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

A. Discussion About Coarse SfM Accuracy

We conduct extensive experiments to show that our coarse SfM is robust and can support the latter refinement stage. We also emphasize that the coarse SfM phase is NOT to guarantee accuracy. Instead, it sacrifices accuracy for better completeness (registration rate). To this end, we use *quantized* detector-free matches, where the pixel threshold is relatively large in this phase (i.e., 4 pixels by default in COLMAP) in both of the matching RANSAC and mapping. Therefore, a sufficient number of images can be registered with an acceptable pose error, which serves as the initialization of the refinement phase for higher pose accuracy. Nevertheless, Tab. 1 shows that the coarse SfM alone can achieve competitive accuracy compared with the state-of-the-art detector-based method on the IMC dataset, resulting in consistently superior results after refinement.

Type	Method	AUC@3°	AUC@5°	AUC@10°	AUC@20°
Detector-Based (Reference)	R2D2 + NN + PixSfM	32.44	42.55	55.01	65.32
	SP + SG + PixSfM	46.30	58.43	71.62	80.83
Detector-Free	Ours Coarse SfM (Quant. to 8×8)	38.97	51.67	66.72	77.84
	Ours full	**46.94**	**59.14**	**72.44**	**82.45**

Table 1. Results of coarse SfM's pose accuracy on IMC 2021 dataset.

B. Discussion and Insight about Using Our Multi-View Feature Transformer in PixSfM

Since the proposed multi-view transformed feature brings improvement to our framework, an interesting exploration is to integrate it into PixSfM [16] to further bring improvement. However, we find that the problem exists with the combination of these two technicals where further developments are needed. The reason is that PixSfM utilizes extracted features to select the reference view for BA, according to Eq. 6, 7 in their paper. In contrast, extracting our multi-view transformed features needs the reference view determined before extraction as shown in the main paper's Fig. 5, which introduces conflict when applied in PixSfM. Similarly, using the multi-view transformer in PixSfM's KA phase also faces the problem of missing reference view determination before feature extraction. Therefore, direct combining is not straightforward where dedicated modifications are needed.

We emphasize that combining our multi-view transformed features with PixSfM is not the intention of our paper. The motivation to propose a new refinement module that "go back" to geometric-BA instead of directly using PixSfM's featuremetric-BA is for the purpose of efficiency, due to the fact that detector-free matchers produce a significant number of matches in SfM, especially on large-scale scenes (Feat.-BA with cost map requires ~50GB memory on *Trafalgar* scene in 1DSfM dataset [30]). Our experiments intend to show that the proposed refinement module can achieve comparable or even better accuracies than PixSfM while preserving Geo.-BA's advantage in efficiency both in terms of memory and speed. The proposed multi-view refinement matching with transformer plays an important role in achieving high accuracy in our Geo.BA-based framework.

C. Detailed Comparisons with OnePose++

OnePose++ [11] is an object pose estimation method that firstly reconstructs object point clouds from a set of posed images using LoFTR [23], and then performs 2D-3D matching between point clouds and query images for object pose estimation. The idea of its reconstruction phase is to detach LoFTR's coarse and fine matching stage that uses coarse matches for initial reconstruction to solve the multiview inconsistency problem, and then use LoFTR's fine matching stage to refine pair-wise matches for higher point cloud accuracy. In the following, we compare each phase of the method and the proposed dataset between OnePose++ and our framework.

Coarse Phase For the coarse phase, OnePose++ is limited to using LoFTR's coarse $1/8$ grid-level points, due to the requirement of its fine-level matching stage in the later

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

refinement phase. Thanks to the flexibility of our multi-view refinement network, our method can use detector-free matches with different quantization ratios (shown in the main paper's Tab. 3), or even sparse detected points (shown in Tab. 2) as input, which is more flexible for practical usage.

