\le n\le N}$ of a static 3D scene, captured with respective cameras $\mathcal{K}_n = (K_n, P_n)$, where $K_n \in \mathbb{R}^{3\times 3}$ denotes the intrinsic parameters (*i.e.* calibration in term of focal length and principal point) and $P_n \in \mathbb{R}^{4\times 4}$ its world-to-camera pose, our goal is to recover all cameras parameters $\{\mathcal{K}_n\}$ as well as the underlying 3D scene geometry $\{X^n\}$, with $X^n \in \mathbb{R}^{H\times W\times 3}$ a pointmap relating each pixel $y = (i, j) \in \mathbb{N}^2$ from $I^n$ to its corresponding 3D point $X^n_{i,j}$ in the scene expressed in a world coordinate system.

**Overview.** We present a novel large-scale 3D reconstruction approach consisting of four steps outlined in Fig. 2. First, we construct a co-visibility graph using efficient and scalable image retrieval techniques. Edges of this graph connect pairs of likely-overlapping images. Second, we perform pairwise local 3D reconstruction and matching using MASt3R for each edge. Third, we coarsely align every local pointmap in the same world coordinate system using gradient descent with a matching loss in 3D space. This serves as initialization for the fourth step, wherein we perform a second stage of global optimization, this time minimizing 2D pixel reprojection errors. We detail each step below.

### 4.1. Scene graph

We first aim at spatially relating scene objects seen under different viewpoints. MASt3R is originally a pairwise image matcher, which has quadratic complexity in the number $N$ of images and therefore becomes infeasible for large collections if done naively.

**Sparse scene graph.** Instead, we wish to only feed a small but sufficient subset of all possible pairs to MASt3R, which structure forms a scene graph $\mathcal{G}$. Formally, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph where each vertex $I \in \mathcal{V}$ is an image, and each edge $e = (n, m) \in \mathcal{E}$ is an undirected connection between two likely-overlapping images $I^n$ and $I^m$. Importantly, $\mathcal{G}$ must have a single connected component, *i.e.* all images must (perhaps indirectly) be linked together.

**Image retrieval.** To select the right subset of pairs, we rely on a scalable pairwise image matcher $h(I^n, I^m) \mapsto s$, able to predict the approximate co-visibility score $s \in [0, 1]$ between two images $I^n$ and $I^m$. While any off-the-shelf image retriever can in theory do, we propose to leverage MASt3R's encoder $\text{Enc}(\cdot)$. Indeed, our findings are that the encoder, due to its role of laying foundations for the decoder, is implicitly trained for image matching (see Sec. 5.3). To that aim, we adopt the ASMK (Aggregated Selective Match Kernels) image retrieval method [56] considering the token features output by the encoder as local features. ASMK has shown excellent performance for retrieval, especially without requiring any spatial verification. In a nutshell, we consider the output $F$ of the encoder as a bag of local features, apply feature whitening, quantize them according to a codebook previously obtained by k-means clustering, then aggregate and binarize the residuals for each codebook element, thus yielding high-dimensional sparse binary representations. The ASMK similarity between two image representations can be efficiently computed by summing a small kernel function on binary representations over the common codebook elements. Note that this method is training-free, only requiring to compute the whitening matrix and the codebook once from a representative set of features. We have also tried learning a small projector on top of the encoder features following the HOW approach [57], but this leads to similar performances. We refer to the supplementary for more details. The output from the retrieval step is a similarity matrix $S \in [0, 1]^{N \times N}$.

**Graph construction.** To get a small number of pairs while still ensuring a single connected component, we build the graph $\mathcal{G}$ as follows. We first select a fixed number $N_a$ of *key images* (or keyframes) using farthest point sampling (FPS) [15] based on $S$. These keyframes constitute the core set of nodes and are densely connected together. All remaining images are then connected to their closest keyframe as well as their $k$ nearest neighbors according to $S$. Such a graph comprises $O(N_a^2 + (k+1)N) = O(N) \ll O(N^2)$ edges, which is linear in the number of images $N$. We typically use $N_a = 20$ and $k = 10$. Note that, while the retrieval step has quadratic complexity in theory, it is extremely fast and scalable in practice, so we ignore it and report quasi-linear complexity overall.

### 4.2. Local reconstruction

We run the inference of MASt3R for every pair $e = (n, m) \in \mathcal{E}$, yielding raw pointmaps and sparse pixel matches $\mathcal{M}^{n,m}$. Since MASt3R is order-dependent in terms of its input, we define $\mathcal{M}^{n,m}$ as the union of correspondences obtained by running both $f(I^n, I^m)$ and $f(I^m, I^n)$. Doing so, we also obtain pointmaps $X^{n,n}, X^{n,m}, X^{m,n}$ and $X^{m,m}$, where $X^{n,m} \in \mathbb{R}^{H \times W \times 3}$ denotes a 2D-to-3D mapping from pixels of image $I^n$ to 3D points in the coordinate system of image $I^m$. Since the encoder features $\{F^n\}_{n=1..N}$ have already been extracted and cached during scene graph construction (Sec. 4.1), we only need to run the ViT decoder $\text{Dec}()$, which substantially saves compute.

