JamMa: Ultra-lightweight Local Feature Matching with Joint Mamba

Supplementary Material

The supplementary material for JamMa is organized as follows: Sec. A reports the evaluation setups of pose estimation experiments in the MegaDepth dataset. Sec. B provides additional details on the Mamba block and MLP-Mixer employed in JamMa. Sec. C presents further qualitative and quantitative experiments, along with discussions.

Category	Method	Image	Keypoint	Match
Sparse	XFeat	1600	4096	-
	SP + SG	1600	2048	-
	SP + LG	1600	2048	-
	DeDoDe$_B$	784	10000	-
	DeDoDe$_G$	784	10000	-
Dense	All	672	-	5000
Semi-Dense	All	832	-	-

Table S1. **Evaluation Setups in the MegaDepth [5] Dataset.**

A. MegaDepth Setups

As shown in Tab. S1, we follow the evaluation setup specified for each method to optimize their performance. For XFeat, images are resized such that the larger dimension is 1600 pixels, with 4096 keypoints extracted per image. For SuperGlue and LightGlue, the larger dimension is similarly resized to 1600 pixels, and 2048 SuperPoint keypoints are extracted per image. For DeDoDe, images are resized to 784×784, with 10000 keypoints extracted per image. For the dense methods DKM and RoMa, images are resized to 674×674, and 5000 balanced matches are sampled using the KDE-based method introduced by DKM. For all semi-dense methods [1, 8, 11, 12], images are resized and padded to 832×832. In the efficiency evaluation, we report the parameters, FLOPs and runtime of the full sparse matching pipelines, including detection, description and matching. All methods utilize LO-RANSAC with an inlier threshold of 0.5 for pose estimation.

B. Details

B.1. Mamba Block

Details of the Mamba block [4] are provided in Fig. S1 and Alg. S1. $\mathtt{B}$ and $\mathtt{N}$ denote the batch size and sequence length, respectively. $\mathtt{C}_1$ denotes the coarse feature dimension, which is 256. The SSM dimension $\mathtt{C}_\mathtt{s}$ is set to 16, and the expanded state dimension $\mathtt{C}_\mathtt{e}$ is set to 512. The input sequence $\mathbf{S}$ is first normalized by the layer normalization LN and then linearly projected to $\mathbf{X}'$ and $\mathbf{Z}$, both with a dimension size of $\mathtt{C}_\mathtt{e}$. A 1D convolution followed by the SiLU nonlinearity is applied

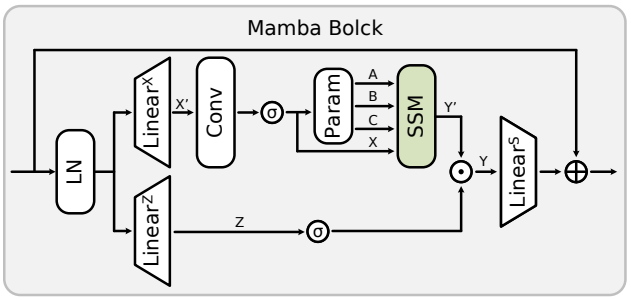

Figure S1. **Mamba Block.**

Algorithm S1 Mamba Block

Require: input sequence $\mathbf{S} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_1)$
Ensure: output sequence $\tilde{\mathbf{S}} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_1)$