Refinement Phase For the refinement phase, OnePose++ only performs multiple pair-wise matching using LoFTR's fine-level matching based on a fixed reference point. Conversely, our refinement network leverages multi-view information of the whole feature track for refinement, with better multi-view consistency. Moreover, the refinement is performed once in OnePose++, where our iterative refinement mechanism is another advantage over it. The quantitative comparison is shown in the main paper's Tab. 2. Provided with the same coarse SfM results, our framework outperforms OnePose++ both in terms of accuracy and completeness. Moreover, with only a single refinement iteration (result shown in main paper's Tab. 3 (2)) as in OnePose++, we can also surpass OnePose++ on both reconstruction accuracy and completeness by a large margin, which demonstrates the capability of our multi-view refinement module.

Dataset As for the comparisons between OnePose++ dataset and our Texture-Poor SfM dataset, both of them contain texture-poor objects but the properties of images are different since they are prepared for different tasks. OnePose++ dataset is proposed for the object pose estimation that assumes using a set of posed images to estimate object poses in arbitrary different environments. Therefore, to provide accurate poses, they place textured markers around objects for better pose tracking during data capture. However, if using this dataset in our SfM evaluation task, these markers will significantly reduce the difficulty since multi-view poses can be easily recovered with strong matches on markers.

Different from it, low-textured objects are placed on a texture-less plane in our dataset. Textured markers are elaborately placed on the plane but far from the object to facilitate the pose tracking in ARKit and BA pose refinement in post-process. They are later cropped and only foreground images without salient features are used for evaluation. Example images of the proposed Texture-Poor SfM dataset are shown in the main paper's Fig. 6. Our dataset mimics the challenging real-world object capture scenario in that texture-poor objects are placed on a markless plane, which imposes significant difficulty for the SfM algorithms for accurate pose recovery.

D. Method Details

D.1. Camera Parameter Estimation Details

Like COLMAP, our method does not require known intrinsic parameters, which can be inferred from image information (EXIF if available, otherwise, using max image edge size as initialization) and refined during BA, both in coarse SfM and refinement phase. All methods are not provided with intrinsics when evaluated on the IMC and ETH3D datasets. In the image registration phase, image poses are solved by the PnP algorithm first, followed by no-linear optimization. Then the poses will be optimized simultaneously with the point cloud in BA.

D.2. Reference View Selection in Feature Track Refinement

A track $\mathcal{T}_j = \{\mathbf{x}_k \in \mathbb{R}^2 | k = 1 : N_j\}$ is divided into reference and query views and features of the reference view are correlated on query views to search for multi-view correspondences. This strategy avoids exhaustively searching correspondences between every pair within a feature track, which is a complex topology [8] and is inefficient for refinement. Our criteria for selecting the reference view is to minimize the keypoint scale differences between the reference view and query views to improve the matchability. Concretely, we define the scale s_k of a 2D observation $\mathbf{x}_k$ in a feature track $\mathcal{T}_j$ as $s_k = {}^{d_k}/_{f_i}$, where f_i is focal length in intrinsic parameter $\mathbf{C}_i$ and d_k is the depth of $\mathbf{x}_k$ that is obtained by projecting its corresponding 3D point $\mathbf{P}_j$ with the current estimated pose $\boldsymbol{\xi}_i$. Then, the view with a medium scale across the track is selected as the reference view, whereas the rest views are query views.

D.3. Multi-View Feature Transformer

The backbone from S2DNet [10] is used as the CNN feature extractor. We interpolate and fuse the output features of adaption layers in S2DNet, which are at original image resolution and $1/8$ resolution respectively, to create a single feature map.

After the feature extraction, flattening, and concatenation, we use the Linear Transformer [14] to efficiently transform the reference feature $\tilde{\mathbf{F}}^r$ and query feature $\tilde{\mathbf{F}}^q$. Linear Transformer reduces the computational complexity of the Transformer [25] from $O(N^2)$ to $O(N)$ by substituting the exponential kernel with an alternative kernel function $\mathrm{sim}(Q, K) = \phi(Q) \cdot \phi(K)^{\mathrm{T}}$, where $\phi(\cdot) = \mathrm{elu}(\cdot) + 1$. Please refer to the original paper [25] for more details.