**Canonical pointmaps.** We wish to estimate an initial depthmap $Z^n$ and camera intrinsics $K_n$ for each image $I^n$. These can be easily recovered from a raw pointmap $X^{n,n}$ [64], but each pair $(n, \cdot)$ or $(\cdot, n) \in \mathcal{E}$ would yield its own estimate of $X^{n,n}$. To average out regression imprecision, we hence aggregate these copycat pointmaps into a canonical pointmap $\tilde{X}^n$. Let $\mathcal{E}^n = \{e | e \in \mathcal{E} \wedge n \in e\}$ be the set of all edges connected to image $I^n$. For each edge $e \in \mathcal{E}^n$, we have a different estimate of $X^{n,n}$ and its respective confidence maps $C^{n,n}$, which we will denote as $X^{n,e}$ and $C^{n,e}$. We compute the canonical pointmap as a simple per-pixel weighted average of all estimates:

$$\tilde{X}^n_{i,j} = \frac{\sum_{e \in \mathcal{E}^n} C^{n,e}_{i,j} X^{n,e}_{i,j}}{\sum_{e \in \mathcal{E}^n} C^{n,e}_{i,j}}. \tag{1}$$

We then recover the canonical depthmap $\tilde{Z}^n = \tilde{X}^n_{:,:,3}$ and the focal length using the Weiszfeld algorithm [64]:

$$f^* = \arg\min_f \sum_{i,j} \left\| \left(i - \frac{W}{2}, j - \frac{H}{2}\right) - f\left(\frac{\tilde{X}^n_{i,j,1}}{\tilde{X}^n_{i,j,3}}, \frac{\tilde{X}^n_{i,j,2}}{\tilde{X}^n_{i,j,3}}\right) \right\|, \tag{2}$$

which, assuming centered principal point and square pixels, yields the canonical intrinsics $\tilde{K}^n$. In this work, we assume

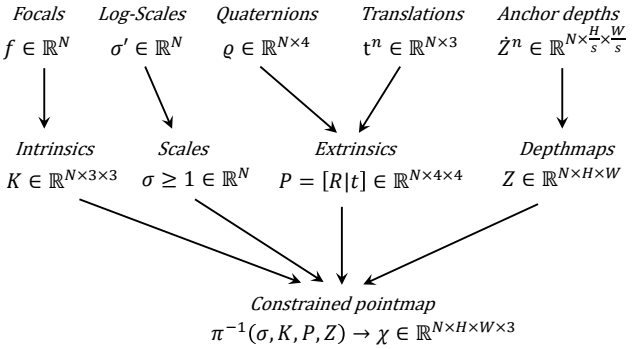


Figure 3. **Factor graph for MASt3R-SfM**. Free variables on the top row serve to construct the constrained pointmap $\chi$, which follows the pinhole camera model by design and onto which the loss functions from Eqs. (3) and (4) are defined.

a pinhole camera model without lens distortion, but our approach could be extended to different camera types.

**Constrained pointmaps.** Camera intrinsics $K$, extrinsics $P$ and depthmaps $Z$ will serve as basic ingredients (or rather, optimization variables) for the global reconstruction phase. Let $\pi_n : \mathbb{R}^3 \mapsto \mathbb{R}^2$ denote the reprojection function onto the camera screen of $I^n$, *i.e.* $\pi_n(x) = K_n P_n \sigma_n x$ for a 3D point $x \in \mathbb{R}^3$ ($\sigma_n > 0$ is a per-camera scale factor, *i.e.* we use scaled rigid transformations). To ensure that pointmaps perfectly satisfy the pinhole projective model (they are normally over-parameterized), we define a *constrained pointmap* $\chi^n \in \mathbb{R}^{H\times W\times 3}$ explicitly as a function of $K_n, P_n, \sigma_n$ and $Z^n$. Formally, the 3D point $\chi^n_{i,j}$ seen at pixel $(i,j)$ of image $I^n$ is defined using inverse reprojection as $\chi^n_{i,j} = \pi_n^{-1}(\sigma_n, K_n, P_n, Z^n_{i,j}) = {}^1\!/_{\sigma_n} P_n^{-1} K_n^{-1} Z^n_{i,j} \, [i, j, 1]^\top$.

### 4.3. Coarse alignment

Recently, DUSt3R [64] introduced a global alignment procedure aiming to rigidly move dense pointmaps in a world coordinate system based on pairwise relationships between them. In this work, we simplify and improve this procedure by taking advantage of pixel correspondences, thereby reducing the overall number of parameters and its memory and computational footprint.

