CoMatcher: Multi-View Collaborative Feature Matching

Supplementary Material

In this supplementary material, we provide the following elements:

- Details of the groupwise matching pipeline.
- Implements of image grouping.
- Additional experiments results on Structure from Motion.
- Method details.
- Experimental details.
- More qualitative results in complex scenes.

A. Details of groupwise matching framework

The proposed approach follows a sequential processing pipeline (Fig. 1). Given an uncontrolled image set $\mathcal{I} = \{I_i \mid i = 1, \dots, N_I\}$ capturing a scene, our method outputs a set of tracks $\{\mathcal{T}_k \mid k = 1, \dots, N_X\}$ corresponding the N_X 3D points.

To this end, We first **extract local features** $\mathcal{F}_i = \{(\mathbf{p}_k^{I_i}, \mathbf{d}_k^{I_i}) \mid k = 1, \dots, N_F\}$ for each image I_i [3, 7, 17]. After that, we **pre-compute a overlap matrix** $\mathbf{O} \in [0, 1]^{N_I \times N_I}$ by retrieval techniques [1, 9, 14]. The output of this stage is a overlap graph, where the nodes represent images and the edges indicate the presence or absence of overlap relationships between image pairs. The weight of each edge quantifies the degree of co-visibility.

To effectively leverage the complementary information provided by multi-view, we then **group images**, i.e., finding subsets of correlated images with overlapping content from the unordered set. Formally, this can be treated as a clustering problem: $\mathcal{I} = \{\mathcal{G}_1, \dots, \mathcal{G}_{N_G}\}$. After that, to leverage the constraints inherent in the multi-view feature space, we explicitly **compute the matches within each group**. The matching is performed using any two-view matching method [6, 11] and refined by removing outliers through geometric verification [4]. We assume that these verified matches are highly reliable and use them serve as priors to guide the subsequent matching process.

Next, we focuses on **calculating the matches for all remaining co-visible image pairs**. Our computation follows a groupwise paradigm, as shown in Fig. 1. Taking a specific group $\mathcal{G}_s$ as an example, there exist numerous images within the entire image set $\mathcal{I}$ that share co-visibility with $\mathcal{G}_s$. Subsequently, we iteratively select a target view from this subset and perform many-to-one matching using the CoMatcher network, acquire all the matching results for $\mathcal{G}_s$. The same computation is then repeated for each group, ultimately yielding the complete set of correspondences for all co-visible image pairs.

Finally, robust estimation techniques [2, 4] are employed to **validate all matching results** on a pairwise basis. These verified two-view correspondences are **linked into tracks** through multi-view consistency.

Compared to classical frameworks such as Colmap [12], the core distinction of our approach lies in the grouping and matching stages, while the remaining steps retain the original implementations.

Algorithm 1: Images Grouping Algorithm

Input: V: Node set, O: Edge weights, $\theta_{\min}$, $\theta_{\max}$: Thresholds, $N_{\mathcal{G}}^{\max}$: Max group size
Output: Groups $\{\mathcal{G}_1, \mathcal{G}_2, \dots\}$

Degree: $d(v) = |\{u \in V \mid O_{v,u} \text{ exists}\}|$
Mark all $v \in V$ as unassigned;
while *exists unassigned* $v \in V$ **do**
  $\mathcal{G} \leftarrow \emptyset$; $i \leftarrow \text{argmax}_{v \in V} d(v)$;
  $\mathcal{G} \leftarrow \mathcal{G} \cup \{i\}$; Mark i as assigned; $N \leftarrow 1$;
  while $N < N_{\mathcal{G}}^{\max}$ **do**
    foreach *unassigned* $v \in V$ *with* $O_{v,u}$ *defined for some* $u \in \mathcal{G}$ **do**
      $\text{score}(v) \leftarrow \frac{\sum_{u \in \mathcal{G}, O_{v,u} \text{ exists}} O_{v,u}}{d(v)}$;
    $C \leftarrow \{v \in V \mid \theta_{\min} < \text{score}(v) < \theta_{\max}, v \text{ unassigned}\}$;
    if $C = \emptyset$ **then**
      break
    $j \leftarrow \text{argmax}_{v \in C} \text{score}(v)$;
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{j\}$; Mark j as assigned; $N \leftarrow N + 1$;
return $\{\mathcal{G}_1, \mathcal{G}_2, \dots\}$;

B. Images grouping algorithm

We impose a maximum size $N_G^{\max}$of each group, based on the performance constraints of the device.