$\mathbf{S}' : (\mathtt{B}, \mathtt{N}, \mathtt{C}_1) \leftarrow \mathbf{LayerNorm}(\mathbf{S})$
$\mathbf{X}' : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}) \leftarrow \mathbf{Linear}^{\mathbf{X}}(\mathbf{S}')$
$\mathbf{Z} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}) \leftarrow \mathbf{Linear}^{\mathbf{Z}}(\mathbf{S}')$
$\mathbf{X} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}) \leftarrow \mathbf{SiLU}(\mathbf{Conv}(\mathbf{X}'))$
/* compute SSM parameters, "Param" in Fig. S1 */
$\mathbf{P}^{\mathbf{\Delta}} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}) \leftarrow \mathbf{Parameter}$
$\mathbf{\Delta} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}) \leftarrow \mathbf{Softplus}(\mathbf{Linear}^{\mathbf{\Delta}}(\mathbf{X}) + \mathbf{P}^{\mathbf{\Delta}})$
$\mathbf{A}' : (\mathtt{C}_\mathtt{e}, \mathtt{C}_\mathtt{s}) \leftarrow \mathbf{Parameter}$
$\mathbf{B}' : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{s}) \leftarrow \mathbf{Linear}^{\mathbf{B}}(\mathbf{X})$
$\mathbf{A}, \mathbf{B} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}, \mathtt{C}_\mathtt{s}) \leftarrow \mathbf{Discretize}(\mathbf{A}', \mathbf{B}', \mathbf{\Delta})$
$\mathbf{C} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{s}) \leftarrow \mathbf{Linear}^{\mathbf{C}}(\mathbf{X})$
/* SSM recurrent*/
$h : (\mathtt{B}, \mathtt{C}_\mathtt{e}, \mathtt{C}_\mathtt{s}) \leftarrow$ zeros $(\mathtt{B}, \mathtt{C}_\mathtt{e}, \mathtt{C}_\mathtt{s})$
$\mathbf{Y} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}) \leftarrow$ zeros $(\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e})$
for i in {0, ..., N-1} **do**
  $h = \mathbf{A}[:, i, :, :]h + \mathbf{B}[:, i, :, :]\mathbf{X}[:, i, :, \mathtt{None}]$
  $\mathbf{Y}'[:, i, :] = h\mathbf{C}[:, i, :]$
end for
/* get gated $\mathbf{Y}$ */
$\mathbf{Y} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_\mathtt{e}) \leftarrow \mathbf{Y}' \odot \mathbf{SiLU}(\mathbf{Z})$
/* residual connection */
$\tilde{\mathbf{S}} : (\mathtt{B}, \mathtt{N}, \mathtt{C}_1) \leftarrow \mathbf{Linear}^{\mathbf{S}}(\mathbf{Y}) + \mathbf{S}$
Return: $\tilde{\mathbf{S}}$

to $\mathbf{X}'$, producing $\mathbf{X}$, which is then linearly projected to $\mathbf{B}'$, $\mathbf{C}$, and $\mathbf{\Delta}$. $\mathbf{\Delta}$ is used to discretize $\mathbf{A}'$ and $\mathbf{B}'$, resulting in $\mathbf{A}$ and $\mathbf{B}$. The state-space model (SSM) computes $\mathbf{Y}'$, which is then gated by $\mathbf{Z}$ to generate $\mathbf{Y}$. The output sequence $\tilde{\mathbf{S}}$ is obtained through the residual connection of $\mathbf{Y}$ and $\mathbf{S}$.

Note that the **for** loop in Alg. S1, *i.e.*, SSM recurrent, can be computed once by a global convolution as

$$\begin{aligned} \mathbf{K} &= (\mathbf{CB}, \mathbf{CAB}, ..., \mathbf{CA}^{N-1}\mathbf{B}), \\ \mathbf{Y}' &= \mathbf{X} \circledast \mathbf{K}, \end{aligned} \tag{S1}$$

where $\circledast$ denotes the convolution operation.

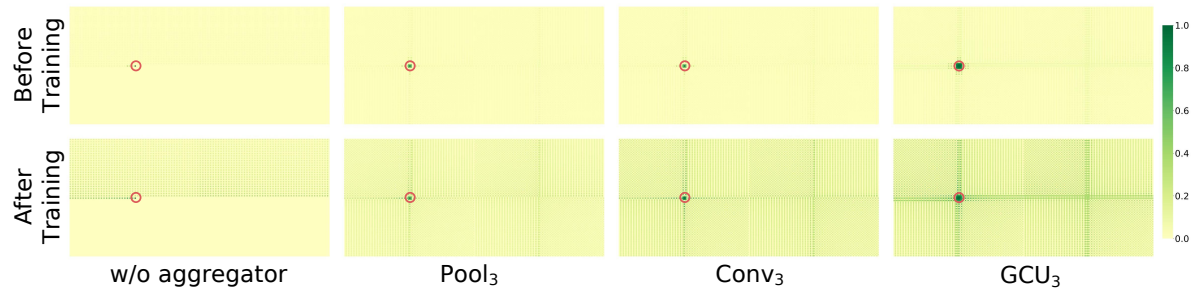

Figure S2. **Effective Receptive Field with Different Aggregators.** All three aggregators expand the local receptive field to a receptive field spread over the image pair.