Specifically, we look for the scaled rigid transformations $\sigma^*, P^*$ of every canonical pointmaps $\chi = \pi^{-1}(\sigma, \tilde{K}, P, \tilde{Z})$ (*i.e.* fixing intrinsics $K = \tilde{K}$ and depth $Z = \tilde{Z}$ to their canonical values) such that any pair of matching 3D points gets as close as possible:

$$\sigma^*, P^* = \arg\min_{\sigma, P} \sum_{\substack{c \in \mathcal{M}^{n,m} \\ (n,m)\in\mathcal{E}}} q_c \left\| \chi^n_c - \chi^m_c \right\|^{\lambda_1}, \qquad (3)$$

where $c$ denotes the matching pixels in each respective image by a slight abuse of notation. In contrast to the global alignment procedure in DUSt3R, this minimization only applies to sparse pixel correspondences $y^n_c \leftrightarrow y^m_c$ weighted by their respective confidence $q_c = \sqrt{Q^{n,e}_c Q^{m,e}_c}$ (with descriptor confidence maps $Q^{\cdot,e} \in \mathbb{R}^{H\times W}$ also output by MASt3R). To avoid degenerate solutions, we enforce $\min_n \sigma_n = 1$ by reparameterizing $\sigma_n = \sigma'_n / (min_n \sigma'_n)$.

**Optimization.** We minimize this objective using first-order optimization for simplicity, using Adam [23] for a fixed number $\nu_1$ of iterations. Alternatively, a second-order LM or Gauss-Newton optimization schemes could result in faster convergence [36], which we leave for future work.

### 4.4. Refinement

Coarse alignment converges well and fast in practice, but restricts itself to rigid motion of canonical pointmaps. Unfortunately, pointmaps are bound to be noisy due to depth ambiguities during local reconstruction. To further refine cameras and scene geometry, we thus perform a second round of global optimization akin to bundle adjustment [59] with gradient descent for $\nu_2$ iterations and starting from the coarse solution $\sigma^*, P^*$ obtained from Eq. (3). In other words, we minimize the 2D reprojection error of 3D points in all cameras:

$$Z^*, K^*, P^*, \sigma^* = \arg\min_{Z,K,P,\sigma} \mathcal{L}_2, \text{ with} \qquad (4)$$

$$\mathcal{L}_2 = \sum_{\substack{c \in \mathcal{M}^{n,m} \\ (n,m)\in\mathcal{E}}} q_c \left[ \rho\left(y^n_c - \pi_n\left(\chi^n_c\right)\right) + \rho\left(y^m_c - \pi_m\left(\chi^n_c\right)\right) \right],$$

with $\rho : \mathbb{R}^2 \mapsto \mathbb{R}^+$ a robust error function able to deal with potential outliers among all extracted correspondences. We typically set $\rho(x) = \|x\|^{\lambda_2}$ with $0 < \lambda_2 \leq 1$ (*e.g.* $\lambda_2$=0.5).

**Forming pseudo-tracks.** Optimizing Eq. (4) has little effect, because sparse pixel correspondences $\mathcal{M}^{n,m}$ are rarely *exactly* overlapping across several pairs. As an illustration, two correspondences $y^m_{\cdot,\cdot} \leftrightarrow y^n_{i,j}$ and $y^n_{i+1,j} \leftrightarrow y^l_{\cdot,\cdot}$ from image pairs $(m,n)$ and $(n,l)$ would independently optimize the two 3D points $\chi^n_{i,j}$ and $\chi^n_{i+1,j}$, possibly moving them very far apart despite this being very unlikely as $(i,j) \simeq (i+1,j)$. Traditional SfM methods resort to forming point tracks, which is relatively straightforward with keypoint-based matching [12, 29, 31, 43, 46]. We propose instead to form pseudo-tracks by defining *anchor points* and rigidly tying together every pixel with their closest anchor point. This way, correspondences that do not overlap exactly are still both tied to the same anchor point with a high probability. Formally, we define anchor points with a regular pixel grid $\dot{y} \in \mathbb{R}^{H/s\times W/s\times 2}$ spaced by $\delta$ pixels:

$$\dot{y}_{u,v} = \left( u\delta + \frac{\delta}{2}, v\delta + \frac{\delta}{2} \right). \qquad (5)$$

We then tie each pixel $(i,j)$ in $I^n$ with its closest anchor $\dot{y}_{u,v}$ at coordinate $(u,v) = (\lfloor i/\delta \rfloor, \lfloor j/\delta \rfloor)$. Concretely, we simply index the depth value at pixel $(i,j)$ to the depth value $\dot{Z}_{u,v}$ of its anchor point, *i.e.* we define $Z_{i,j} = o_{i,j} \dot{Z}_{u,v}$