Each group is expected to satisfy two key criteria: (1) mutual visibility between each pairs and (2) a reasonable degree of overlap. Excessive overlap adds limited value, as it often stems from images captured from nearly identical viewpoints. To address these, we propose a search algorithm designed to identify correlated subsets which utilizes the connectivity of each node (the number of co-visible images) and the weight of edges (the degree of overlap).

As shown in Alg. 1, We iteratively search to construct image groups. Specifically, to form a new group, we first select the image with the highest connectivity among the ungrouped images—namely, the image with the largest number of overlapping views—as the initial image for the current group. Subsequently, for all images overlapping with the initial image, we calculate co-visibility scores based on the weights of their edges and filter potential candidates us-

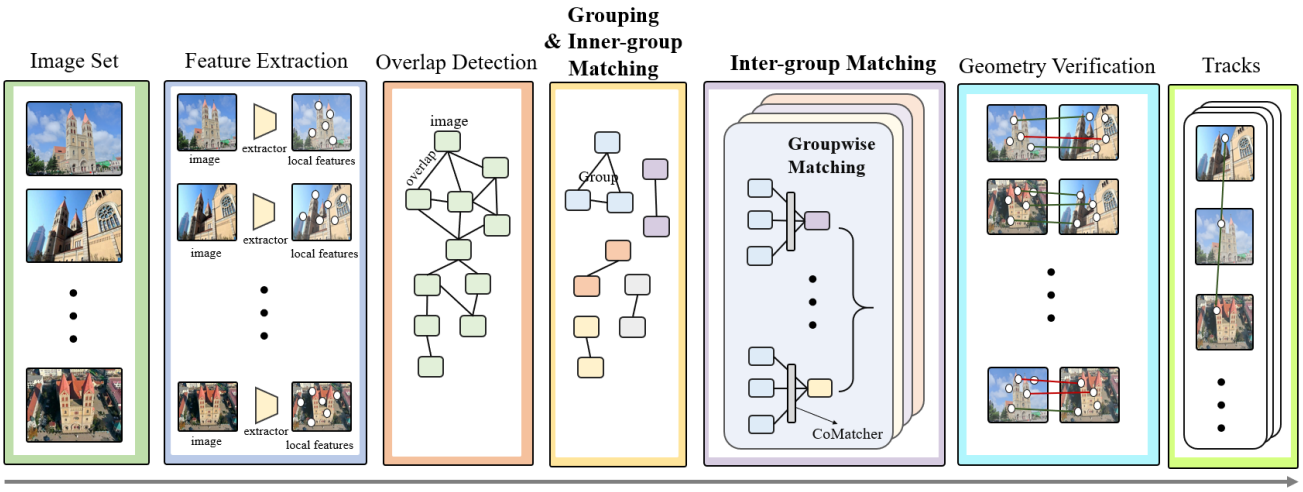

Figure 1. Groupwise matching pipeline

ing a predefined threshold. The candidate with the highest score is then added to the group. This process is repeated iteratively until no suitable candidates remain or the group reaches its maximum allowable size.

C. Additional results

We additionally evaluate our method for Structure from Motion (SfM) on the MegaDepth [5] and ETH-Colmap benchmarks [13]. Unlike previous evaluations, this section primarily focuses on two key metrics: the number of landmarks (NL) and the track length (TL), which represents the average number of observations per landmark. Using SuperPoint [3] to extract local features, we compared our method with two-view matching approach: NN+mutual and LightGlue [6].

We selected a test scenes from the MegaDepth dataset and two smaller scenes from ETH-COLMAP benchmark [13] for reconstruction. In each scene, we sampled 50 images from sparse viewpoints and extracted 2048 keypoints from each image. The matches obtained from different matching methods were reconstructed using COLMAP [12], with the default settings maintained.

We acquire a greater number of landmarks and longer tracks compared to the two-view matching method (See Tab. 1). This improvement is attributed to the ability of CoMatcher to leverage multi-view features and multi-view consistency to infer globally optimal correspondences, which are often more reliable.

Scene	Method	NL	TL
Sacre Coeur	SuperPoint+NN	18.1k	6.57
	SuperPoint+LightGlue	18.7k	6.92
	SuperPoint+**CoMatcher**	**19.3k**	**7.21**
Fountain	SuperPoint+NN	11.9k	4.62
	SuperPoint+LightGlue	12.6k	5.08
	SuperPoint+**CoMatcher**	**13.1k**	**5.31**
Herzjesu	SuperPoint+NN	10.4k	3.82
	SuperPoint+LightGlue	11.7k	4.12
	SuperPoint+**CoMatcher**	**12.4k**	**4.37**

Table 1. **Structure from Motion on MegaDepth and ETH-COLMAP.** We report the number of landmarks (NL) and the track length (TL) of reconstruction.

