Supplementary Material for "CoMatch: Dynamic Covisibility-Aware Transformer for Bilateral Subpixel-Level Semi-Dense Image Matching"

Zizhuo Li Yifan Lu Linfeng Tang Shihua Zhang Jiayi Ma†
Electronic Information School, Wuhan University, Wuhan 430072, China
{zizhuo_li, lyf048}@whu.edu.cn, {linfeng0419, suhzhang001, jyma2010}@gmail.com

A. Implementation Details

A.1. Training Details

We follow the same training-test split as LoFTR [15]. The network is trained for 30 epochs using the AdamW optimizer with an initial learning rate of 1×10^{-3} and a batch size of 8. Training is conducted on 4 NVIDIA RTX 4090 GPUs and completes in approximately 22 hours.

A.2. Architecture

A.2.1. Local Feature Extraction

The local feature extraction is fleshed out by a modified ResNet-18 [9] without the bottom-up part. Specifically, we use a width of 64 and a stride of 1 for the stem and widths of [64, 128, 256] and strides of 2 for the subsequent three stages. The output of the last stage at $1/8$ image resolution is processed by our dynamic covisibility-aware Transformer (DCAT) to derive discriminative coarse-level features. The second and third stages' feature maps are at $1/2$ and $1/4$ image resolutions, respectively, which are progressively fused with transformed coarse-level features to produce cross-view perceived fine-level ones for subsequent match refinement.

A.2.2. Position Encoding

The spatial location context is essential for matching, typically modeled by absolute positional encoding (PE) [2]. However, in projective camera geometry, the position of visual observations showcases equivariance concerning the camera's translation motion within the image plane [12]. This reveals that an encoding should exclusively consider the relative but not the absolute position of keypoints. To this end, we adopt Rotary position encoding (RoPE) [14] to encode the spatial positional context between coarse-level features. More concretely, for each coarse feature i, we first decompose it into query and key vectors $\mathbf{q}_i$ and $\mathbf{k}_i$ via *different* linear transformations, then the attention score between two coarse features i and j is defined as follows:

$$a_{ij} = \mathbf{q}_i^\top \mathbf{R}(\mathbf{x}_j - \mathbf{x}_i)\mathbf{k}_j, \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the 2D image coordinates of $\mathbf{q}_i$ and $\mathbf{k}_j$, respectively, and $\mathbf{R}(\cdot) \in \mathbb{R}^{d\times d}$ is a block diagonal matrix encoding the relative position between coarse features. We partition the space into $d/2$ 2D subspaces and rotate each of them with an angle corresponding to the projection onto a learnable basis $\mathbf{b}_k \in \mathbb{R}^2$, following Fourier Features [10]:

$$\mathbf{R}(\mathbf{x}) = \begin{pmatrix} \hat{\mathbf{R}}(\mathbf{b}_1^\top \mathbf{x}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}(\mathbf{b}_2^\top \mathbf{x}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \hat{\mathbf{R}}(\mathbf{b}_{d/2}^\top \mathbf{x}) \end{pmatrix} \tag{2}$$

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

By doing so, the model can retrieve coarse feature j located at a learned relative position from i, concentrating more on interaction between features instead of their specific locations. Note that the encoding remains identical across all self-attention layers, allowing it to be computed once and then cached for reuse.

A.3. Supervision

Fine-Level Refinement Supervision. We supervise the second stage with an epipolar geometry loss $\mathcal{L}_{f2}$, as defined in Eq. (8) of the main paper, where $\mathcal{E}(\widehat{i}', \widehat{j}', \mathbf{E})$ is the Sampson distance [8] that measures the geometric error of the match $(\widehat{i}', \widehat{j}')$ *w.r.t.* $\mathbf{E}$ and is defined as:

$$\mathcal{E}(\widehat{i}', \widehat{j}', \mathbf{E}) = \frac{(\mathbf{p}_{\widehat{j}'}^\top \mathbf{E} \mathbf{p}_{\widehat{i}'})^2}{\|\mathbf{E}\mathbf{p}_{\widehat{i}'}\|_{[1]}^2 + \|\mathbf{E}\mathbf{p}_{\widehat{i}'}\|_{[2]}^2 + \|\mathbf{E}^\top \mathbf{p}_{\widehat{j}'}\|_{[1]}^2 + \|\mathbf{E}^\top \mathbf{p}_{\widehat{j}'}\|_{[2]}^2}. \tag{3}$$

Importantly, $\mathbf{p}_{\widehat{i}'}$ and $\mathbf{p}_{\widehat{j}'}$ are the homogeneous coordinates of two keypoints $\widehat{i}'$ and $\widehat{j}'$ which form a final fine-level correspondence, and $\mathbf{v}_{[k]}$ denotes the k-th element of vector $\mathbf{v}$.