| Method | 25 views | | 50 views | | 100 views | | 200 views | | full | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ATE↓ | Reg.↑ | ATE↓ | Reg.↑ | ATE↓ | Reg.↑ | ATE↓ | Reg.↑ | ATE↓ | Reg.↑ |
| COLMAP [46] | 0.03840 | 44.4 | 0.02920 | 60.5 | 0.02640 | 85.7 | 0.01880 | 97.0 | - | - |
| ACE-Zero [9] | 0.11160 | 100.0 | 0.07130 | 100.0 | 0.03980 | 100.0 | 0.01870 | 100.0 | 0.01520 | 100.0 |
| FlowMap [50] | 0.10700 | 100.0 | 0.07310 | 100.0 | 0.04460 | 100.0 | 0.02420 | 100.0 | N/A | 66.7 |
| VGGSfM [62] | 0.05800 | 96.2 | 0.03460 | 98.7 | 0.02900 | 98.5 | N/A | 47.6 | N/A | 0.0 |
| DF-SfM [19] | 0.08110 | 99.4 | 0.04120 | 100.0 | 0.02710 | 99.9 | N/A | 33.3 | N/A | 76.2 |
| **MASt3R-SfM** | 0.03360 | 100.0 | 0.02610 | 100.0 | 0.01680 | 100.0 | 0.01300 | 100.0 | 0.01060 | 100.0 |

| Method | MIP-360 | LLFF | T&T | CO3Dv2 |
|---|---|---|---|---|
| NoPE-NeRF [8] | 0.04429 | 0.03920 | 0.03709 | 0.03648 |
| DROID-SLAM [54] | 0.00017 | 0.00074 | 0.00122 | 0.01728 |
| FlowMap [50] | 0.00055 | 0.00209 | 0.00124 | 0.01589 |
| ACE-Zero [9] | 0.00173 | 0.00396 | 0.00973 | 0.00520 |
| **MASt3R-SfM** | 0.00079 | 0.00098 | 0.00215 | 0.00538 |

Table 1. **Results on Tanks&Temples** in terms of ATE and overall registration rate (Reg.). For easier readability, we color-code ATE results as a linear gradient between worst and best ATE for a given dataset or split; and Reg results with linear gradient between 0% and 100%. **Left**: Impact of the number of input views, regularly sampled from the full set. 'N/A' indicates that at least one scene did not converge. **Right**: ATE↓ on different datasets with the arbitrary splits defined in FlowMap [50].

where $o_{i,j} = \tilde{Z}_{i,j}/\tilde{Z}_{u,v}$ is a constant relative depth offset calculated at initialization from the canonical depthmap $\tilde{Z}$. Here, we make the assumption that canonical depthmaps are locally accurate. All in all, optimizing a depthmap $Z^n \in \mathbb{R}^{H\times W}$ thus only comes down to optimizing a reduced set of anchor depth values $\dot{Z}^n \in \mathbb{R}^{H/\delta\times W/\delta}$ (*e.g.* reduced by a factor of 64 if $\delta = 8$).

## 5. Experimental Results

After presenting the datasets and metrics, we extensively compare our approach with state-of-the-art SfM methods in diverse conditions. We finally present several ablations.

### 5.1. Experimental setup

We use the publicly available MASt3R checkpoint for our experiments, which we do *not* finetune unless otherwise mentioned. When building the sparse scene graph in Sec. 4.1, we use $N_a = 20$ anchor images and $k = 10$ non-anchor nearest neighbors. We use the same grid spacing of $\delta = 8$ pixels for extracting sparse correspondences with fastNN (Sec. 4.2) and defining anchor points (Sec. 4.4). For the two gradient descents, we use the Adam optimizer [23] with a learning rate of 0.07 (resp. 0.014) for $\nu_1 = 300$ iterations and $\lambda_1 = 1.5$ (resp. $\nu_2 = 300$ and $\lambda_2 = 0.5$) for the coarse (resp. refinement) optimization, each time with a cosine learning rate schedule and without weight decay. Unless otherwise mentioned, we assume shared intrinsics and optimize a shared per-scene focal parameter for all cameras.

**Datasets.** To showcase the robustness of our approach, we experiment in different conditions representative of diverse experimental setups (video or unordered image collections, simple or complex scenes, outdoor, indoor or object-centric, *etc.*). Namely, we employ Tanks&Temples [24] (T&T), a 3D reconstruction dataset comprising 21 scenes ranging from 151 to 1106 images; ETH3D [49], a multi-view stereo dataset with 13 scenes for which ground-truth is available; CO3Dv2 [39], an object-centric dataset for multi-view pose estimation; and RealEstate10k [70], MIP-360 [6] and LLFF [33], three datasets for novel view synthesis. We note that 14 scenes of T&T are part of MegaDepth [27], which is used for training the MASt3R checkpoint we used.

| | Method | Co3Dv2↑ | | | RealEstate10K↑ |
|---|---|---|---|---|---|
| | | RRA@15 | RTA@15 | mAA(30) | mAA(30) |
| (a) | Colmap+SG [12, 43] | 36.1 | 27.3 | 25.3 | 45.2 |
| | PixSfM [29] | 33.7 | 32.9 | 30.1 | 49.4 |
| | RelPose [68] | 57.1 | - | - | - |
| | PosReg [61] | 53.2 | 49.1 | 45.0 | - |
| | PoseDiff [61] | 80.5 | 79.8 | 66.5 | 48.0 |
| | RelPose++ [28] | (85.5) | - | - | - |
| | RayDiff [69] | (93.3) | - | - | - |
| | DUSt3R-GA [64] | **96.2** | 86.8 | 76.7 | 67.7 |
| | **MASt3R-SfM** | 96.0 | **93.1** | **88.0** | **86.8** |
| (b) | DUSt3R [64] | 94.3 | 88.4 | 77.2 | 61.2 |
| | MASt3R [26] | 94.6 | 91.9 | 81.8 | 76.4 |

Table 2. **Multi-view pose regression on CO3Dv2 [39] and RealEstate10K [70] with 10 random frames**. Parenthesis () denote methods that do not report results on the 10 views set, we report their best for comparison (8 views). We distinguish between (a) multi-view and (b) pairwise methods.

