# Detector-Free Structure from Motion

Xingyi He[1] Jiaming Sun[1] Yifan Wang[1] Sida Peng[1]
Qixing Huang[2] Hujun Bao [1] Xiaowei Zhou[1†]

[1]Zhejiang University [2]The University of Texas at Austin

## Abstract

*We propose a structure-from-motion framework to recover accurate camera poses and point clouds from unordered images. Traditional SfM systems typically rely on the successful detection of repeatable keypoints across multiple views as the first step, which is difficult for texture-poor scenes, and poor keypoint detection may break down the whole SfM system. We propose a detector-free SfM framework to draw benefits from the recent success of detector-free matchers to avoid the early determination of keypoints, while solving the multi-view inconsistency issue of detector-free matchers. Specifically, our framework first reconstructs a coarse SfM model from quantized detector-free matches. Then, it refines the model by a novel iterative refinement pipeline, which iterates between an attention-based multi-view matching module to refine feature tracks and a geometry refinement module to improve the reconstruction accuracy. Experiments demonstrate that the proposed framework outperforms existing detector-based SfM systems on common benchmark datasets. We also collect a texture-poor SfM dataset to demonstrate the capability of our framework to reconstruct texture-poor scenes. Based on this framework, we take the* ***first place*** *in Image Matching Challenge 2023 [9]. Project page:* https://zju3dv.github.io/DetectorFreeSfM/.

## 1. Introduction

Structure-from-Motion (SfM) is a fundamental task in computer vision, which aims to recover camera poses, intrinsic parameters, and point clouds from multi-view images of a scene. The estimated camera poses and optional point clouds benefit downstream tasks, such as visual localization, multi-view stereo, and novel view synthesis.

SfM has been studied for decades, with many well-established methods [4, 10, 11, 54], open source systems such as Bundler [43] and COLMAP [40], and commercial software [1, 3] that are accurate and scalable for large-scale scenes. As a routine, they require to detect and match sparse feature points across multiple views [12, 28, 37] at the beginning of the pipeline to build multi-view point-to-point correspondences. This requirement could not be fulfilled in many cases. For example, in texture-poor regions, it is hard to robustly detect repeatable keypoints across multiple views for matching. Poor feature detection and matching become the bottleneck of the whole SfM pipeline, leading to missing image registration or even failing reconstruction of the entire model. Fig. 1 presents an example.

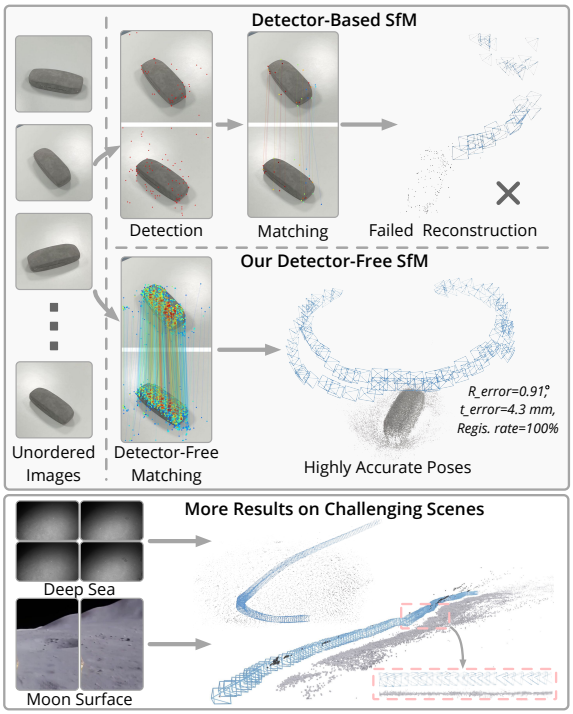


Figure 1. **Comparison between traditional detector-based SfM and the proposed detector-free SfM.** For the texture-poor scene, detector-based SfM fails due to the poor repeatability of detected keypoints at the beginning, while our detector-free SfM framework leverages detector-free matching and achieves complete reconstruction with highly accurate camera poses. Our framework is applicable to real-world challenging scenes such as the deep sea and the moon surface.

The authors from Zhejiang University are affiliated with the State Key Lab of CAD&CG. †Corresponding author: Xiaowei Zhou.

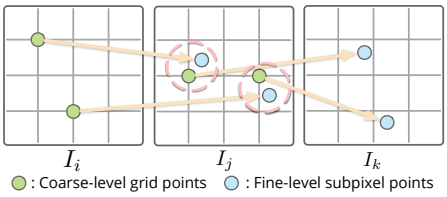


Figure 2. **Multi-view Inconsistency Issue of Detector-Free Matching.** The resulting feature locations of $I_j$ are varied when $I_j$ is matched to $I_i$ and $I_k$, yielding fragmentary feature tracks.

Recently, detector-free matchers [8, 44, 51] have achieved state-of-the-art performance on the image matching task. They have shown a strong capability to match low-textured regions with the help of the detector-free design and the attention mechanism [49]. They often use a coarse-to-fine matching strategy for efficiency. The dense matching on a coarse grid is first performed between down-sampled feature maps of two images. Then, the feature locations of coarse matches on one image are fixed, while their subpixel correspondences are searched on the other image with fine-level feature maps. Therefore, the resulting locations of features in an image depend on the other image, as shown in Fig. 2. This pair-dependent nature leads to fragmentary feature tracks when running pairwise matching over multiple views, which makes detector-free matchers not directly applicable to existing SfM systems.

In this paper, we propose an SfM framework that is able to leverage the recent success of detector-free matching and recover highly-accurate camera poses even for texture-poor scenes. An overview of our pipeline is depicted in Fig. 3. To solve the inconsistency issue of detector-free matching, our SfM framework reconstructs the scene in a coarse-to-fine manner, which first builds a coarse SfM model with the quantized matches, and then iteratively refines the model towards higher accuracy.