We denote a set of self- and cross-attention layers as an attention block:

$$\begin{cases} \tilde{\mathbf{F}}'^{r}_{(l+1)} = \mathrm{SelfAtten}(\tilde{\mathbf{F}}^r_{(l)}, \tilde{\mathbf{F}}^r_{(l)}) \ , \\ \tilde{\mathbf{F}}'^{q}_{(l+1)} = \mathrm{SelfAtten}(\tilde{\mathbf{F}}^q_{(l)}, \tilde{\mathbf{F}}^q_{(l)}) \ , \\ \tilde{\mathbf{F}}^r_{l+1}, \tilde{\mathbf{F}}^q_{l+1} = \mathrm{CrossAtten}(\tilde{\mathbf{F}}'^{r}_{(l+1)}, \tilde{\mathbf{F}}'^{q}_{(l+1)}) \ . \end{cases}$$

The indices of intermediate features are indicated by $\cdot_{(l)}$. $\tilde{\mathbf{F}}'$ represents an intermediate feature processed by a self-attention layer. Our attention module sequentially performs the attention block $n = 2$ times to transform the reference and query features.

The transformed features are reshaped into feature patches $\{\hat{\mathbf{F}}_k \in \mathbb{R}^{p\times p\times c}\}$ for multi-view feature correlation.

D.4. Geometry Refinement

With refined feature tracks, we perform bundle adjustment (BA) to optimize the scene geometry by reprojection error. The Cauchy function is used as the robust loss function $\rho(\cdot)$. For efficiency, we form the reduced camera system by the Schur Complement and then solve it by dense or sparse decomposition for small- or medium-scale scenes (number of images smaller than 500), respectively. On large-scale scenes, the reduced camera system is solved by Preconditioned Conjugate Gradients algorithm (PCG) [2, 4]. Moreover, to reduce the drift during BA, we select the farthest two views in the coarse model and fix the pose of one image and one translation DoF of the other image during BA, following [22].

E. Training of Multi-View Feature Transformer

E.1. Ground Truth Generation

Our multi-view feature transformation module is trained on the MegaDepth [15], which is a large-scale outdoor dataset with 196 different scenes. To construct ground truth feature tracks for training, we first sample image bags for each scene and then project the grid-level points of a randomly selected reference view to other query views by depth maps.

Specifically, we sample 2000 image bags for each scene with a maximum of six images in each bag. The co-visibility extracted from the provided scene SfM model is used to sample image bags. We define the co-visibility ratio v of a sampled image bag as:

$$v = \frac{|\{\mathbf{P}\}_0 \cap \{\mathbf{P}\}_1 \cap \cdots \cap \{\mathbf{P}\}_i|}{\min(|\{\mathbf{P}\}_0|, |\{\mathbf{P}\}_1|, \cdots, |\{\mathbf{P}\}_i|)} \ ,$$

where $\{\mathbf{P}\}_i$ is the set of 3D points observed by image $\mathbf{I}_i$, and $|\cdot|$ is the operator that calculates the number of elements in a set. The image bags with a co-visibility ratio $0.02 < v < 0.6$ are kept for training. Moreover, the low-quality scenes reported by [8, 24] ('0000', '0002', '0011', '0020', '0033', '0050', '0103', '0105', '0143', '0176', '0177', '0265', '0366', '0474', '0860', '4541') and scenes that overlap with IMC test set ('0024', '0021', '0025', '1589', '0019', '0008', '0032', '0063') are removed from training.

After the image bag sampling, to construct ground truth feature tracks, we randomly select a reference image in the bag and project its grid-level points to other query views by the depth map, intrinsic parameters, and poses. Since the depth maps in MegaDepth are obtained by the MVS algorithm, inaccurate depth values exist. For accurate ground-truth multi-view matches, projection depth error and cycle projection error with strict thresholds are checked after the projection to filter inaccurate 2D observations in a feature track. The projection depth error e_d and cycle projection error e_c are defined as follows:

$$\begin{cases} e_d = \frac{\|\mathbf{D}_q(\mathbf{x}_{proj}) - d_{proj}\|}{\mathbf{D}_q(\mathbf{x}_{proj})} \ , \\ e_c = \|\mathbf{x}_r - \boldsymbol{\pi}_r \cdot \boldsymbol{\xi}_{r\to q}^{-1} \cdot \mathbf{D}_q(\mathbf{x}_{proj}) \cdot \boldsymbol{\pi}_q^{-1}(\mathbf{x}_{proj})\| \ , \end{cases}$$
$$\text{where } \ \mathbf{x}_{proj} = \boldsymbol{\pi}_q \cdot \boldsymbol{\xi}_{r\to q} \cdot \mathbf{D}_r(\mathbf{x}_r) \cdot \boldsymbol{\pi}_r^{-1}(\mathbf{x}_r) \ .$$