D. Method details

D.1. Architecture

GNN unit: Given the point feature $\mathbf{f}_u^{I_i}$ and a message point set $\mathcal{W}$, each GNN unit learns to integrate the message vector from $\mathcal{W}$ with $\mathbf{f}_u^{I_i}$ to update [6, 11, 15]:

$$\mathbf{f}_u^{I_i} \leftarrow \mathbf{f}_u^{I_i} + \mathrm{MLP}\left(\left[\mathbf{f}_u^{I_i} \mid \mathbf{m}_u^{I_i \leftarrow \mathcal{W}}\right]\right). \quad (1)$$

Here, $[\cdot \mid \cdot]$ denotes the concatenation and MLP represents a multi-layer perception. The message vector $\mathbf{m}_u^{I_i \leftarrow \mathcal{W}}$ is computed through an attention mechanism [18], representing a form of feature interaction between point u and all points in $\mathcal{W}$.

The update MLP consists of a single hidden layer with a dimension of $d_h = 2d$, followed by a LayerNorm operation, a GeLU activation, and a linear projection from $(2d, d)$ with a bias term.

Self-attention: CoMatcher first performs self-attention at each layer, where each point attends to all points within the same image. For each point u in I_i, the attention score is

computed using a relative positional encoding scheme consistent with LightGlue [6]:

$$a_{uv}^{I_iI_i} = \left(\mathbf{q}_u^{I_i}\right)^\top \mathbf{R}\left(\Delta \mathbf{p}_{uv}^{I_i}\right) \mathbf{k}_v^{I_i}. \tag{2}$$

where $\mathbf{R}(\cdot)$ is a rotary encoding of the relative position between the points.

Two-view cross-attention: For each source-target view pair, each point in I_i attends to all points in I_t, and vice versa [11]. This results in a mutual computation performed twice. Taking a point u in I_i as an example query, the attention scores are computed as:

$$a_{ux}^{I_iI_t} = \left(\mathbf{q}_u^{I_i}\right)^\top \mathbf{k}_x^{I_t}, \tag{3}$$

where $\mathbf{q}_u^{I_i}$ and $\mathbf{k}_x^{I_t}$ represent the linearly transformed feature embeddings of the corresponding point features.

In the two-view cross-attention module where the source view serves as the query, we embed the multi-view feature correlation strategy.

Implementation of rotary encoding: Following [6], we devide the space into $d/2$ subspaces, each of which is rotated by an angle determined:

$$\mathbf{R}(\mathbf{p}) = \begin{pmatrix} \hat{\mathbf{R}}\left(\mathbf{b}_1^\top \mathbf{p}\right) & & 0 \\ & \ddots & \\ 0 & & \hat{\mathbf{R}}\left(\mathbf{b}_{d/2}^\top \mathbf{p}\right) \end{pmatrix}, \tag{4}$$

where

$$\hat{\mathbf{R}}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{5}$$

$\mathbf{b}_k \in \mathbb{R}^2$ is a learned basis.

D.2. Loss

Correspondence loss: For each pair I_i and I_t, we perform two-view transformations using relative poses or homography to compute matches ground truth labels $\mathcal{C}_{i,t}$, following prior works [6, 11, 15]. If no other points are reprojected nearby, we label keypoints $\mathcal{C}_{i,t}^{\emptyset} \subseteq \mathcal{W}_i$ or $\mathcal{C}_{t,i}^{\emptyset} \subseteq \mathcal{W}_t$ as non-matching, where $\mathcal{W}_i$ and $\mathcal{W}_t$ represent the indices of feature points in I_i and I_t, respectively. We minimize the negative log-likelihood of the assignment matrix:

$$\begin{aligned} \mathcal{L}_{\text{corr}}(I_i, I_t) = &- \frac{1}{|\mathcal{C}_{i,t}|} \sum_{(u,x)\in\mathcal{C}_{i,t}} \log \mathbf{P}(u,x) \\ &- \frac{1}{2|\mathcal{C}_{i,t}^{\emptyset}|} \sum_{u\in\mathcal{C}_{i,t}^{\emptyset}} \log\left(1-\sigma_u^{I_i}\right) \\ &- \frac{1}{2|\mathcal{C}_{t,i}^{\emptyset}|} \sum_{x\in\mathcal{C}_{t,i}^{\emptyset}} \log\left(1-\sigma_x^{I_t}\right). \end{aligned} \tag{6}$$

This loss function is designed to balance positive and negative samples.