Specifically, our framework first matches image pairs with a detector-free feature matcher, e.g., LoFTR [44]. Then, in the coarse reconstruction phase, we quantize the feature locations by rounding them into a coarse grid to improve consistency and reconstruct a coarse SfM model. This coarse model provides initial camera poses and scene structures for the later refinement phase. Next, we propose an iterative refinement pipeline that alternates between a feature track refinement phase and a geometry refinement phase to improve pose and point cloud accuracy. The feature track refinement module is built on a novel transformer-based multi-view matching network, which enhances the discrimitiveness of extracted features by encoding multi-view contexts with self- and cross-attention mechanisms. Based on refined feature tracks, the geometry refinement module uses bundle adjustment and track topology adjustment to improve the accuracy of camera poses and point clouds.

Experiments on the public ETH3D dataset [41] and Image Matching Challenge (IMC) [21] dataset demonstrate that our detector-free SfM framework outperforms state-of-the-art detector-based SfM systems with respect to various metrics. To further evaluate and demonstrate the capability of our SfM framework on challenging scenes, we also collect a texture-poor SfM dataset which is composed of 17 scenes with 1020 image bags. Thanks to the detector-free design and the iterative refinement pipeline, our framework can recover accurate camera poses with high registration rates even for challenging texture-poor scenes. Fig. 1 presents some examples.

In summary, this paper has the following contributions:

- A detector-free SfM framework built upon detector-free matchers to handle texture-poor scenes.
- An iterative refinement pipeline with a transformer-based multi-view matching network to efficiently refine both feature tracks and reconstruction results.
- A new texture-poor SfM dataset with ground-truth pose annotations.

## 2. Related Work

**Structure-from-Motion.** Feature correspondence-based SfM methods have long been investigated [6, 15, 30, 32]. Many previous work has focused on improving the efficiency and robustness of large-scale scene reconstruction [4, 5, 10, 11, 40, 54]. Some methods try to disambiguate matches when applied to scenes with highly repetitive or symmetric structures [34, 53]. As discussed in the introduction, these methods require feature detection and matching at the beginning of the pipeline. In challenging scenes, especially in texture-poor regions, poor keypoint detection will affect the overall SfM pipeline.

More recent end-to-end SfM methods propose to directly regress poses [31, 50, 56, 58] or solve poses using differential bundle adjustment (BA) [17, 45]. These methods avoid explicit feature matching and thus don't suffer from poor feature matching. However, they have limited scalability and generalizability in real-world settings. With the success of recent neural scene representations, some methods [20, 25] try to optimize poses with differentiable rendering. However, they often rely on the use of previous correspondence-based methods, e.g., COLMAP [40], to provide initial poses, as joint pose and scene optimization from scratch are difficult to converge and prone to local minima, c.f. [25, 29].

Unlike them, our detector-free SfM framework eliminates the requirement of sparse feature detection at the beginning of the pipeline, which is more robust in challenging scenarios such as low-textured regions and repetitive patterns. Moreover, our framework is scalable to large-scale scenes and can handle in-the-wild data with wide baseline and illumination changes. [52] and OnePose++ [19] also eliminate feature detection by performing coarse grid-level matching first and then refining 2D points for subpixel accuracy. Different

from their refinement that is single- or two-view-based, our framework can leverage multi-view information to refine a feature track. More comparisons and detailed discussions with OnePose++ are given in Sec. 4.4 and the supplementary materials.

**Feature Matching.** Feature Matching is often a prerequisite for SfM and SLAM. A typical feature matching pipeline [12, 13, 28, 33, 36] is to detect and describe key points in each image and then match them using nearest-neighbor search or learning-based matchers [7, 27, 37]. The merit of these methods is the high matching efficiency based on the sparse points. However, for challenging scenarios, especially regions with a low texture, poor feature detection at the beginning is the bottleneck and affects the overall SfM system.

In recent years, many methods have directly matched image pairs in a dense [48] or semidense manner [8, 23, 35, 44, 46, 51], avoiding feature detection. With the help of Transformer [49], some semi-dense matching methods [8, 44, 51] achieve higher accuracy compared to detector-based baselines and show strong capabilities in building correspondences in low-textured regions. However, due to their inconsistency problem when matching multiple views (shown in Fig. 2), it is difficult to directly apply them to the current SfM systems, as discussed in the introduction. While the rounding [8] or merging strategies [42] could be used to produce long feature tracks for SfM, these strategies sacrifice the accuracy of the match, which will significantly reduce the accuracy of the reconstructed SfM models. Unlike them, our detector-free SfM framework with a coarse-to-fine manner can recover highly accurate poses and point clouds.

**Multi-View Refinement.** Accurate multi-view correspondences are crucial for recovering accurate point clouds and camera poses in SfM. The technical challenge is that per-view detection of feature points cannot guarantee their geometric consistency among multiple views. To solve this problem, some previous methods perform multi-view refinement with flow [14] or dense features [26], which bring a significant improvement in accuracy for SfM. PatchFlow [14] first estimates the dense flow field within the local patch of each tentative pair and then refines multi-view 2D locations by minimizing the energy function based on the estimated flow. PixSfM [26] performs feature-metric keypoint adjustment and bundle adjustment to refine 2D feature locations before SfM and the entire scene after SfM, respectively. Our detector-free SfM framework may adopt these two methods to refine the quantized matches and SfM models. However, PatchFlow suffers from high computation due to pairwise flow estimations. PixSfM needs to preserve feature patches or cost maps of all 2D observations in memory for the feature-metric BA. Given that detector-free matchers produce significantly more correspondences than sparse matchers, the memory footprint of adapting PixSfM to our detector-free SfM pipeline is inevitably large, especially on large-scale scenes. Different from them, we devise a transformer-based multi-view refinement matching module, which can efficiently and accurately refine a feature track with a single forward pass. Moreover, thanks to the design of our refinement phase that separately refines feature tracks and performs geometry refinement, the geometric BA can be leveraged for efficiency in terms of speed and memory. Experimental comparisons are provided in Sec. 4.6.