$\mathbf{x}_r$ is a sampled 2D point in reference view, $\mathbf{D}_{(\cdot)}$ is the depth map of reference or query view, $\boldsymbol{\pi}$ is the projection determined by intrinsic parameters, and $\boldsymbol{\xi}_{r\to q} = \boldsymbol{\xi}_q \cdot \boldsymbol{\xi}_r^{-1}$ is the relative pose between reference view and a query view. d_{proj} is the z value of 3D points in query view corresponding to $\mathbf{x}_{proj}$. A point in the query view is kept in the ground-truth feature track when projection depth error $e_d < 0.005$ and cycle projection error $e_c < 1px$.

E.2. Loss

The multi-view transformer module is trained by minimizing the average ℓ_2 loss on keypoint locations between the refined tracks and the ground-truth tracks. Following [23, 28], we make our loss uncertainty weighted with a variance term $\sigma^2(\mathbf{x})$:

$$\mathcal{L} = \frac{1}{N} \sum_{j\in n_t} \sum_{k\in n_v} \frac{1}{\sigma^2(\mathbf{x})} \|\mathbf{x} - \mathbf{x}_{gt}\|_2 \ ,$$

where n_t is the number of feature tracks, n_v is the number of query views in a track, and N is the total number of refined keypoints. $\sigma^2(\mathbf{x})$ is calculated by the trace of the heatmap's covariance matrix, which is detached during training to prevent the network from decreasing the loss by increasing the variance.

E.3. Training Details

The images are resized to have the longest edge of 840. The feature backbone is initialized by the pretrained weighted from S2DNet, and the attention blocks are randomly initialized. We use the AdamW optimizer to train the entire network, where the initial learning rate of backbone and attention blocks are 2×10^{-4} and 4×10^{-4}, respectively. The network training takes about 30 hours with a batch size of 8 on 8 NVIDIA V100 GPUs.

F. Texture-Poor SfM Dataset

In the proposed Texture-Poor SfM dataset, low-textured objects are placed on a texture-less plane, and video is captured surrounding each object. Each video is recorded at 30 fps for about 30 seconds in 1920×1440 resolution with per-frame poses and intrinsic parameters estimated by ARKit [1]. To stabilize the feature tracking and pose estimation in ARKit, textured markers are elaborately placed on the plane but far from the object. Then we annotate the 3D foreground region

Sparse Det. & Matcher	Refinement	ETH3D Dataset			IMC2021 Dataset		
		AUC@1°	AUC@3°	AUC@5°	AUC@3°	AUC@5°	AUC@10°
SIFT + NN	PixSfM	26.94	39.01	42.19	26.45	35.73	47.24
	Ours	**29.28**	**41.76**	**45.12**	**27.29**	**36.92**	**48.31**
R2D2 + NN	PixSfM	43.58	62.09	66.89	32.44	42.55	55.01
	Ours	**46.84**	**64.31**	**68.75**	**33.26**	**43.12**	**55.97**
SP + SG	PixSfM	50.82	68.52	72.86	46.30	58.43	71.80
	Ours	**52.66**	**70.15**	**74.85**	**46.43**	**58.51**	**72.33**

Table 2. Comparison of sparse local features accompanied with our refinement and PixSfM on ETH3D dataset and IMC2021 dataset.

for later filter backgrounds that are discriminative and can reduce the difficulty of the dataset.