†Corresponding author.

Table 1. **Image matching challenge.** mAA@10° of the pose error is reported. The superscript $*$ denotes our re-implemented version.

Method	**mAA@10° ↑**
LoFTR	78.3
MatchFormer	78.3
QuadTree	81.2
ASpanFormer	82.2
ELoFTR	81.3
CoMatch (ours)	82.3
DKM	82.6*/ 83.1
RoMa	**85.5* / 88.0**

B. Experiments

B.1. Image Matching Challenge

To further substantiate CoMatch's performance on relative pose estimation, we evaluate it on Kaggle competition Image Matching Challenge (IMC) 2022 benchmark [1], where we estimate a fundamental matrix via RANSAC and decomposing it into rotation and translation accordingly.

Dataset. IMC 2022 offers a test set comprising roughly 10,000 Google Street View images that exhibit significant visual diversity. These images are captured from a wide range of viewpoints, featuring varied aspect ratios, lighting and weather conditions, and occlusions from both pedestrians and vehicles. Notably, the evaluation dataset remains undisclosed to participants, being securely hosted on Kaggle's competition platform to ensure fair benchmarking.

Baselines. We compare CoMatch with LoFTR [15], MatchFormer [18], QuadTree [17], ASpanFormer [3], ELoFTR [19], DKM [5], and RoMa [6].

Evaluation Protocol. We calculate the mean average accuracy (mAA) between the estimated fundamental matrix and the hidden ground-truth counterpart. This assessment considers pose errors through two criteria: rotation deviation in degrees and translation discrepancy in meters. A pose is classified as accurate if it meets both thresholds. In IMC 2022, ten pairs of thresholds are considered: the rotation threshold ranges from 1° to 10° while the translation threshold spans 0.2 m to 5 m, with both thresholds following uniform distributions across their respective ranges. After that, the percentage of image pairs that meet every pair of thresholds can be determined, and the average of results over all threshold pairs is mAA.

Results. Quantitative results on IMC 2022 are reported in Tab. 1, where CoMatch outperforms all semi-dense matchers. Compared to DKM and RoMa, CoMatch is much faster (see Tab. 1 of the main paper) with comparable performance, showing a better trade-off between accuracy and efficiency.

B.2. Comparison with More Recent Semi-Dense Baselines

To further highlight the superiority of our CoMatch, we compare it with TopicFM+ [7] and PATS [13] on MegaDepth [11], ScanNet [4], and Inloc [16]. Results are presented in Tab. 2, where TopicFM+'s results on MegaDepth differ from the original since we adjust its 0.2 RANSAC threshold to standard 0.5 for fair comparison. Results show CoMatch's strong generalizability on ScanNet and Inloc. On MegaDepth, CoMatch outperforms PATS by being over 6× faster with comparable accuracy.

B.3. Comparison with Dense Matchers DKM and RoMa on Homography Estimation and Visual Localization

We also compare CoMatch with DKM and RoMa on relevant benchmarks (as shown in Tab. 3). Results demonstrate CoMatch's competitive performance against SOTA dense matchers DKM and RoMa while being significantly faster (see Tab. 1 of the main paper).

B.4. Potential of CoMatch to Surpass RoMa

To explore the potential of CoMatch, we conduct pose estimation experiments on MegaDepth, as presented in Tab. 4. By simply replacing RANSAC with LO-RANSAC (applied to all methods), CoMatch outperforms DKM and achieves performance comparable to RoMa, while being substantially faster in both matching and pose estimation (see 3rd row of Tab. 4). Furthermore, increasing the input resolution from 1152×1152 to 1312×1312 allows CoMatch* (see the last row of Tab. 4) to slightly exceed RoMa across all thresholds while still maintaining a significant speed advantage. Additionally, visual localization experiments (see Tab. 3) also demonstrate that CoMatch can surpass RoMa in downstream performance. These results collectively highlight CoMatch's strong potential to not only match but exceed RoMa in both accuracy and efficiency.