## 3. Method

An overview of our detector-free SfM framework is shown in Fig. 3. Given a set of unordered images $\{\mathbf{I}_i\}$, our objective is to recover camera poses $\{\boldsymbol{\xi}_i \in \mathbb{SE}(3)\}$, intrinsic parameters $\{\mathbf{C}_i\}$ and a scene point cloud $\{\mathbf{P}_j\}$. The recovered camera poses are in a global coordinate system. To achieve this goal, we propose a two-stage pipeline, in which we first establish correspondences between image pairs with a detector-free matcher and reconstruct an initial coarse SfM model (Sec. 3.1). Then, we perform an iterative refinement to improve the accuracy of poses and point clouds (Sec. 3.2).

### 3.1. Detector-Free Matching and Coarse SfM

For a set of unordered images, our framework performs a detector-free semi-dense feature matching directly between image pairs instead of first detecting sparse keypoints as in the traditional SfM pipeline [40]. Eliminating the keypoint detection phase can help avoid poor detection affecting the overall SfM system and can benefit the reconstruction of challenging texture-poor scenes.

**Match Quantization.** Directly adapting the correspondences of semi-dense matchers for SfM is not straightforward, due to the inconsistent multi-view matches as depicted in Fig. 2 and discussed in the introduction. Our idea is to strive for match consistency by sacrificing accuracy in the coarse SfM phase. Concretely, we quantize the 2D locations of matches into a grid: $\lfloor \mathbf{x}/r \rceil * r$, where $\lfloor \cdot \rceil$ is the rounding operator and $r$ is the size of the cell of the grid. This quantization step forces multiple matches of subpixels that are close to each other to merge into a single grid node, which improves consistency. Note that the coarse-level correspondences output by some detector-free [8, 44, 51] matchers are typically at $1/8$ image resolution, which can be directly used as quantized matches. The ablation analysis of $r$ is given in Sec. 4.5.

After match quantization, we utilize these coarse matches for incremental mapping [40] to obtain a coarse SfM model. The accuracy of recovered camera poses and point clouds is limited due to match quantization, which serves as the

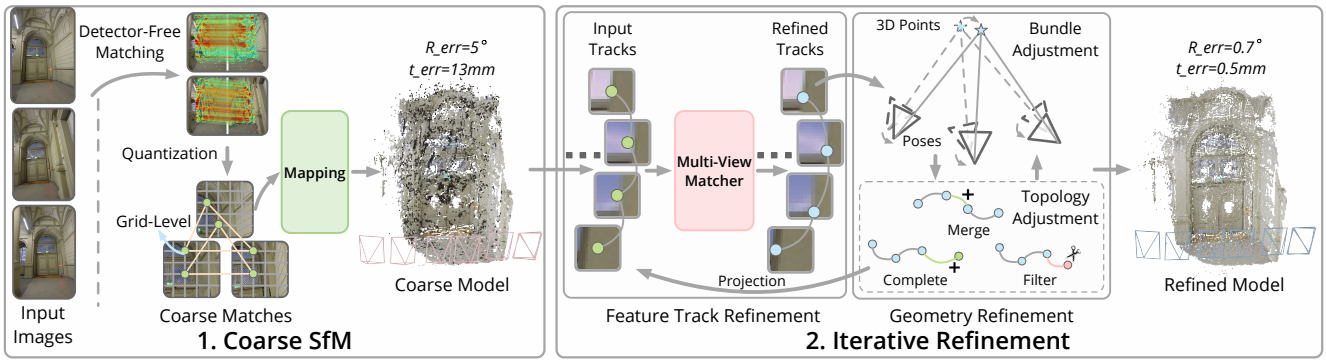


Figure 3. **Pipeline Overview.** Beginning with a collection of unordered images, the **Coarse SfM** stage generates an initial SfM model based on multi-view matches from a detector-free matcher. Then, the **Iterative Refinement** stage improves the accuracy of the SfM model by alternating between the feature track refinement module and the geometry refinement module.

initialization of our refinement framework introduced in the next section.

### 3.2. Iterative SfM Refinement

We proceed to refine the initial SfM model to obtain improved camera poses and point clouds. To this end, we propose an iterative refinement pipeline. Within each iteration, we first enhance the accuracy of feature tracks with a multi-view matching module. These refined feature tracks are then fed into a geometry refinement phase, which optimizes camera poses and point clouds jointly. The refinement process can be performed multiple times for higher accuracy. An overview is shown in Fig. 3.

#### 3.2.1 Feature Track Refinement

A feature track $\mathcal{T}_j = \{\mathbf{x}_k \in \mathbb{R}^2 | k = 1 : N_j\}$ is a set of 2D keypoint locations in multi-view images corresponding to a 3D scene point $\mathbf{P}_j$. We devised a multi-view matching module to efficiently refine feature tracks $\{\mathcal{T}_j\}$ for high accuracy, which is illustrated in Fig. 4. The basic idea is to locally adjust the keypoint locations in all views so that the correlation among their features is maximized.

As exhaustively correlating all pairs of views is computationally intractable, we select a reference view, extract the feature at the keypoint in the reference view, and correlate it with local feature maps with a size of $p \times p$ around the keypoints in other views (called query views), yielding a set of $p \times p$ heat maps that can be viewed as distributions of the keypoint locations. In each query view, we compute the expectation and variance over each heatmap as the refined keypoint location and its uncertainty, respectively. This process gives us a candidate feature track with refined keypoint locations in all query views as well as the uncertainty of this candidate track, i.e., the sum of variance over all the heatmaps. To also refine the location of the keypoint in the reference view, we sample a $w \times w$ grid of reference locations around the original keypoint in the reference view.