As shown in the detailed results provided in Section 4 of the supplementary material, we do *not* observe any significant differences between seen and unseen scenes in terms of accuracies and comparison with the state of the art.

**Evaluation metrics.** We evaluate all methods w.r.t. ground-truth cameras poses. For Tanks&Temples where it is not provided, we make a pseudo ground-truth with COLMAP [46] using all frames. Even though this is not perfect, COLMAP is known to be reliable in conditions where there is a large number of frames with high overlap. We evaluate the average translation error (ATE) as in FlowMap [50], *i.e.* we align estimated camera positions to ground-truth ones with Procrustes [32] and report an average normalized error. We ignore unregistered cameras when doing Procrustes, which favors methods that can reject hard images (such as COLMAP [46] or VGGSfM [62]). Note that our method always outputs a pose estimate for all cameras by design, thus negatively impacting our results with this metric. We also report the relative rotation and translation accuracies (resp. RTA@$\tau$ and RRA@$\tau$, where $\tau$ indicates the threshold in degrees), computed at the pairwise level and averaged over all image pairs [61]. Similarly, the mean Average Accuracy (mAA)@$\tau$ is defined as the area under the curve of the angular differences at min(RRA@$\tau$, RTA@$\tau$). Finally, we report the successful registration rate as a percentage, denoted as Reg. When reported at the dataset level, metrics are averaged over all scenes.

| Scenes | COLMAP [46] | | ACE-Zero [9] | | FlowMap [50] | | VGGSfM [62] | | DF-SfM [19] | | **MASt3R-SfM** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RRA@5 | RTA@5 | RRA@5 | RTA@5 | RRA@5 | RTA@5 | RRA@5 | RTA@5 | RRA@5 | RTA@5 | RRA@5 | RTA@5 |
| courtyard | 56.3 | 60.0 | 4.0 | 1.9 | 7.5 | 3.6 | 50.5 | 51.2 | 80.7 | 74.8 | 89.8 | 64.4 |
| delivery area | 34.0 | 28.1 | 27.4 | 1.9 | 29.4 | 23.8 | 22.0 | 19.6 | 82.5 | 82.0 | 83.1 | 81.8 |
| electro | 53.3 | 48.5 | 16.9 | 7.9 | 2.5 | 1.2 | 79.9 | 58.6 | 82.8 | 81.2 | 100.0 | 95.5 |
| facade | 92.2 | 90.0 | 74.5 | 64.1 | 15.7 | 16.8 | 57.5 | 48.7 | 80.9 | 82.6 | 74.3 | 75.3 |
| kicker | 87.3 | 86.2 | 26.2 | 16.8 | 1.5 | 1.5 | 100.0 | 97.8 | 93.5 | 91.0 | 100.0 | 100.0 |
| meadow | 0.9 | 0.9 | 3.8 | 0.9 | 3.8 | 2.9 | 100.0 | 96.2 | 56.2 | 58.1 | 58.1 | 58.1 |
| office | 36.9 | 32.3 | 0.9 | 0.0 | 0.9 | 1.5 | 64.9 | 42.1 | 71.1 | 54.5 | 100.0 | 98.5 |
| pipes | 30.8 | 28.6 | 9.9 | 1.1 | 6.6 | 12.1 | 100.0 | 97.8 | 72.5 | 61.5 | 100.0 | 100.0 |
| playground | 17.2 | 18.1 | 3.8 | 2.6 | 2.6 | 2.8 | 37.3 | 40.8 | 70.5 | 70.1 | 100.0 | 93.6 |
| relief | 16.8 | 16.8 | 16.8 | 17.0 | 6.9 | 7.7 | 59.6 | 57.9 | 32.9 | 32.9 | 34.2 | 40.2 |
| relief 2 | 11.8 | 11.8 | 7.3 | 5.6 | 8.4 | 2.8 | 69.9 | 70.3 | 40.9 | 39.1 | 57.4 | 76.1 |
| terrace | 100.0 | 100.0 | 5.5 | 2.0 | 33.2 | 24.1 | 38.7 | 29.6 | 100.0 | 99.6 | 100.0 | 100.0 |
| terrains | 100.0 | 99.5 | 15.8 | 4.5 | 12.3 | 13.8 | 70.4 | 54.9 | 100.0 | 91.9 | 58.2 | 52.5 |
| Average | 49.0 | 47.8 | 16.4 | 9.7 | 10.1 | 8.8 | 65.4 | 58.9 | 74.2 | 70.7 | 81.2 | 79.7 |

Table 3. **Detailed per-scene translation and rotation accuracies (↑) on ETH-3D.** For clarity, we color-code results with a linear gradient between the worst and best result for a given scene.