B.5. How BSR Contributes to CoMatch

In the main paper, we have ablated our BSR module in rows (e)-(g) of Tab. 4 (Sec. 4.5.1), which clearly verify its positive contribution. To further explore its effectiveness, we have retrained LoFTR with BSR. Tab. 5 reveals that our BSR module leads to significant performance improvements for LoFTR, owing to the enhanced matches with bilateral subpixel accuracy. Moreover, Fig. 1 qualitatively and quantitatively illustrates that our BSR module refines both views' keypoints that are spatially limited to the center of coarse patches (see the top left of Fig. 1) to subpixel level (see the right top of Fig. 1). This results in better keypoint distributions to express structural information (see the bottom of Fig. 1), benefiting keypoint location-sensitive pose estimation.

Table 2. **Comparison with TopicFM+ and PATS.** The runtime to match an image pair on MegaDepth is reported.

Method	MegaDepth	ScanNet	DUC1	DUC2	Time (ms) ↓
	AUC@5° / 10° / 20° ↑		(0.25m, 2°) / (0.5m, 5°) / (5.0m, 10°) ↑		
TopicFM+	54.2 / 70.5 / 82.5	20.4 / 38.3 / 54.6	52.0 / 74.7 / 87.4	53.4 / 74.8 / 83.2	135.6
PATS	61.0 / 74.2 / 83.0	20.9 / 40.1 / 57.2	55.6 / 71.2 / 81.0	58.8 / 80.9 / 85.5	773.4
CoMatch	58.0 / 73.2 / 84.2	21.7 / 40.2 / 56.7	54.5 / 75.3 / 86.9	59.5 / 84.7 / 87.8	123.8

Table 3. **Comparison with DKM and RoMa on homography estimation and visual localization.**

Method	HPatches	Aachen Day-Night v1.1		InLoc	
	AUC@3 / 5 / 10px ↑	(0.25m, 2°) / (0.5m, 5°) / (5.0m, 10°) ↑		(0.25m, 2°) / (0.5m, 5°) / (5.0m, 10°) ↑	
		Day	Night	DUC1	DUC2
DKM	70.6 / 80.1 / 88.4	88.1 / 95.3 / 98.5	72.3 / 91.1 / 97.9	50.5 / 73.7 / 84.8	53.4 / 72.5 / 74.0
RoMa	72.6 / 81.4 / 89.1	88.1 / 95.6 / 98.4	71.7 / 90.1 / 97.9	55.6 / 77.3 / 88.4	59.5 / 80.9 / 83.2
CoMatch (ours)	68.4 / 78.2 / 86.8	89.4 / 95.8 / 99.0	78.5 / 91.6 / 99.5	54.5 / 75.3 / 86.9	59.5 / 84.7 / 87.8

Table 4. **Potential of CoMatch to surpass RoMa.** The runtime to match an image pair (**Time**$_M$) and estimate the relative pose (**Time**$_E$) is reported. CoMatch* refers to CoMatch evaluated with higher-resolution image pairs as input (*i.e.*, 1312×1312).

Method	MegaDepth	Time$_M$ (ms) ↓	Time$_E$ (ms) ↓
	AUC@5° / 10° / 20° ↑		
DKM	69.07 / 80.72 / 88.73	587.9	535.2
RoMa	70.00 / 81.36 / 89.10	759.2	567.2
CoMatch (ours)	70.13 / 81.34 / 88.98	123.7	289.4
CoMatch* (ours)	70.25 / 81.46 / 89.12	175.4	366.5

Table 5. **LoFTR with BSR.**

Method	MegaDepth
	AUC@5° / 10° / 20° ↑
LoFTR	52.8 / 69.2 / 81.2
+ BSR	**55.1**$_{+4.4\%}$ / **71.1**$_{+2.7\%}$ / **82.8**$_{+2.0\%}$

Table 6. **Quantitative evaluation of covisibility scores.**

Metric	Source View	Target View
Precision	88.7	88.3
Recall	83.8	84.3

B.6. Covisibility Quantitative Evaluation

We predict soft covisibility scores per token via Eq. (2) instead of using hard masks to guide feature matching. To quantitatively evaluate the classifier, we adopt a threshold of 0.5 to classify tokens as covisible or non-covisible and compute its precision and recall on MegaDepth at 832×832 resolution. As reported in Tab. 6, the classifier achieves (88.7, 83.8) and (88.3, 84.3) for two views, respectively, showing its reliability in guiding our CGTC and CAA modules toward robust and compact context interaction.