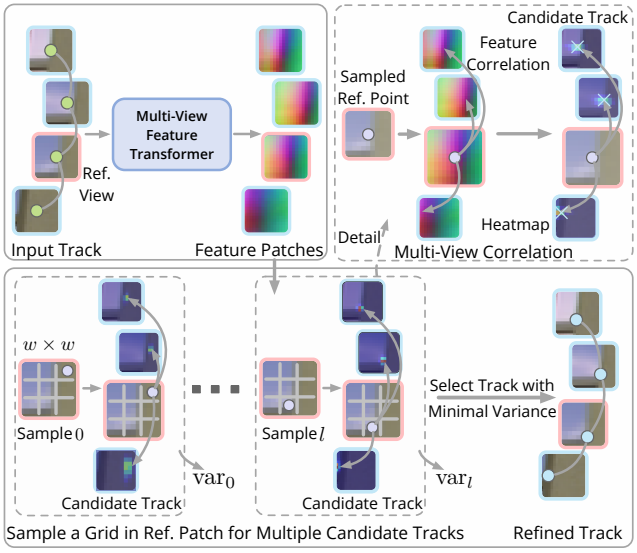


Figure 4. **Multi-View Matching Module.** Given an input feature track with a selected reference view (□), the local patches centered at the keypoints are fed into a multi-view feature transformer to extract feature patches. A $w \times w$ grid of reference locations is sampled in the reference view. For each reference location (●), its feature is correlated with the feature patches of query views (□) to obtain heatmaps that indicate the expected keypoint locations and their variances in the query views, yielding a candidate feature track. This process is repeated for all reference locations. Finally, the candidate track with the smallest variance is selected as the refined track.

Then, we repeat the above feature correlation procedure to produce a candidate feature track for each sampled reference location. Finally, the candidate track with the smallest uncertainty is selected as the refined feature track $\mathcal{T}_j^*$.

**Reference View Selection.** For each feature track, our criteria to select the reference view is to minimize the keypoint scale differences between the reference view and query views to improve the matchability. Specifically, we com-

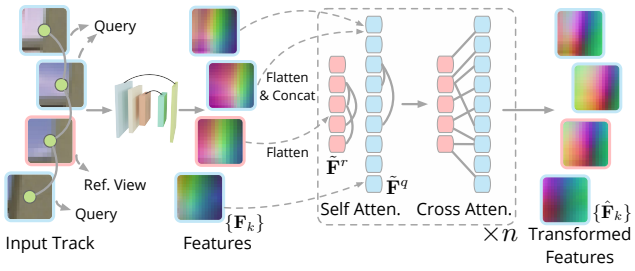


Figure 5. **Multi-View Feature Transformer.** The local patches centered at the keypoints of an input feature track are fed into a CNN to extract features and then flattened and concatenated to perform multiple self- and cross-attentions.

pute the depth values of keypoints based on the currently recovered poses and point clouds, which indicate the scale information. Then, the view with a medium scale across the track is selected as the reference view whereas the rest views are query views. More details about scale estimation can be found in the supplementary material.

**Multi-View Feature Transformer.** The multi-view matching needs to extract local feature patches centered at 2D keypoints of each $\mathcal{T}_j$. Instead of using a CNN, we design a multi-view feature transformer to enhance the discretitiveness of extracted features by encoding multi-view contexts with attention mechanisms. As shown in Fig. 5, we feed the $p \times p$ image patches centered at each keypoint into a CNN backbone to obtain a set of feature patches $\{\mathbf{F}_k \in \mathbb{R}^{p \times p \times c}\}$, where $c$ is the number of channels. Then, $\{\mathbf{F}_k\}$ is flattened to $\{\tilde{\mathbf{F}}_k \in \mathbb{R}^{m \times c}\}$, where $m = p \times p$. The flattened features of the query views are concatenated into a single query feature $\tilde{\mathbf{F}}^q$ along the first dimension. Then, we perform self- and cross-attention by $n$ times between the flattened reference feature $\tilde{\mathbf{F}}^r$ and the query feature $\tilde{\mathbf{F}}^q$ to obtain the transformed multi-view features $\{\hat{\mathbf{F}}_k\}$, which are used for feature correlation to refine the feature track.

**Training.** Besides the detector-free matcher, the only module learned in our framework is the multi-view feature transformer. It is trained on MegaDepth [24] by minimizing the average $\ell_2$ loss at keypoint locations between the refined tracks and the ground-truth tracks. We constructed training data by sampling image bags on each scene with a maximum of six images in each bag. Image bags are sampled by the covisibility extracted from the provided scene SfM model. Then, the ground-truth feature tracks in each bag are built by randomly selecting a reference image and projecting its grid points to other views by depth maps. The 2D locations of tracks in the query views are perturbed randomly by a maximum of seven pixels to generate the coarse feature tracks, which are the input of our multi-view matching module. More details are provided in the supplementary material.

#### 3.2.2 Geometry Refinement

Based on the previously refined feature tracks $\{\mathcal{T}_j^*\}$, our geometry refinement pipeline iteratively refines the poses, intrinsics, point clouds, as well as the topology of the feature tracks. Track topology means the graph structure of a set of connected 2D keypoints.

Unlike PixSfM [26] that needs to preserve feature patches or cost maps of all 2D observations in memory to perform feature-metric BA, we can directly perform efficient geometric BA [47] to optimize poses and point clouds based on the refined feature tracks. Formally, we minimize the reprojection error to optimize intrinsic parameters $\{\mathbf{C}_i\}$, poses $\{\boldsymbol{\xi}_i\}$, and 3D points $\{\mathbf{P}_j\}$:

$$E = \sum_j \sum_{\mathbf{x}_k^* \in \mathcal{T}_j^*} \rho\left(\|\pi\left(\boldsymbol{\xi}_i \cdot \mathbf{P}_j, \mathbf{C}_i\right) - \mathbf{x}_k^*\|_2^2\right) \ ,$$

where $\pi(\cdot)$ project points in the camera coordinate to image plane by $\mathbf{C}_i$, $\rho(\cdot)$ is a robust loss function [18].

After BA, we perform the feature track topology adjustment (TA) based on the refined model, which benefits further BA and multi-view matching. Since the overall scene is more accurate after multi-view refinement and BA, we adjust the topology of feature tracks by adding 2D keypoints that previously failed to be registered into feature tracks and merging the tracks that can meet the reprojection criteria at this time, following [40, 55]. Outlier filtering [40, 43, 55] is also performed to further reject points that cannot meet the maximum reprojection threshold $\epsilon$ after refinement.