## 5.2. Comparison with the state of the art

We first evaluate the impact of the amount of overlap between images on the quality of the SfM output for different state-of-the-art methods. To that aim, we choose Tanks&Temple, a standard reconstruction dataset captured with high overlap (originally video frames). We form new splits by regularly subsampling the original images for 25, 50, 100 and 200 frames. Following [50], we report results in terms of Average Translation Error (ATE) against the COLMAP pseudo ground-truth in Tab. 1 (left), computed from the full set of frames and likewise further subsampled. MASt3R-SfM provides nearly constant performance for all ranges, significantly outperforming COLMAP, Ace-Zero, FlowMap and VGGSfM in all settings. Unsurprisingly, the performance of these methods strongly degrades in small-scale settings (or does not even converge on some scenes for COLMAP). On the other hand, we note that FlowMap and VGGSfM crash when dealing with large collections due to insufficient memory despite using 80GB GPUs.

**FlowMap splits.** We also report results on the custom splits from the FlowMap paper [50], which concerns 3 additional datasets beyond T&T (LLFF, Mip-360 and CO3Dv2). We point out that, not only these splits select a *subset* of scenes for each dataset (in details: 3 scenes from Mip-360, 7 from LLFF, 14 from T&T and 2 from CO3Dv2), they also select an *arbitrary subset* of consecutive frames in the corresponding scenes. Results in Tab. 1 (right) show that our method achieves better results than NopeNeRF and ACE-Zero, on par with FlowMap overall and slightly worse than DROID-SLAM [54], a method that only works in video settings. Since we largely outperform FlowMap when using regularly sampled splits, we hypothesize that FlowMap is very sensitive to the input setting.

**Multi-view pose estimation.** In Fig. 1 (top), we evaluate on CO3Dv2 and RealEstate10K, varying the number of input images by random sampling. We follow the PoseDiffusion [61] splits and protocol. We provide detailed comparisons in Tab. 2 with state-of-the-art multi-view pose estimation methods, whose goal is only to recover cameras poses but not the scene geometry. Our approach compares favorably to existing methods, particularly when the number of input images is low. Overall, this highlights that MASt3R-SfM is extremely robust to sparse view setups, with its performance not degrading when decreasing the number of views, even for as little as three views.

**Unordered collections.** We note that benchmarks in previous experiments were originally acquired as videos later subsampled into frames. This might introduce biases that may not well represent the general case of unconstrained SfM. We thus experiment on the ETH3D dataset, a photograph dataset, composed of 13 scenes with up to to 76 images per scene. Results reported in Tab. 3 shows that MASt3R-SfM outperforms all competing approaches by a large margin on average. This is not surprising, as neither ACE-Zero nor FlowMap can handle non-video setups. The fact that COLMAP and VGGSfM also perform relatively poorly indicates a high sensitivity to not having highly overlapping images, meaning that in the end these methods cannot really handle truly unconstrained collections, in spite of some opposite claims [62].

## 5.3. Ablations

We now study the impact of various design choices. All experiments are conducted on the Tanks&Temples dataset regularly subsampled for 200 views per scene.

**Scene graph.** We evaluate different construction strategies for the scene graph in Tab. 4: 'complete' means that we extract all pairs, 'local window' is an heuristic for video-based collections that connects every frame with its neighboring frames, and 'random' means that we sample random pairs. Except for the 'complete' case, we try to match the number of pairs used in the baseline retrieval strategy. Slightly better results are achieved with the complete graph, but it is about 10x slower than retrieval-based graph and not

| Scene Graph | ATE↓ | RTA@5↑ | RRA@5↑ | #Pairs | GPU MEM | Avg. T |
|---|---|---|---|---|---|---|
| Complete | 0.01256 | 75.9 | 74.8 | 39,800 | 29.9 GB | 2.2 h |
| Local window | 0.02509 | 33.1 | 28.8 | 2,744 | 7.6 GB | 14.1 min |
| Random | 0.01558 | 55.2 | 48.8 | 2,754 | 6.9 GB | 14.7 min |
| **Retrieval** | 0.01243 | 70.9 | 67.6 | 2,758 | 8.4 GB | 14.3 min |

Table 4. **Ablation of scene graph construction** on Tanks&Temples (200 view subset).

| Ablation | | ATE↓ | RTA@5↑ | RRA@5↑ | #Pairs |
|---|---|---|---|---|---|
| Retrieval | kNN | 0.01440 | 64.1 | 61.9 | 3,042 |
| | Keyframes | 0.01722 | 58.1 | 57.1 | 740 |
| | **Keyframes + kNN** | **0.01243** | **70.9** | **67.6** | 2,758 |
| Optimization level | Coarse | 0.01504 | 47.4 | 57.7 | 2,758 |
| | Fine (w/o depth) | 0.01315 | 67.3 | 66.9 | 2758 |
| | **Fine** | **0.01243** | **70.9** | **67.6** | 2,758 |
| Intrinsics | Separate | 0.01329 | 66.9 | 64.2 | 2,758 |
| | **Shared** | **0.01243** | **70.9** | **67.6** | 2,758 |