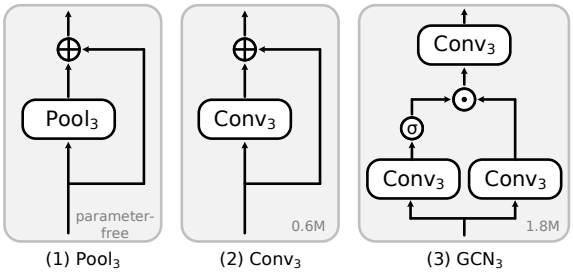

Figure S3. **Three Types of Aggregators.**

Method	Pose est. AUC		
	@5°	@10°	@20°
w/o Aggregator	62.3	75.1	84.3
(1) Pool_3	63.3	75.9	85.2
(2) Conv_3	64.2	76.9	86.0
(3) GCU_3 (JamMa)	64.5	77.3	86.3

Table S2. **Ablation Study on Aggregator.**

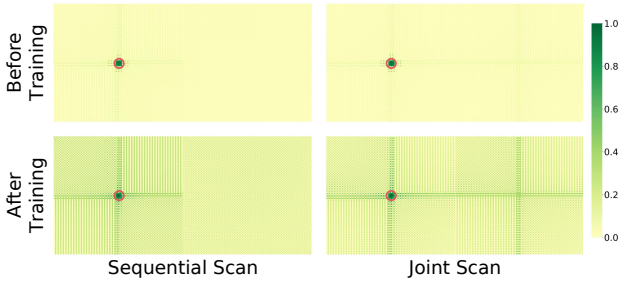

Figure S4. **Effective Receptive Field of Sequential Scan.**

B.2. MLP-Mixer

The MLP-Mixer [9] is a purely MLP-based network that first performs spatial mixing using token-wise MLP_s, followed by channel mixing using channel-wise MLP_c.

$$\begin{aligned} F_{mid} &= F_{in} + \text{MLP}_s(F_{in}), \\ F_{out} &= F_{mid} + \text{MLP}_c(F_{mid}). \end{aligned} \tag{S2}$$

In fine matching module, two 5×5 fine feature windows $\hat{F}_A^f, \hat{F}_B^f \in \mathbb{R}^{M \times 25 \times C_2}$ are spatially concatenated to form $\hat{F}^f \in \mathbb{R}^{M \times 50 \times C_2}$, which is processed by a MLP-Mixer.

In sub-pixel refinement module, two fine features $F_A^s, F_B^s \in \mathbb{R}^{M \times 1 \times C_2}$ are concatenated along the channel dimension, resulting in $F^s \in \mathbb{R}^{M \times 1 \times 2C_2}$. The MLP and Tanh activation are then employed to regress the offsets $\delta_A^x, \delta_A^y, \delta_B^x, \delta_B^y$ of the matching points in images I_A and I_B.

C. More Experiments

C.1. Ablation Study on Aggregator

As shown in Tab. S2 and Fig. S2, we conduct additional ablation studies on the aggregator. We evaluate an average pooling layer with a kernel size of 3, referred to as Pool_3, which represents the simplest parameter-free aggregator. As illustrated in Fig. S2, compared to JamMa without an aggregator, Pool_3 extends the effective receptive field to a receptive field spread over the image pair. As shown in Tab. S2(1), the minimalist aggregator Pool_3 achieves a performance improvement of $(+1.0\%, +0.8\%, +0.9\%)$, validating the importance of global dependencies and omnidirectionality. Further performance gains are observed when learnable parameters are incorporated into the aggregators, as shown in Tab. S2(2)(3). Specifically, the gated convolutional unit (GCN_3) improves the performance by $(+2.2\%, +2.2\%, +2\%)$. Additionally, we visualize the effective receptive fields of the models *before training* in Fig. S2. Training allows Mamba to establish long-distance dependencies within sequences, while the aggregator extends the sequence dependencies to global dependencies.

C.2. Effective Receptive Field of Sequential Scan.

As shown in Fig. S4, we compare the effective receptive fields of JamMa using sequential and joint scan. Sequential scan primarily emphasizes internal interactions within a single image but exhibits limited perception of the other image. In contrast, joint scan enables more comprehensive mutual interactions, making it better suited for image matching tasks that require establishing correspondences between *two images*.

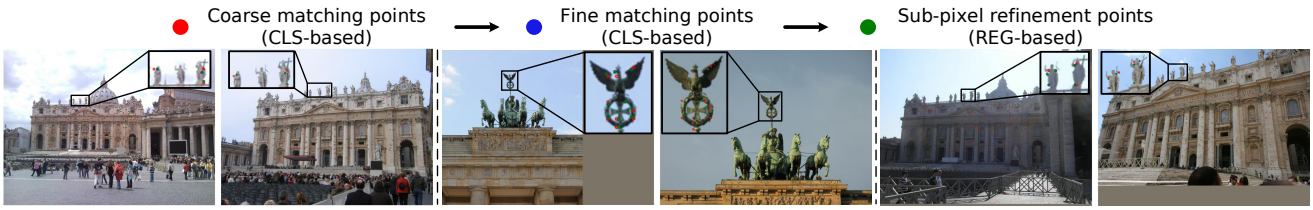

Figure S5. **Visualization of the Coarse-to-Fine Matching Module.** Zoom in for a clearer view.