We alternate BA and TA multiple times to obtain the refined poses and point clouds. Then, we project the refined point clouds to images with the current poses to update their 2D locations, which will serve as the initialization of the multi-view matching in the next refinement iteration.

### 3.3. Texture-Poor SfM Dataset

We collect an SfM dataset composed of 17 object-centric texture-poor scenes with accurate ground-truth poses. In our dataset, low-textured objects are placed on a textureless plane. For each object, we record a video sequence of around 30 seconds surrounding the object. The ground-truth poses per frame are estimated by ARKit [2] and BA post-processing, with the help of textured markers, which are cropped in the test images. To impose larger viewpoint changes, we sample 60 subset image bags for each scene, similar to the IMC dataset [21]. Example images are shown in Fig. 6 and more details are given in the supplementary material.

## 4. Experiments

### 4.1. Baselines and Datasets

**Baselines.** We compare our method with a few baseline methods in two categories: 1) Detector-based SfM

| Type | Method | ETH3D Dataset | | | IMC Dataset | | | Texture-Poor SfM Dataset | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | AUC@1° | AUC@3° | AUC@5° | AUC@3° | AUC@5° | AUC@10° | AUC@3° | AUC@5° | AUC@10° |
| Detector-Based | COLMAP (SIFT+NN) | 26.71 | 38.86 | 42.14 | 24.87 | 34.47 | 45.94 | 2.87 | 3.85 | 4.95 |
| | SIFT + NN + PixSfM | 26.94 | 39.01 | 42.19 | 26.45 | 35.73 | 47.24 | 3.13 | 4.08 | 5.09 |
| | D2Net + NN + PixSfM | 34.50 | 49.77 | 53.58 | 10.27 | 13.12 | 17.25 | 1.54 | 2.63 | 4.54 |
| | R2D2 + NN + PixSfM | 43.58 | 62.09 | 66.89 | 32.44 | 42.55 | 55.01 | 3.79 | 5.51 | 7.84 |
| | SP + SG + PixSfM | 50.82 | 68.52 | 72.86 | 46.30 | 58.43 | 71.62 | 14.00 | 19.23 | 24.55 |
| Detector-Free | LoFTR + PixSfM | 54.35 | 73.97 | 78.86 | 44.80 | 57.00 | 70.43 | 20.66 | 30.49 | 42.01 |
| | Ours (LoFTR) | **59.12** | **75.59** | **79.53** | 46.94 | 59.14 | 72.44 | 26.07 | 35.77 | 45.43 |
| | Ours (AspanTrans.) | 57.23 | 73.71 | 77.70 | **47.58** | **59.88** | **73.29** | 25.78 | 35.69 | 45.64 |
| | Ours (MatchFormer) | 56.70 | 73.00 | 76.84 | 46.32 | 58.50 | 71.99 | **26.90** | **37.57** | **48.55** |

Table 1. **Results of Multi-View Camera Pose Estimation.** Our framework is compared with detector-based and detector-free baselines on multiple datasets by the AUC of pose error at different thresholds. For all datasets, images are considered unordered for all methods. **Bold** and underline indicate the best and second-best results.

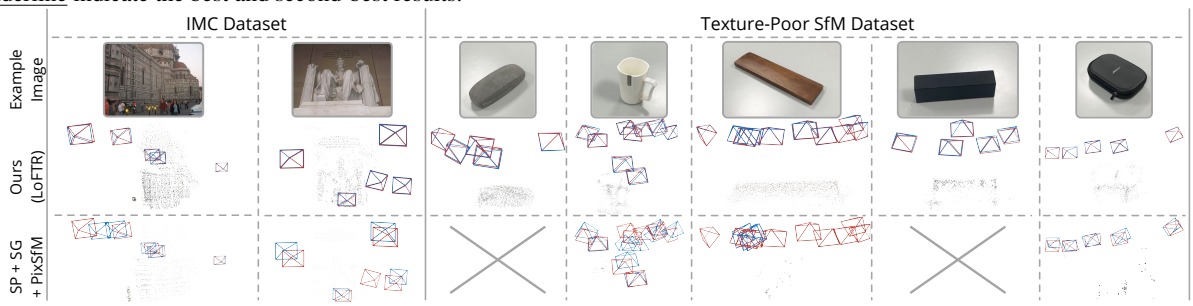


Figure 6. **Qualitative Results.** Our method with detector-free matcher LoFTR [44] is qualitatively compared with the detector-based baseline *SP + SG + PixSfM* on multiple scenes. The red cameras (▽) are ground-truth poses while the blue cameras (▽) are recovered poses.

pipeline [40] with different features, including SIFT [28], D2-Net [13], R2D2 [33] and SuperPoint (SP) [12], and matchers, including Nearest Neighbor (NN), SuperGlue (SG) [37]. All these detector-based baselines are coupled with PixSfM [26], which is the state-of-the-art SfM refinement method. 2) Detector-free SfM baseline LoFTR [44] matches with PixSfM [26] and OnePose++ [19], where these methods are fed with LoFTR's quantized matches, same as our pipeline. Note that OnePose++ can only triangulate 3D points by known camera poses, the comparison is given in Sec. 4.4.

**Datasets.** Datasets used for the evaluation include the Image Matching Challenge (IMC) 2021 dataset [21], the ETH3D dataset [41], and the proposed Texture-Poor SfM dataset. These datasets cover multiple types of scenes with different challenges. The IMC Phototourism dataset contains large-scale outdoor scenes. All nine test scenes with 1575 subsampled image bags are used for evaluation. The key challenge of this data set is the sparse views with large viewpoint and illumination changes. The ETH3D dataset contains 25 indoor and outdoor scenes with sparsely captured high-resolution images and accurately calibrated poses by Lidar as ground truth. The proposed Texture-Poor SfM dataset contains low-textured object-centric scenes with 1020 subsampled image bags in total. On all datasets, images are considered *unordered* for all methods.