Table 5. **Other ablations on Tanks&Temples** (200 view subset).

scalable in general. Assuming we use retrieval, we further ablate the scene graph building strategy from the similarity matrix in Tab. 5. As a reminder, it consists of building a small but complete graph of keyframes, and then connecting each image with the closest keyframe and with $k$ nearest non-keyframes. We experiment with using only k-NN with an increased $k = 13$ to compensate for the missing edges, denoted as 'k-NN', or to only use the keyframe graph (*i.e.* $k = 0$), denoted as 'Keyframe'. Overall, we find that combining short-range ($k$-NN) and long-range (keyframes) connections is important for reaching top performance.

**Retrieval with MASt3R.** To better assess the effectiveness of our image retrieval strategy alone, we conduct experiments for the task of retrieval-assisted visual localization. We follow the protocol from [26] and retrieve the top-$k$ posed images in the database for each query, extract 2D-3D correspondences and run RANSAC to predict camera poses. We compare ASMK on MASt3R features to the off-the-shelf FIRe retrieval method [66], also based on ASMK, on Aachen-Day-Night [45] and InLoc [53]. We report standard visual localization accuracy metrics, *i.e.* the percentages of images successfully localized within (0.25m,2°) / (0.5m,5°) / (5m,10°) and (0.25m,2°) / (0.5m,10°) / (1m,10°) respectively.[1] in Tab. 6. Interestingly, using frozen MASt3R features for retrieval performs on par with FIRe, a state-of-the-art method specifically trained for image retrieval and operating on multi-scale features (bottom row). Our method is also competitive with dedicated visual localization pipelines (top rows), even setting a new state of the art for InLoc. We refer to the supplementary material for further comparisons.

**Optimization level.** We study the impact of the coarse optimization and refinement (Tab. 5). Coarse optimization alone, which is somewhat comparable to the global alignment of DUSt3R (except we are using sparse matches and less optimization variables), yields significantly less precise pose estimates. Fig. 4 shows the pose accuracy as a function of the number of iterations during coarse optimization and refinement. As expected, refinement, a strongly non-convex bundle-adjustment problem, cannot recover from a random initialization ($\nu_1 = 0$). Good enough poses are typically obtained after $\nu_1 \simeq 250$ iterations of coarse optimization, from which point refinement consistently improves. We also try to perform the optimization without optimizing depth (*i.e.* using frozen canonical depthmaps, which proves useful for purely rotational cases, denoted as 'Fine without depth' in Tab. 5), in which case we observe a smaller impact on the performance, indicating the high-quality of canonical depthmaps output by MASt3R (Sec. 4.2).

[1] https://www.visuallocalization.net/

| Method | Aachen-Day-Night↑ | | InLoc↑ | |
|---|---|---|---|---|
| | Day | Night | DUC1 | DUC2 |
| Kapture [20]+R2D2 [41] | **91.3/97.0**/99.5 | **78.5**/91.6/**100** | 41.4/60.1/73.7 | 47.3/67.2/73.3 |
| SuperPoint [12]+LightGlue [30] | 90.2/96.0/99.4 | 77.0/91.1/**100** | 49.0/68.2/79.3 | 55.0/74.8/79.4 |
| LoFTR [52] | 88.7/95.6/99.0 | **78.5**/90.6/99.0 | 47.5/72.2/84.8 | 54.2/74.8/85.5 |
| DKM [13] | - | - | 51.5/75.3/86.9 | 63.4/82.4/87.8 |
| MASt3R (FIRe top20) | 89.8/96.8/**99.6** | 75.9/**92.7/100** | **60.6/83.3**/93.4 | 65.6/86.3/88.5 |
| **MASt3R (MASt3R-ASMK top20)** | 88.7/94.9/98.2 | 77.5/90.6/97.9 | 58.1/82.8/**94.4** | **69.5/90.8/92.4** |

Table 6. **Comparison of retrieval based on MASt3R features** using ASMK with the state-of-the-art FIRe method when localizing with MASt3R (bottom rows), as well as with other state-of-

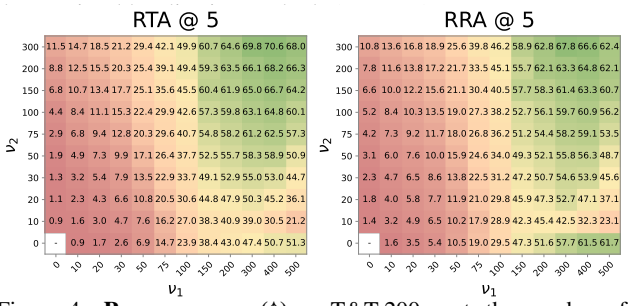


Figure 4. **Pose accuracy** (↑) on T&T-200 w.r.t. the number of iterations of the coarse and refinement stages (resp. $\nu_1$ and $\nu_2$).