B.7. Timing

In the main paper, we average the runtime across all image pairs in the test dataset, *i.e.*, MegaDepth [11], for efficiency evaluation, with a warm-up of 50 pairs to ensure accurate measurement. All comparative methods are implemented on a single NVIDIA GeForce RTX 4090 with 32 cores of Intel(R) Xeon(R) Platinum 8336C CPU.

In this supplementary material, we further present the average runtime per procedure of CoMatch in Tab. 7 for a more detailed efficiency analysis. We notice that a large fraction of time is spent on the coarse-level match determination, where a dual-softmax operation is used to generate the assignment matrix but may substantially increase the latency during the inference phase, particularly for high-resolution cases (*i.e.*, the large number of tokens). As the bilateral subpixel-level refinement module comprises a progressive feature fusion layer and a two-stage correlation layer, we also report their average runtime in row (d) of Tab. 7.

B.8. Impact of Condensing Range

Adaptively condensing tokens in light of their covisibility scores that are dynamically estimated within the network lays the foundation for the subsequent covisibility-assisted attention module. Thereby, we investigate the impact of different condensing ranges on the matching performance of CoMatch, with results presented in Tab. 8, where $s = 4$ serves as the default setting. Notably, employing a smaller condensing range, *i.e.*, 2×2, increases the number of reduced tokens, resulting in a slight drop in accuracy but significantly slower speed. This also underscores the suitability of our chosen condensing range parameter.

B.9. More Qualitative Results

Fig. 2 illustrates the covisibility prediction of CoMatch on four representative examples. Evidently, our approach demonstrates the capability to precisely predict the covisible regions between image pairs, benefiting our covisibility-guided token condensing and covisibility-assisted attention. Fig. 3 presents the matching results on ScanNet [4]. Com-

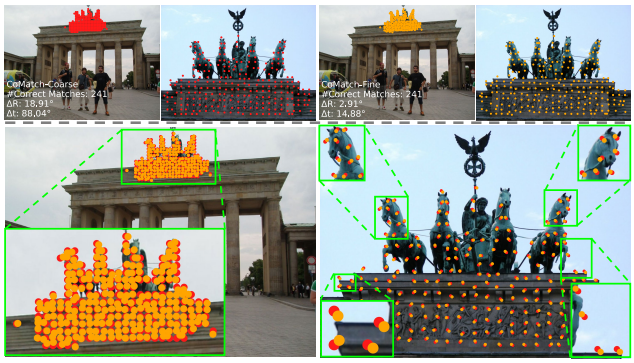

Figure 1. **Keypoint locations from coarse and fine matches that conform to the ground-truth essential matrix.** The bottom row overlays coarse and their corresponding fine keypoints (shown separately in the top row) to better illustrate the distributional changes due to refinement.

Table 7. **The average runtime per process required when matching an image pair on MegaDepth at a resolution of 1152×1152.**

Process	Time (ms) ↓
(a) Local Feature Extraction	9.7
(b) Dynamic CovisibilityAware Transformer	19.4
(c) Coarse-Level Match Determination	62.0
(d) Bilateral Subpixel-Level Refinement	32.7 (12.6 / 20.1)
Total	123.8

Table 8. **Impact of condensing range on MegaDepth.** AUC of the pose error at multiple thresholds, together with the average runtime required to match an image pair at a resolution of 1152×1152, is reported. The best results are in **bold**.

Condensing Range	Pose Estimation AUC	Time (ms) ↓
	AUC@5° / 10° / 20° ↑	
$s = 2$	57.3 / 73.0 / **84.2**	207.6
$s = 4$	**58.0 / 73.2 / 84.2**	**123.8**

pared to LoFTR [15] and ELoFTR [19], our CoMatch establishes more reliable correspondences and recovers more accurate relative camera poses.