After the data capture, we perform a global BA [22] to further optimize camera poses estimated by ARKit and reduce the potential drift. We extract features [7], match [19] them, and then perform triangulation using the currently estimated poses. Then the global BA is performed to optimize poses. The placed discriminative markers can also facilitate feature extraction and matching in this phase. After the pose refinement, we crop out the background with salient features, and images after crop are resized to 840×840. With the refined poses, we project the annotated foreground regions to each image to filter backgrounds with salient features, where only cropped foreground images without markers are used for evaluation.

To impose larger viewpoint changes, we sample 60 subset image bags for each scene based on co-visibility, similar to the IMC 2021 dataset [13]. Each bag contains either 5, 10, or 20 images.

G. Experiments

G.1. Datasets

On the IMC dataset, the validation set is already separated. We follow their protocol and use all nine test scenes for evaluation, and use validation scenes *Sacre Coeur*, *Saint Peter's Square*, and *Reichstag* for tuning hyperparameters. Images are resized so that the longest edge dimension is equal to 1200 pixels for all methods. As for the ETH3D dataset, the images are resized to have a maximum edge dimension of 1600 pixels for all methods. Since only training data GT 3D dense reconstructions are given in ETH3D, we can only use 13 training scenes for the evaluation of Triangulation, which is the same with PixSfM. For SfM, we can evaluate methods as long as we have GT poses. Therefore we use all training and test scenes (25 scenes in total) in ETH3D for evaluation, for more comprehensive experiments. Among 25 scenes, randomly sampled three validation scenes are *boulders*, *relief* and *relief_2*, where the rest 22 scenes are used for evaluation. On the proposed Texture-Poor SfM dataset, we use randomly selected three scenes as validation sets for tuning hyper-parameters and the remaining scenes for evaluation, where the original 840×840 image size is kept for matching. Note that due to the image crop in the post-process of the Texture-Poor SfM dataset for removing salient markers, the principle points of intrinsic parameters are not in the image center. To avoid the degeneration of estimating principle points in SfM, all of the methods are provided with known intrinsic parameters, which are kept fixed during SfM.

G.2. Details about Multi-View Camera Pose Estimation

The AUC of pose error at different thresholds is used as the metric to evaluate the accuracy of estimated multi-view poses, following the IMC benchmark [13] and PixSfM [16]. This metric converts N multi-view poses to C_N^2 pair-wise relative transformation, which is invariant to the difference of coordinate system between reconstructed and ground-truth poses. The pose error is defined as the maximum angular error in rotation and translation. Specifically, we adopt the exact AUC metric using explicit integration rather than coarse histograms for the evaluation of all three datasets, which is commonly used in [19, 23].

On the IMC dataset and Texture-Poor SfM dataset, we use pose error at $(3°, 5°, 10°)$ thresholds. Since the ETH3D dataset has high-resolution images and accurate ground truth calibration, we further report a more strict $1°$ threshold to evaluate the capability of highly accurate pose estimation.

We follow PixSfM's setting that for baseline SIFT+NN, the DEGENSAC with an inlier threshold of 0.5px is utilized for match filtering. For other baselines and our method, no additional match filtering method is incorporated, where only COLMAP's inherent geometric verification is used.

G.3. Sparse Features with Our Refinement.

Our refinement module can also be used to refine SfM models reconstructed by sparse feature detecting and matching to further bring pose improvement. Pose accuracy is evaluated on the ETH3D dataset and IMC 2021 dataset. Results shown in Tab. 2 demonstrate that our framework can consistently outperform PixSfM when accompanied by the same sparse detectors and matchers.

G.4. More Ablation Studies

In this part, we validate the effectiveness of our refinement pipeline by the multi-view pose metric on multiple datasets.

The results in Tab. 3 indicate that our iterative refinement pipeline consistently improves pose accuracy for various detector-free matchers across different datasets. As shown in Tab 4, using other reference view selection strategies, including using a view with the smallest or largest scale, and randomly selecting a reference view for each track, will reduce the final pose accuracy.