## 4.2. Implementation Details

Our detector-free SfM framework is implemented with multiple detector-free matchers, including LoFTR [44], MatchFormer [51] and AspanTransformer [8], to demonstrate the compatibility of our pipeline. In the coarse SfM phase, we use their coarse-level matches ($r = 8$) for ETH3D, IMC dataset and $r = 4$ for challenging Texture-Poor SfM dataset as quantized matches for SfM [40]. Then, the refinement is performed twice. A maximum of 16 views are used for multi-view refinement matching, where longer tracks will be divided into segments and processed separately. The local patch size for feature extraction $p = 15$ and the region size for reference location search $w = 7$. The S2DNet backbone [16] is used as the CNN feature extractor and the number of attention groups $n = 2$. Linear attention [22] is used in all attention layers for efficiency. In geometry refinement, the BA and topology adjustment are alternated five times. The running time reported in the experiments was measured using four NVIDIA-V100 GPUs for parallelized matching and 16 CPU cores for BA.

## 4.3. Multi-View Camera Pose Estimation

Camera pose estimation is a central goal of SfM. This section evaluates the recovered multi-view poses.

**Evaluation Protocols.** On all datasets, matches are built exhaustively between all tentative image pairs, and the same image resizing strategy is used for all methods. For all the baselines, we follow PixSfM's implementations. The AUC

| | Method | Accuracy (%) | | | Completeness (%) | | |
|---|---|---|---|---|---|---|---|
| | | 1cm | 2cm | 5cm | 1cm | 2cm | 5cm |
| Detector-Based | SIFT + NN + PixSfM | 76.18 | 85.60 | 93.16 | 0.17 | 0.71 | 3.29 |
| | D2Net + NN + PixSfM | 74.75 | 83.81 | 91.98 | 0.83 | 2.69 | 8.95 |
| | R2D2 + NN + PixSfM | 74.12 | 84.49 | 91.98 | 0.43 | 1.58 | 6.71 |
| | SP + SG + PixSfM | 79.01 | 87.04 | 93.80 | 0.75 | 2.77 | 11.28 |
| Detector-Free | OnePose++ | 71.51 | 82.86 | 92.41 | 3.11 | 10.06 | 28.44 |
| | LoFTR + PixSfM | 74.42 | 84.08 | 92.63 | 2.91 | 9.39 | 27.31 |
| | Ours (LoFTR) | **80.24** | **88.93** | **95.82** | 3.73 | 11.07 | 29.54 |
| | Ours (AspanTrans.) | 77.34 | 87.14 | 94.86 | **4.24** | **12.93** | **34.12** |
| | Ours (MatchFormer) | 79.86 | 88.51 | 95.48 | 3.76 | 11.06 | 29.05 |

Table 2. **Results of 3D Triangulation.** Our method is compared with the baselines on the ETH3D [41] dataset using accuracy and completeness metrics with different thresholds.

of pose error at different thresholds is used as a metric to evaluate the accuracy of estimated multi-view poses, following the IMC benchmark [21] and PixSfM [26]. More details are provided in the supplementary material.

**Results.** As shown in Tab. 1, our detector-free SfM framework outperforms existing baselines over all datasets. On the ETH3D dataset with high-resolution images, our framework with LoFTR achieves the highest multi-view pose accuracy. Even when detector-based methods are further refined with PixSfM for multi-view consistency, our framework still surpasses them by a large margin. On the IMC dataset with large viewpoint and illumination changes, the detector-based baseline SP+SG+PixSfM achieves remarkable performances, while our detector-free framework consistently performs better on all metrics. The results demonstrate the robustness and effectiveness of our framework on challenging outdoor scenes with images collected from the Internett. Due to the severe low-textured scenario and viewpoint changes in the Texture-Poor SfM dataset, detector-based methods struggle with poor keypoint detection, as shown in Fig. 6. Due to the detector-free design, our framework achieves significantly higher accuracy.

Compared with LoFTR+PixSfM, the detector-free baseline that uses the same LoFTR coarse matches as ours, our framework is more accurate on all datasets and metrics, especially on the AUC@1° metric with a strict error threshold, which demonstrates the effectiveness of our iterative refinement pipeline with the multi-view matching module.

### 4.4. 3D Triangulation

With known camera poses and intrinsics, triangulating accurate scene point clouds based on image correspondences is another important task in SfM. This section evaluates the accuracy and completeness of triangulated point clouds.

**Evaluation Protocols.** The training set of ETH3D is used for evaluation, which is made up of 13 indoor and outdoor scenes with millimeter-accurate scanned dense point clouds as ground truth. We follow the protocol used in [14, 26], which triangulates the point clouds of the scene with fixed camera poses and intrinsics. Then we use the ETH3D benchmark [41] to evaluate the triangulated point clouds in terms of accuracy and completeness. The metrics are reported with different distance thresholds, including $(1cm, 2cm, 5cm)$, which are averaged across all scenes. The results of the SIFT, D2Net and R2D2 descriptors are from the PixSfM [26] paper, while the results of other baselines are obtained by running their open-source code.

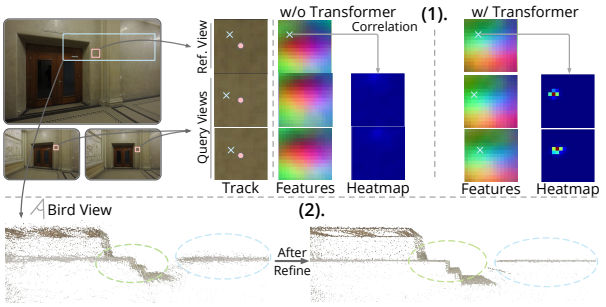


Figure 7. **Effects of Transformer and Refinement. 1.** For a feature track in the texture-poor region, its feature patches (visualized by PCA) become more discriminative after the multi-view transformer. ○ and × represent coarse and refined keypoint locations, respectively. **2.** The point cloud after refinement becomes more accurate.