Ground-truth label of confidence loss: The confidence of each point in source views is quantified as the consistency probability between its correspondence estimated at the current layer and the final estimation. The ground truth label indicates whether these two estimations are consistent, and the final estimation corresponds to the results produced by the matching head after threshold-based filtering. To compute matching results at intermediate layers, we directly apply dual-softmax to the intermediate features, followed by mutual nearest neighbor matching and threshold filtering. This lightweight computation effectively supervises the learning of the confidence estimator while maintaining computational efficiency.

D.3. Grouping and Intra-group matching

As discussed in Sec. 4.6, the size of the group has a significant impact on the matching results. Additionally, since the local features of an entire group are processed during a single forward pass, memory consumption becomes a critical consideration. In our primary experiments, we set the maximum group size to 4. The co-visibility thresholds for group formation are denoted as θ_{min} = 0.3 and θ_{max}= 0.7.

The choice of intra-group matching method is flexible; by default, we adopt LightGlue [6].

E. Experimental details

E.1. Training details

Pre-training on synthetic homography datasets: Following [6, 11], we first pre-train CoMatcher on synthetic homographies of real-images. We use 150k images from the Oxford-Paris 1M distractors dataset [8] for training.

To train CoMatcher, each image is subjected to four different homography transformations, generating quadruplets consisting of three source views and one target view. We generate largely skewed homographies by randomly sampling four image corners within each quarter of the image, ensuring a convex enclosed area to avoid degeneracies. Random rotations and translations are applied while keeping the corners within the image boundaries, creating extreme perspective changes without border artifacts. Additionally, we apply a series of photometric augmentations to each image.

The extracted images are resized to 640×480 during interpolation. Correspondences with 3px symmetric reprojection error are deemed inliers, and points without any correspondence under this threshold are outliers. We extract 512 keypoints for SuperPoint [3] and 1024 keypoints for DISK [17].

Finetuning on MegaDepth: The model is fine-tuned of MegaDepth [5] with pseudo ground-truth camera poses and depth images. We sample 200 co-visible multi-view quadruplets per scene and randomly select one image as the