References

[1] CVPR 2022 image matching challenge. https://www.kaggle.com/competitions/image-matching-challenge-2022/overview. Accessed June 15, 2022. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 1

[3] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, pages 20–36, 2022. 2

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 2, 3

[5] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *CVPR*, pages 17765–17775, 2023. 2

[6] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *CVPR*, pages 19790–19800, 2024. 2

[7] Khang Truong Giang, Soohwan Song, and Sungho Jo. Topicfm+: Boosting accuracy and efficiency of topic-assisted feature matching. *IEEE TIP*, 2024. 2

[8] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[10] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable fourier features for multi-dimensional spatial positional encoding. In *NeurIPS*, pages 15816–15829, 2021. 1

[11] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 2, 3

[12] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *ICCV*, pages 17627–17638, 2023. 1

[13] Junjie Ni, Yijin Li, Zhaoyang Huang, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pats: Patch area transportation with subdivision for local feature matching. In *CVPR*, pages 17776–17786, 2023. 2

[14] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 1

[15] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 1, 2, 4

[16] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, pages 7199–7209, 2018. 2

[17] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *ICLR*. 2

[18] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, pages 2746–2762, 2022. 2

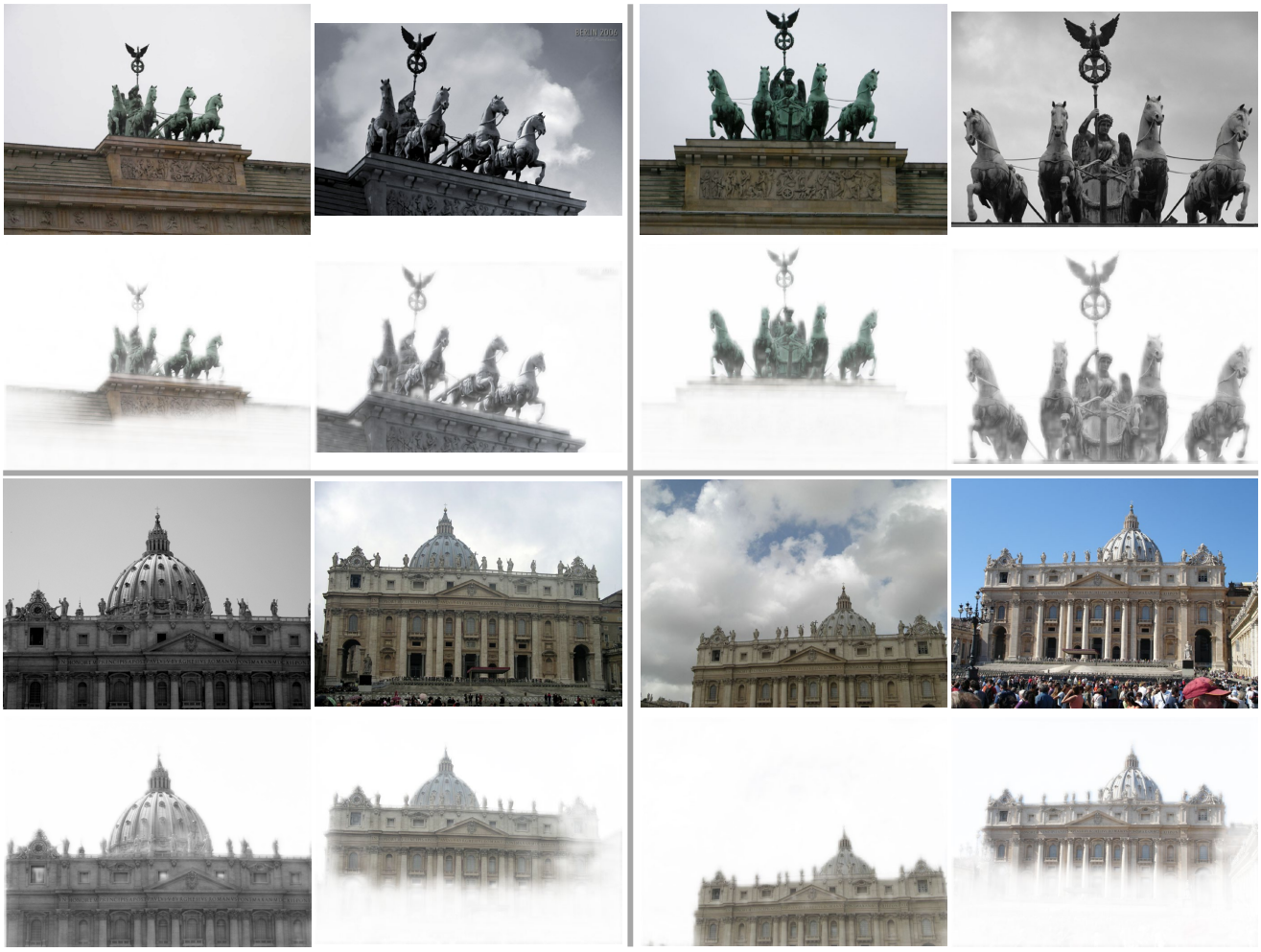

Figure 2. **Visualization of covisibility prediction.** We first bilinearly up-sample the covisibility score map to match the original image resolution, and then multiply it with the input image.