**Results.** The results are presented in Tab. 2. Despite the trade-off between accuracy and completeness, our detector-free SfM framework achieves better performance on both metrics. Compared to strong detector-based baseline SP+SG+ PixSfM, our framework with LoFTR coarse matches achieves better precision with higher reconstruction completeness, thanks to the iterative refinement module. Our framework with AspanTransformer coarse matches achieves higher completeness while sacrificing a little accuracy compared to using the LoFTR matches. Compared with detector-free methods including LoFTR with PixSfM and OnePose++, our method using the same input matches achieves higher accuracy while maintaining high completeness.

### 4.5. Ablation Studies

We conduct several experiments to validate the efficacy of our design choices on the ETH3D dataset with triangulation metrics. More ablation studies with pose metrics are in the supplementary material.

**Coarse Match Quantization.** Tab. 3 (1) shows the impact of the match quantization rounding ratio $r$. Our framework achieves satisfying accuracy and completeness directly using the coarse-level matches output by LoFTR ($r = 8$). Using a smaller quantization ratio yields better matching accuracy but significantly more 2D and 3D points, thus decreasing running efficiency.

**Number of Refinement Iterations.** Tab. 3 (2) reports the results after each refinement iteration. Without refinement, the coarse SfM point cloud is inaccurate due to the match

| | | Accu. (%) | | Complete. (%) | | Time (s) |
|---|---|---|---|---|---|---|
| | | 1cm | 2cm | 1cm | 2cm | |
| (1) Quantization ratio | $r = 8$ | 80.24 | 88.93 | 3.73 | 11.07 | 557 |
| | $r = 4$ | **81.58** | **89.82** | 4.41 | 12.27 | 718 |
| | $r = 2$ | 81.18 | 89.78 | **5.41** | **14.15** | 791 |
| (2) Number of iterations | No refine. | 42.13 | 59.92 | 2.21 | 8.45 | 296 |
| | 1 iter | 77.62 | 87.04 | **3.83** | **11.44** | 430 |
| | 2 iter | 80.24 | 88.93 | 3.73 | 11.07 | 557 |
| | 3 iter | **81.26** | **89.59** | 3.57 | 10.64 | 678 |
| (3) Number of views in multi-view matching | 2 views | 69.77 | 81.69 | 2.02 | 7.10 | 438 |
| | 4 views | 72.30 | 83.42 | 2.48 | 8.35 | 435 |
| | 8 views | 74.68 | 85.02 | 3.09 | 9.84 | 431 |
| | 16 views | **77.62** | **87.04** | **3.83** | **11.44** | 430 |
| (4) Refinement designs | Full model | **80.24** | **88.93** | 3.73 | 11.07 | 557 |
| | w/o transformer | 75.12 | 85.50 | 3.00 | 9.37 | 543 |
| | w/o ref. location search | 76.66 | 86.79 | **4.16** | **12.56** | 554 |
| | w/o topology adjustment | 75.58 | 85.47 | 4.07 | 12.17 | 552 |

Table 3. **Ablation Studies.** On the ETH3D dataset, we quantitatively evaluate the impact of the quantization ratio, the number of iterations of refinement, the number of views used for multi-view matching, and other designs in refinement. The reported triangulation accuracy and completeness are averaged across all scenes, while the running time is evaluated on a single scene *Kicker*.

quantization. After the first iteration, the accuracy improves significantly, especially on the *1cm* distance threshold. Increasing the number of iterations can improve accuracy, with a slight decrease in completeness due to the track merge. Refining more than twice brings little accuracy improvement while spending more time. Therefore, we only perform refinement twice for both efficiency and accuracy.

**Maximum Number of Views in Multi-View Matching.** Tab. 3 (3) shows the effect of the number of views used for multi-view matching in a single iteration of refinement. It is shown that using more views for multi-view matching consistently improves both accuracy and completeness without significantly affecting running time.

**Refinement Designs.** Tab. 3 (4) shows the benefits of the feature transformer and reference location search in multi-view matching and the track topology adjustment in the geometry refinement. Compared with multi-view matching that directly uses backbone CNN features for matching, using multi-view transformed features can significantly improve accuracy and completeness. The result demonstrates the effectiveness of the proposed transformer module, which considers feature relations among multiple views and helps disambiguate features for more accurate matching, as visualized in Fig. 7 (1). Reference location search in the reference view brings a 3.7% improvement on the *1cm* metric. Without the track topology adjustment in geometry refinement, the point clouds' accuracy drops by 4.6% on the strict threshold (*1cm*), which demonstrates the benefits of topology adjustment on accuracy. More insights and discussions about the robustness of coarse SfM and using our multi-view transformer in PixSfM are in the supplementary material.

### 4.6. Scalability

We conduct experiments on the Aachen v1.1 dataset [38, 39, 57] to demonstrate the scalability of our framework,

| | | 500 Images | 1000 Images | 2000 Images |
|---|---|---|---|---|
| Number of 3D Points | | 553k | 1525k | 3235k |
| Ours Refinement Time (s) | | 312 | 969 | 2319 |
| BA Memory (GB) | PixSfM (Feature Map) | 161.7 | 393.8 | 904.5 |
| | PixSfM (Cost Map) | 3.79 | 9.23 | 21.2 |
| | Ours | **0.37** | **1.21** | **2.63** |

Table 4. **Running Time and BA Memory.** Our method is compared with PixSfM. Both of them use LoFTR coarse matches as input and share the same coarse SfM initialization. The refinement time and peak memory footprint during BA are reported.

following PixSfM [26]. The time and memory costs for refinement are shown in Tab. 4. We compare our method with PixSfM that uses the same LoFTR [44] coarse matches as ours, where its cost map approximation is used to reduce the memory footprint and improve efficiency. Given that there are a significant number of 2D and 3D points when using detector-free matches for SfM, the memory footprint of PixSfM's featuremetic-BA is large because it needs to preserve feature patches or cost maps of each 2D point in memory. Conversely, since we separately refine 2D points and scene geometry, the geometric-BA can be used in our pipeline, which is very efficient with a small memory footprint on large scenes and significantly outperforms PixSfM in memory efficiency.