		ETH3D Dataset			IMC (*Mount Rushmore*)		
		AUC@1°	AUC@3°	AUC@5°	AUC@3°	AUC@5°	AUC@10°
LoFTR [23]	No Refine	30.88	58.90	68.06	21.26	32.09	47.96
	Iter 1	57.20	74.61	78.85	29.69	41.42	56.61
	Iter 2	**59.12**	**75.59**	**79.53**	**32.35**	**43.92**	**58.91**
AspanTrans. [5]	No Refine	28.41	55.87	65.40	19.04	29.02	44.26
	Iter 1	55.48	72.84	77.07	29.06	40.29	55.11
	Iter 2	**57.23**	**73.71**	**77.70**	**31.77**	**43.23**	**57.79**
MatchFormer [29]	No Refine	26.30	53.95	63.48	8.48	15.46	28.47
	Iter 1	54.30	71.29	75.52	24.25	35.10	49.80
	Iter 2	**56.70**	**73.00**	**76.84**	**29.31**	**39.66**	**53.32**

Table 3. **Ablation Study of Refinement Iterations.** On the ETH3D dataset and scene *Mount Rushmore* in the IMC dataset, we quantitatively evaluate the impact of the number of refinement iterations. The AUC of pose error at different thresholds is reported.

	IMC (*Mount Rushmore*)		
	AUC@3°	AUC@5°	AUC@10°
Medium Scale	**32.35**	**43.92**	**58.91**
Smallest Scale	30.63	42.11	56.99
Largest Scale	31.94	42.98	57.54
Random Selection	31.21	42.45	57.24

Table 4. **Ablation Study of Reference View Selection.** On the scene *Mount Rushmore* in the IMC dataset, we evaluate the impact of the reference view selection strategies. The AUC of pose error at different thresholds is reported.

G.5. Efficiency on Large-Scale Scenes

Experiments of the large-scale scenes are conducted on a server using 16 CPU cores (Intel Xeon Gold 6146) and four NVIDIA V100 GPUs. The subset images are uniformly sampled from the Aachen v1.1 dataset [20, 21, 34]. For each image, the top 20 most covisible images determined by image retrieval [3] are used for matching, where images are resized so that the longest edge equals 1200. To refine the large-scale scene with a large number of 3D points caused by semi-dense matchers, we perform refinement only once and use four GPUs for parallelized multi-view matching. As for geometry refinement, we use multi-core bundle adjustment [31] to leverage multiple CPU cores.

Comparison with detector-based pipelines. We show the overall running time comparison with detector-based pipelines on the four largest scenes in the 1DSfM [30] dataset in Tab. 5. On large-scale scenes, our framework is slower than detector-based pipeline with sparse features. On the one hand, detector-free matching is inherently slower than sparse matching. Moreover, due to a significant number of matches produced by detector-free matchers, the incremental mapping phase is also slower. However, on the scene with images collected from the internet and with large viewpoint and illumination changes, our framework can register significantly more images compared with sparse methods. These results also show that our framework applies to large-scale scenes (with more than 15000 images). Visualizations of

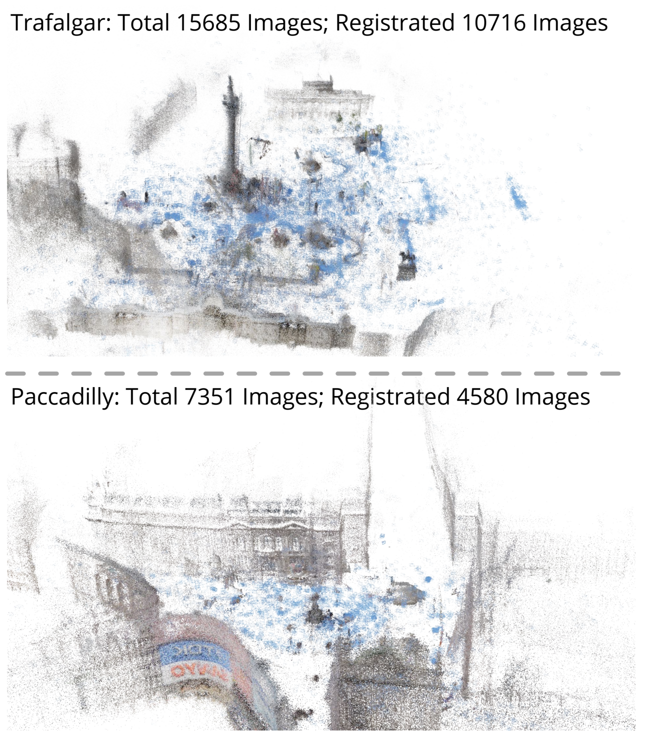
Trafalgar: Total 15685 Images; Registrated 10716 Images
Paccadilly: Total 7351 Images; Registrated 4580 Images

Figure 1. Reconstruction of scenes in the 1DSfM dataset.