[19] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *CVPR*, pages 21666–21675, 2024. 2, 4

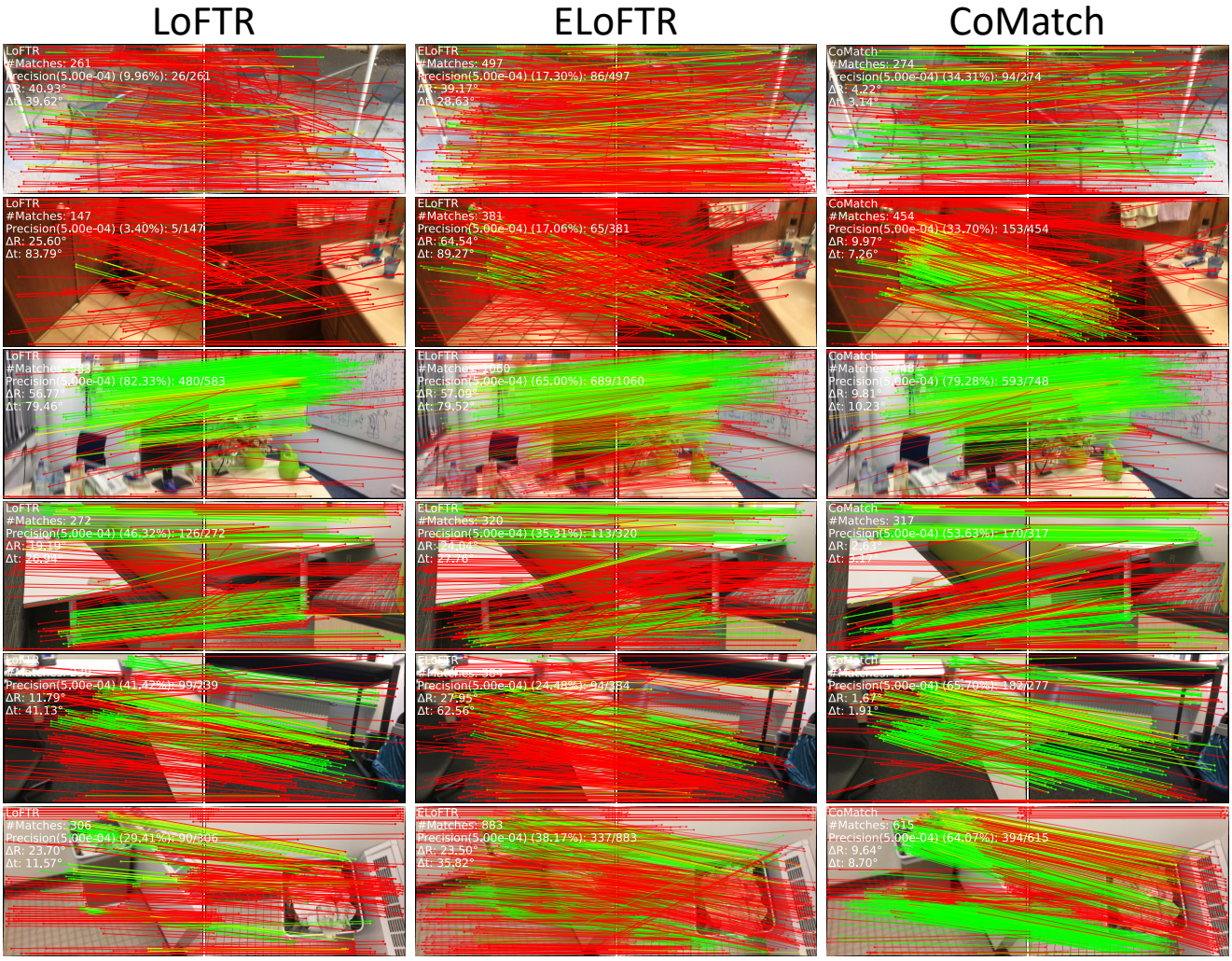

Figure 3. **Visualization of matching results on ScanNet.** A match is "——" if its epipolar error is below 5×10^{-4}, and "——" otherwise.