On the scene with 2000 images, the detector-free matching and coarse SfM takes 4.2 hours, due to a large number of semi-dense matches and 3D points. Thus, the overall speed of our framework is slower than detector-based systems that are based on sparse features. More results and running time comparisons on large-scale scenes in the 1DSfM [54] dataset are shown in the supplementary material.

## 5. Conclusions

We propose a detector-free SfM framework to recover camera poses and point clouds from unordered images. In contrast to traditional SfM systems that depend on keypoint detection at the beginning, our framework leverages the recent success of detector-free matchers to avoid early determination of keypoints that may break down the whole SfM system if the detected keypoints are not repeatable, which often occurs in challenging texture-poor scenes. Extensive experiments demonstrate that our framework outperforms detector-based SfM baselines across all datasets and metrics. We believe that the proposed SfM framework opens up the possibility of reconstructing texture-poor scenes from unordered images as shown in Fig. 1 and Fig. 6 and benefits downstream tasks such as dense reconstruction and view synthesis. Please see supplementary material for discussions about limitations, failure cases, and future works.

**Acknowledgement.** This work was partially supported by National Key Research and Development Program of China (No. 2020AAA0108900) and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

## References


[1] Metashape. https://www.agisoft.com/. 1

[2] ARKit. https://developer.apple.com/augmented-reality/. 5

[3] Reality capture. https://www.capturingreality.com/. 1

[4] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *ICCV*, 2009. 1, 2

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

[6] Paul A. Beardsley, Philip H. S. Torr, and Andrew Zisserman. 3d model acquisition from extended image sequences. In *ECCV*, 1996. 2

[7] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. *ICCV*, 2021. 3

[8] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David N. R. McKinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, 2022. 2, 3, 6

[9] Ashley Chow, Eduard Trulls, HCL-Jevster, Kwang Moo Yi, lcmrll, old ufo, Sohier Dane, tanjigou, WastedCode, and Weiwei Sun. Image matching challenge 2023, 2023. 1

[10] David J. Crandall, Andrew Owens, Noah Snavely, and Daniel P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. *CVPR*, 2011. 1, 2

[11] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. *ICCV*, 2015. 1, 2

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. *CVPRW*, 2018. 1, 3, 6

[13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, 2019. 3, 6

[14] Mihai Dusmanu, Johannes L. Schönberger, and Marc Pollefeys. Multi-View Optimization of Local Feature Geometry. In *ECCV*, 2020. 3, 7

[15] Andrew William Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV*, 1998. 2

[16] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2dnet: Learning image features for accurate sparse-to-dense matching. In *ECCV*, 2020. 6

[17] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, Chengzhou Tang, and Ping Tan. Dro: Deep recurrent optimizer for structure-from-motion. *ArXiv:2103.13201*, 2021. 2

[18] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*. Wiley, 1986. 5

[19] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. Onepose++: Keypoint-free one-shot object pose estimation without CAD models. In *NeurIPS*, 2022. 2, 6

[20] Yoonwoo Jeong, Seokjun Ahn, Christopher Bongsoo Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. *ICCV*, 2021. 2

[21] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *IJCV*, 2021. 2, 5, 6, 7

[22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 6

[23] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 3

[24] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. *CVPR*, 2018. 5

[25] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. *ICCV*, 2021. 2

[26] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. *ICCV*, 2021. 3, 5, 6, 7, 8

[27] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 3

[28] G LoweDavid. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 3, 6

[29] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. *ICCV*, 2021. 2

[30] Roger Mohr, Long Quan, and Francoise Veillon. Relative 3d reconstruction using multiple uncalibrated images. *The International Journal of Robotics Research*, 1993. 2

[31] Chethan Parameshwara, Gokul Hari, Cornelia Fermuller, Nitin J. Sanket, and Yiannis Aloimonos. Diffposenet: Direct differentiable camera pose estimation. *CVPR*, 2022. 2

[32] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *IJCV*, 2004. 2

[33] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 3, 6

[34] Richard Roberts, Sudipta N. Sinha, Richard Szeliski, and Drew Steedly. Structure from motion for scenes with large duplicate structures. *CVPR*, 2011. 2

[35] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic. Neighbourhood consensus networks. *NeurIPS*, 2018. 3

[36] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. Orb: An efficient alternative to sift or surf. *ICCV*, 2011. 3

[37] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 3, 6

[38] Torsten Sattler, Tobias Weyand, B. Leibe, and Leif P. Kobbelt. Image retrieval for image-based localization revisited. In *British Machine Vision Conference*, 2012. 8

[39] Torsten Sattler, William P. Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, M. Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomás Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. *CVPR*, 2018. 8

[40] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 3, 5, 6

[41] Thomas Schöps, Johannes L. Schönberger, S. Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. *CVPR*, 2017. 2, 6, 7

[42] Zehong Shen, Jiaming Sun, Yuang Wang, Xinying He, Hujun Bao, and Xiaowei Zhou. Semi-dense feature matching with transformers and its applications in multiple-view geometry. *TPAMI*, 2022. 3

[43] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *TOG*, 2006. 1, 5

[44] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *CVPR*, 2021. 2, 3, 6, 8

[45] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment networks. In *ICLR*, 2019. 2

[46] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *ICLR*, 2022. 3

[47] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew William Fitzgibbon. Bundle adjustment - a modern synthesis. In *Workshop on Vision Algorithms*, 1999. 5

[48] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. *CVPR*, 2021. 3

[49] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[50] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *ArXiv:1704.07804*, 2017. 2

[51] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022. 2, 3, 6

[52] Aji Resindra Widya, Akihiko Torii, and M. Okutomi. Structure from motion using dense cnn features with keypoint relocalization. *IPSJ Transactions on Computer Vision and Applications*, 2018. 2

[53] Kyle Wilson and Noah Snavely. Network principles for sfm: Disambiguating repeated structures with local context. *ICCV*, 2013. 2

[54] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 1, 2, 8

[55] Changchang Wu. Towards linear-time incremental structure from motion. *3DV*, 2013. 5

[56] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, 2022. 2

[57] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *IJCV*, 2020. 8

[58] Tinghui Zhou, Matthew A. Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *CVPR*, 2017. 2