Method	Trafalgar (15685 Img)		Piccadilly (7351)		Vienna Cathe. (6288)		Union Squere (5960)	
	Num. Img.	Time	Num. Img.	Time	Num. Img.	Time	Num. Img.	Time
COLMAP	7001	**3.2h**	2950	**1.8h**	1139	**1.1h**	1026	**0.9h**
COLMAP (SP + SG)	9482	6.8h	3488	3.4h	1764	2.7h	1835	2.5h
Ours	**10716**	19.7h	**4580**	11.2h	**2436**	8.8h	**2026**	8.2h

Table 5. Comparsion with detector-based methods on the 1DSfM dataset.

reconstruction are shown in Fig. 1

H. Limitations and Future Works

The main limitation of our framework is efficiency. Due to the significant number of matches produced by detector-free matches, the overall mapping phase will be inevitably slower than the previous detector-based pipelines, especially on large-scale scenes.

As future work, our framework can be extended with more advanced parallelized BA methods [12, 17] for better efficiency and integration with multi-modality data such as depth maps and IMUs if available in real applications.

H.1. Failure cases

As shown in Fig. 2, on the scenes [18] with strong duplicated structures, our framework may yield error registrations, which may come with the side effects of detector-free matchers that are capable of matching texture-poor scenes. Many previous methods [6, 18, 26, 32, 33] have focused on solving

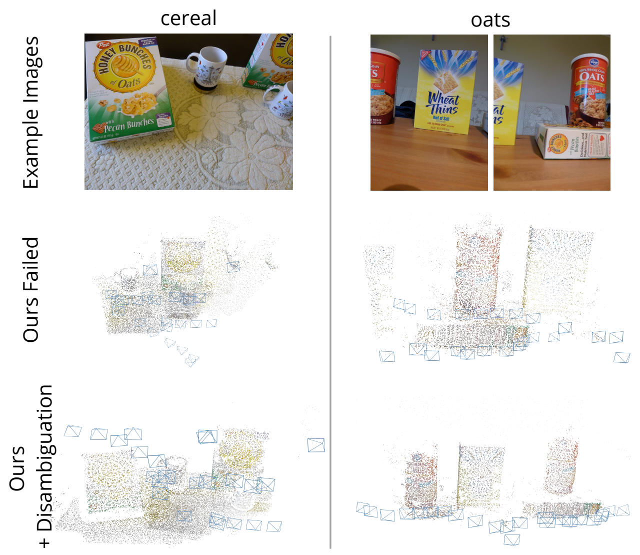

Figure 2. Failure cases of our framework on scenes [18] with strong duplicated structures. With the help of disambiguation method [6], our framework can correctly reconstruct ambiguous scenes.

scene disambiguations, which can be further integrated into our framework to alleviate this problem. With the Missing Correspondence Disambiguation [6, 33], our framework can successfully reconstruct the ambiguous scenes, as shown in Fig. 2 (row 3).

I. Real-World Scenes "Deep Sea" and "Moon Surface"

In this section, we introduce the data collection and running of challenging real-world scenes shown in the main paper's Fig. **1** and the demo video. The *Deep Sea* scene is from sequence 5 of the Aqualoc dataset [9]. This sequence was chosen because it contains a texture-poor section that presents significant challenges. The *Moon Surface* sequence is taken from an internet video that has low-texture and repetitive patterns, as well as severe motion blur.

For the scene *Deep Sea*, we use the image retrieval [3] to select the top 30 most covisible images of each image for matching. As for the *Moon Surface*, we use sequential matching to match an image with its nearest 20 frames and run our framework.