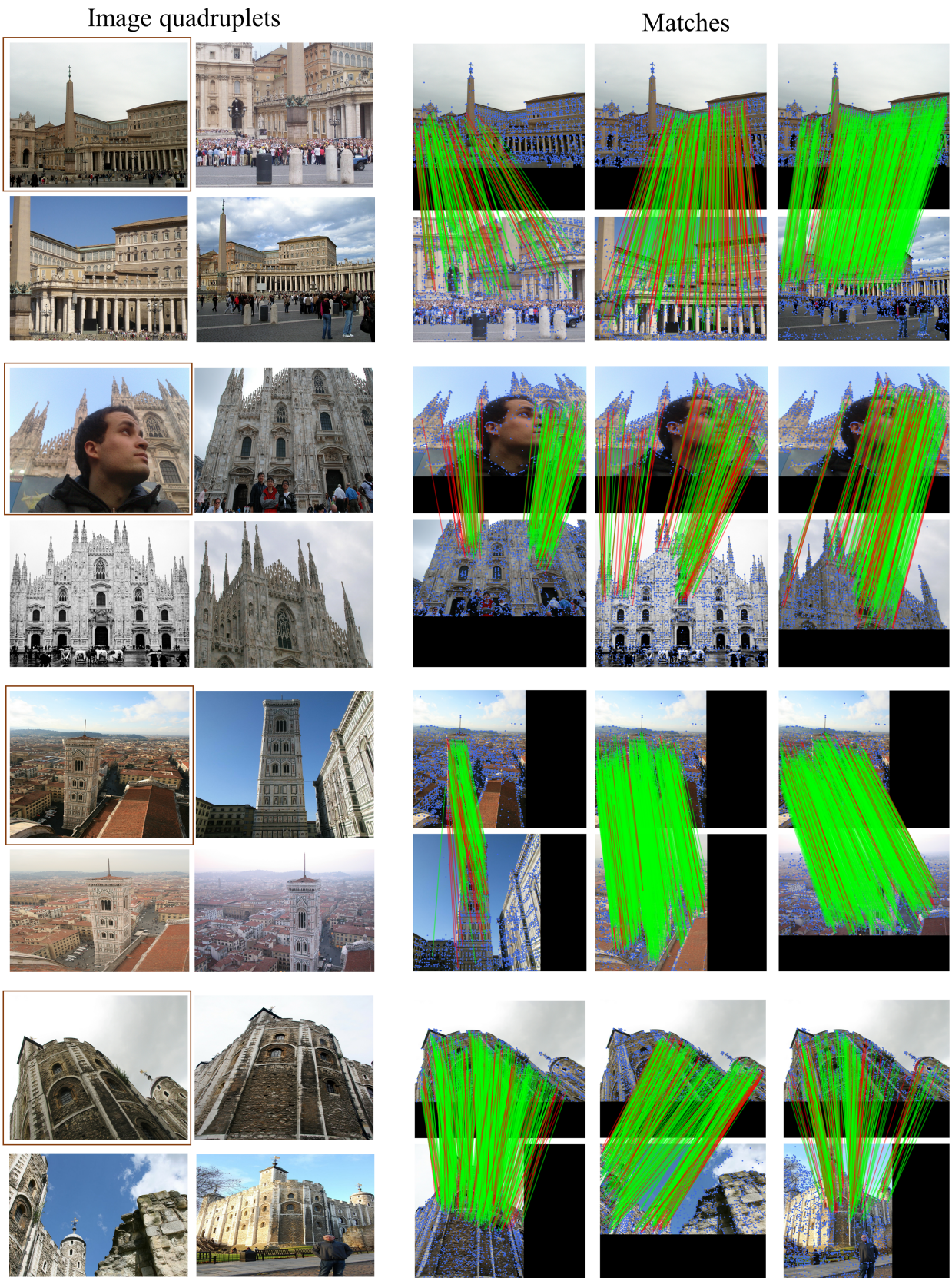

Figure 2. **Qualitative results of CoMatcher under challenging case.** In each quadruplet, the image in the top-left corner (highlighted with a red box) represents the target view.

target view for training. The sampling process is guided by the covisibility score of image pairs. We design the sampling strategy to ensure that the final training images are evenly distributed across the score intervals [0.1, 0.3], [0.3, 0.5], and [0.5, 0.9].

Images are resized such that their larger edge is 1024, and they are zero-padded to a resolution of 1024×1024. Correspondences with a reprojection error ≤ 3 pixels and mutual nearest neighbors are labeled as inliers, while those with a reprojection error > 5 pixels are labeled as outliers. Points without depth or without a correspondence having a Sampson Error ≤ 3 pixels are also marked as outliers. We extract 2048 keypoints per image.

E.2. Evaluation details

Homography estimation: HPatches is well-suited for evaluating our many-to-one matching approach. For each scene, it provides the ground truth homographies for a target view I_0 and source views $\{I_1, \ldots, I_5\}$. We use a single forward pass of CoMatcher to obtain the matches of these five pairs. End2End [10] computes all pairwise correspondences in one pass for an small image set, we extract the five required pairs for evaluation. For two-view matching methods, we perform five forward passes to compute target matches in a pairwise manner.

We assess the accuracy of the estimated homography using the mean absolute corner distance from the ground-truth homography. Following [6, 15], we resize images to a maximum edge length of 480. For each method, we fine-tune inlier threshold of RANSAC [4] and report the highest scores.

Relative pose estimation: From MegaDepth, we sampled 750 co-visible quadruplets per scene, and the difficulty is balanced based on visual overlap, following prior work [6, 15, 19]. A target view is sampled in each quadruplets, resulting in a total of 4500 image pairs. We match the sampled target view with the remaining images.

For each quadruplet, We use a single forward pass for CoMatcher and End2End [10], and perform three forward passes for two-view matching methods. We extract 2048 local features per images, each resized such that its larger dimension is 1600 pixels. For dense methods [15, 16], we retain the original settings from their respective papers for evaluation.

F. More qualitative results

We report additional qualitative results of CoMatcher under challenging case, as shown in Fig. 2.

CoMatcher effectively leverages multi-view connection to reason about spatial knowledge, such as occlusions, in complex scenes. This capability allows it to estimate reliable correspondences even in cases with partial visibility.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 1

[2] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *Pattern Recognition*, pages 236–243. Springer, 2003. 1

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 1, 2, 3

[4] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *In Communications of the ACM*, 24(6):381–395, 1981. 1, 5

[5] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2, 3

[6] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 1, 2, 3, 5

[7] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 1

[8] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting Oxford and Paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5706–5715, 2018. 3

[9] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2018. 1

[10] Barbara Roessle and Matthias Nießner. End2end multi-view feature matching with differentiable pose optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 477–487, 2023. 5

[11] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. 1, 2, 3

[12] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-Motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1, 2

[13] Johannes L Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1482–1491, 2017. 2

[14] Sivic and Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings in the IEEE/CVF International Conference on Computer Vision*, pages 1470–1477. IEEE, 2003. 1

[15] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021. 2, 3, 5

[16] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2021. 5

[17] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 1, 3

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[19] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient LoFTR: Semi-dense local feature matching with sparse-like speed. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21666–21675, 2024. 5