Method	Day	Night
	$(0.25m, 2°)$ / $(0.5m, 5°)$ / $(1m, 10°)$	
DeDoDe$_B$ [3]	87.4 / 94.7 / 98.5	70.7 / 88.0 / 97.9
SP [2]+LG [6]	**89.6** / **95.8** / **99.2**	72.8 / 88.0 / **99.0**
LoFTR [8]	88.7 / 95.6 / 99.0	**78.5** / 90.6 / **99.0**
ASpanFormer [1]	89.4 / 95.6 / 99.0	77.5 / **91.6** / **99.0**
JamMa	87.7 / 95.1 / 98.4	73.3 / **91.6** / **99.0**

Table S3. **Visual Localization on the Aachen Day-Night Benchmark v1.1 [13].**

C.3. Visual Localization

Dataset. We evaluate our method on the Aachen Day-Night v1.1 benchmark [13], which includes 824 day-time and 191 night-time images selected as query images for outdoor visual localization.

Metric. We employ the open-source HLoc pipeline [7] for localization and report the percentage of successfully localized images under three error thresholds: $(0.25m, 2°)$, $(0.5m, 5°)$, and $(1m, 10°)$.

Results. As shown in Tab. S3, JamMa demonstrates performance comparable to LoFTR and ASpanFormer in outdoor visual localization tasks. Note that JamMa is significantly more lightweight, achieving over a $2\times$ reduction in parameters and runtime speedup.

C.4. Advantage in Low-Resolution Images.

Quantitative comparisons on low-resolution images are presented in Fig. S6. The results demonstrate that JamMa outperforms ELoFTR by a substantial margin of $+15.8\%$ at a resolution of 256, while also achieving higher speed. This highlights JamMa's potential for resource-constrained applications that demand extreme efficiency ($>$100 FPS). The superior performance of JamMa in low-resolution scenarios is attributed to shorter input sequences, which alleviate the perceptual attenuation issue of Mamba, *i.e.*, the tendency to overlook distant features within a sequence.

C.5. Visualization of Coarse-to-Fine Module.

We adopt the coarse-to-fine matching module proposed in XoFTR [10], which first performs classification-based (CLS-based) coarse matching on coarse grids, followed by classification-based fine matching on fine grids, and finally regression-based (REG-based) sub-pixel refinement. As shown in Fig. S5, coarse matching on $1/8$ resolution grids often lacks precision, particularly for small-scale images. Fine matching on $1/2$ resolution grids significantly enhances precision, while regression-based refinement further improves accuracy to the sub-pixel level.

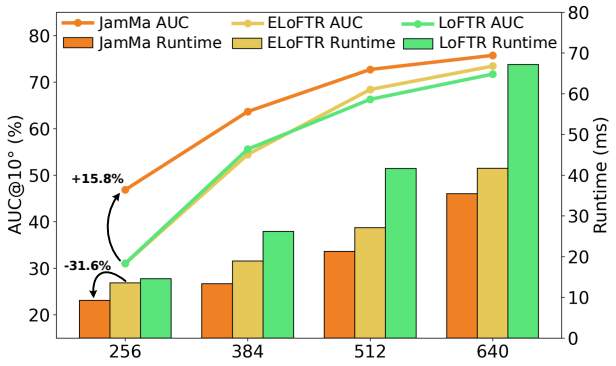

Figure S6. **Comparison in Low-Resolution Images.**

Method	Time	Pose est. AUC		
	(ms)	@5°	@10°	@20°
JamMa	61.8	**64.5**	**77.3**	**86.3**
(1) w/o sub-pixel ref.	60.2	62.1	75.5	84.7
(2) w/ C2F of LoFTR	**54.4**	61.8	74.9	84.4

Table S4. **Ablation Study on Coarse-to-Fine Module.** The ref. and C2F denote refinement and coarse-to-fine module, respectively.