**Shared intrinsics.** We also report the impact of only optimizing one set of intrinsics for all views ('shared'), which is small: our method is not sensitive to varying intrinsics.

**Limitations.** Convergence for large scenes can be slow as we use a first-order optimization only. Moreover, our method can be sensible to a poor initialization of the sparse scene graph, which can occur *e.g.* with symmetric structures that looks similar but are not connected, see Section 3 and Figure 2 of the supplementary material. Robust methods like RANSAC would also fail in such cases.

## 6. Conclusion

We have introduced MASt3R-SfM, a fully-integrated solution for unconstrained SfM. In contrast with existing pipelines, it can handle very small image collections. Thanks to the strong priors encoded in the underlying MASt3R foundation model upon which our approach is built, it can even deal with cases without motion, and does not rely at all on RANSAC, both features that are normally not possible with standard triangulation-based SfM.

## References


[1] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *ECCV*, 2010. 2

[2] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 2023. 2

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 3

[4] Daniel Barath and Jiří Matas. Graph-cut ransac. In *CVPR*, 2018. 1

[5] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *CVPR*, 2020. 1

[6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 1, 6

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 2

[8] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, 2023. 6

[9] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene Coordinate Reconstruction: Posing of Image Collections via Incremental Learning of a Relocalizer. In *ECCV*, 2024. 2, 6, 7

[10] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robotics*, 2021. 1

[11] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, 2022. 2

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised Interest Point Detection and Description. In *CVPR*, 2018. 5, 6, 8

[13] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *CVPR*, 2023. 8

[14] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *CVPR*, 2024. 2

[15] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. In *ICPR*, 1994. 4

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. 1

[17] Annika Hagemann, Moritz Knorr, and Christoph Stiller. Deep geometry-aware camera self-calibration from video. In *ICCV*, 2023. 1

[18] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 1

[19] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *CVPR*, 2024. 2, 3, 6, 7

[20] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv preprint arXiv:2007.13867*, 2020. 3, 8

[21] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *ICCV*, 2021. 2

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graphics*, 2023. 1

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5, 6

[24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graphics*, 2017. 6

[25] Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized ransac. In *BMVC*, 2012. 1

[26] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024. 2, 3, 6, 8

[27] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6

[28] Amy Lin, Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. In *3DV*, 2024. 6

[29] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *ICCV*, 2021. 2, 5, 6

[30] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *ICCV*, 2023. 8

[31] G Lowe. Sift-the scale invariant feature transform. *IJCV*, 2004. 2, 5

[32] Bin Luo and Edwin R Hancock. Procrustes alignment with the em algorithm. In *ICCAIP*, 1999. 6

[33] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graphics*, 2019. 6

[34] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 2021. 1

[35] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Trans. Robotics*, 2015. 1

[36] Riku Murai, Eric Dexheimer, and Andrew J Davison. MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors. *arXiv preprint arXiv:2412.12392*, 2024. 5

[37] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *CVPR*, 2022. 1

[38] MV Peppa, JP Mills, KD Fieber, I Haynes, S Turner, A Turner, M Douglas, and PG Bryan. Archaeological feature detection from archive aerial photography with a sfm-mvs and image enhancement pipeline. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018. 1

[39] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotný. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021. 6

[40] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 3

[41] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, 2019. 2, 8

[42] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 2

[43] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 5, 6

[44] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 3

[45] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018. 8

[46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 5, 6, 7

[47] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1

[48] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 1

[49] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 6

[50] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. FlowMap: High-Quality Camera Poses, Intrinsics, and Depth via Gradient Descent. In *ECCV*, 2024. 2, 6, 7

[51] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, 2006. 2

[52] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2, 8

[53] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 8

[54] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, 2021. 6, 7

[55] Sebastian Thrun. Probabilistic robotics. *Commun. ACM*, 2002. 1

[56] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, 2013. 3, 4

[57] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *ECCV*, 2020. 4

[58] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *CVIU*, 2000. 1

[59] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle Adjustment — A Modern Synthesis. In *Vision Algorithms: Theory and Practice*. Springer Berlin Heidelberg, 2000. 1, 5

[60] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, 2021. 1

[61] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *ICCV*, 2023. 6, 7

[62] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Visual Geometry Grounded Deep Structure From Motion. In *CVPR*, 2024. 1, 2, 3, 6, 7

[63] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022. 2

[64] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 2, 3, 4, 5, 6

[65] Tong Wei, Yash Patel, Alexander Shekhovtsov, Jiri Matas, and Daniel Barath. Generalized differentiable ransac. In *ICCV*, 2023. 1

[66] Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Learning super-features for image retrieval. In *ICLR*, 2022. 8

[67] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *CVPR*, 2020. 1

[68] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, 2022. 6

[69] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *ICLR*, 2024. 6

[70] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *SIGGRAPH*, 2018. 6