J. Application for Dense Reconstruction

To demonstrate the application of our framework that can provide accurate poses for dense reconstruction on texture-poor scenes, we run our framework on the scene *Headphone Box* and *Eyeglass Box*. Results are shown in Fig 3. The sequences are downsampled to 6fps for running our detector-free SfM framework, which provides the recovered poses

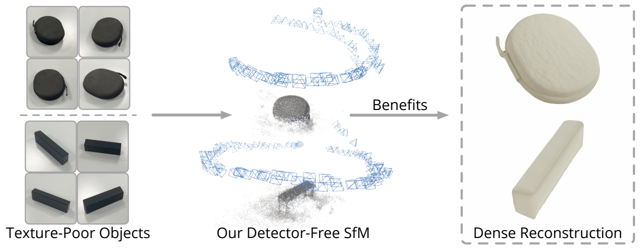

Figure 3. **Applications.** The recovered poses on texture-poor scenes by our detector-free SfM framework benefit substream tasks, e.g., dense reconstruction using neural implicit fields [27].

for neural surface reconstruction method NeuS [27] to reconstruct the scene. We manually filter the background reconstruction and only keep the object of interest for visualization.

References

[1] ARKit. `https://developer.apple.com/augmented-reality/`. 3

[2] Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle adjustment in the large. In *ECCV*, 2010. 3

[3] Relja Arandjelović, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *TPAMI*, 2015. 5, 6

[4] Martin Byröd and Kalle Åström. Conjugate gradient bundle adjustment. In *ECCV*, 2010. 3

[5] 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. 5

[6] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. *ICCV*, 2015. 5, 6

[7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. *CVPRW*, 2018. 4

[8] Mihai Dusmanu, Johannes L. Schönberger, and Marc Pollefeys. Multi-View Optimization of Local Feature Geometry. In *ECCV*, 2020. 2, 3

[9] Maxime Ferrera, Vincent Creuze, Julien Moras, and Pauline Trouvé-Peloux. Aqualoc: An underwater dataset for visual–inertial–pressure localization. *The International Journal of Robotics Research*, 2019. 6

[10] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2dnet: Learning image features for accurate sparse-to-dense matching. In *ECCV*, 2020. 2

[11] 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. 1

[12] Jingwei Huang, Shan Huang, and Mingwei Sun. Deeplm: Large-scale nonlinear least squares on deep learning frameworks using stochastic domain decomposition. *CVPR*, 2021. 5

[13] 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. 4

[14] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 2

[15] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. *CVPR*, 2018. 3

[16] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. *ICCV*, 2021. 1, 4

[17] Jie Ren, Wenteng Liang, Ran Yan, Luo Mai, Shiwen Liu, and Xiao Liu. Megba: A gpu-based distributed library for large-scale bundle adjustment. In *ECCV*, 2022. 5

[18] Richard Roberts, Sudipta N. Sinha, Richard Szeliski, and Drew Steedly. Structure from motion for scenes with large duplicate structures. *CVPR*, 2011. 5, 6

[19] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 4

[20] Torsten Sattler, Tobias Weyand, B. Leibe, and Leif P. Kobbelt. Image retrieval for image-based localization revisited. In *British Machine Vision Conference*, 2012. 5

[21] 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. 5

[22] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 3, 4

[23] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *CVPR*, 2021. 1, 3, 4, 5

[24] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *NeurIPS*, 2020. 3

[25] 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

[26] Lei Wang, Lin-Lin Ge, Shan Luo, Zi Jun Yan, Zhaopeng Cui, and Jieqing Feng. Tc-sfm: Robust track-community-based structure-from-motion. *ArXiv*, abs/2206.05866, 2022. 5

[27] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 6

[28] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, 2020. 3

[29] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022. 5

[30] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 1, 5

[31] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore bundle adjustment. *CVPR*, 2011. 5

[32] Qingan Yan, Long Yang, Ling Zhang, and Chunxia Xiao. Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context. *CVPR*, pages 152–160, 2017. 5

[33] Christopher Zach, Arnold Irschara, and Horst Bischof. What can missing correspondences tell us about 3d structure and motion? *CVPR*, pages 1–8, 2008. 5, 6

[34] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *IJCV*, 2020. 5