C.6. Ablation Study on Coarse-to-Fine Module

We evaluate the coarse-to-fine module in LoFTR and the coarse-to-fine module in XoFTR without sub-pixel refinement. As shown in Tab. S4(1), sub-pixel refinement supervised by epipolar distance enhances performance by allowing regression-based matching points to achieve sub-pixel accuracy. Although the coarse-to-fine module in LoFTR is faster, its performance is hindered by two key limitations: 1) one-to-one coarse matching struggles in scenes with significant scale variations, and 2) its fine matching does not adjust matching points in the source image.

C.7. More Qualitative Comparisons

We provide additional qualitative comparisons of JamMa with LightGlue and ELoFTR in Fig. S9. JamMa consistently delivers robust matching results with shorter runtime, achieving lower pose estimation errors. Further qualitative comparisons for indoor and outdoor scenes are shown in

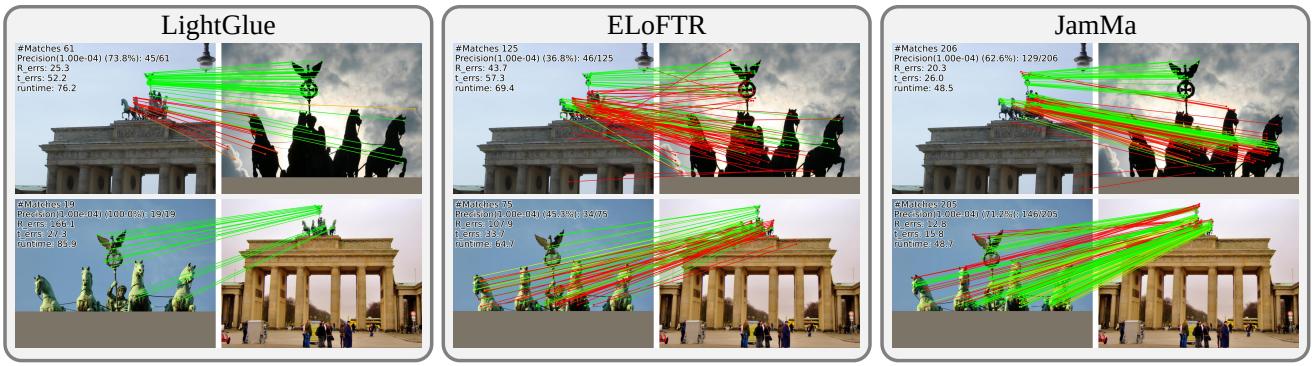

Figure S7. **Challenging Scenes.**

Figure S8. **Failure Cases.**

Fig. S10 and Fig. S11, with matched points color-coded for clarity.

C.8. Challenging Scenes.

Qualitative comparisons in challenging scenarios are presented in Fig. S7. All methods exhibit a significant reduction in the number of matches under drastic illumination and scale variations. Nevertheless, JamMa maintains a higher number of correct matches, resulting in more robust pose estimation.

C.9. Failure Cases.

Failure cases of JamMa are illustrated in Fig. S8. These cases typically arise in scenarios with extreme scale and viewpoint variations or in texture-less regions.

References

[1] H. Chen, Z. Luo, L. Zhou, Y. Tian, M. Zhen, T. Fang, D. McKinnon, Y. Tsin, and L. Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *Proceedings of the ECCV*, pages 20–36, 2022. 1, 3

[2] D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the CVPRW*, pages 224–236, 2018. 3

[3] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Dedode: Detect, don't describe—describe, don't detect for local feature matching. In *Proceedings of the 3DV*, pages 148–157, 2024. 3

[4] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. 1

[5] Z. Li and N. Snavely. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In *Proceedings of the CVPR*, pages 2041–2050, 2018. 1

[6] P. Lindenberger, P. Sarlin, and M. Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the ICCV*, pages 17627–17638, 2023. 3

[7] P. Sarlin, R. Cadena, C.and Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the CVPR*, pages 12716–12725, 2019. 3

[8] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. LoFTR: Detector-Free Local Feature Matching With Transformers. In *Proceedings of the CVPR*, pages 8922–8931, 2021. 1, 3

[9] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *Proceedings of the NeurIPS*, pages 24261–24272, 2021. 2

[10] Önder Tuzcuoğlu, Aybora Köksal, Buğra Sofu, Sinan Kalkan, and A Aydin Alatan. Xoftr: Cross-modal feature matching transformer. In *Proceedings of the CVPR*, pages 4275–4286, 2024. 3

[11] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *Proceedings of the ACCV*, pages 2746–2762, 2022. 1

[12] 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 CVPR*, pages 21666–21675, 2024. 1

[13] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision*, pages 821–844, 2021. 3

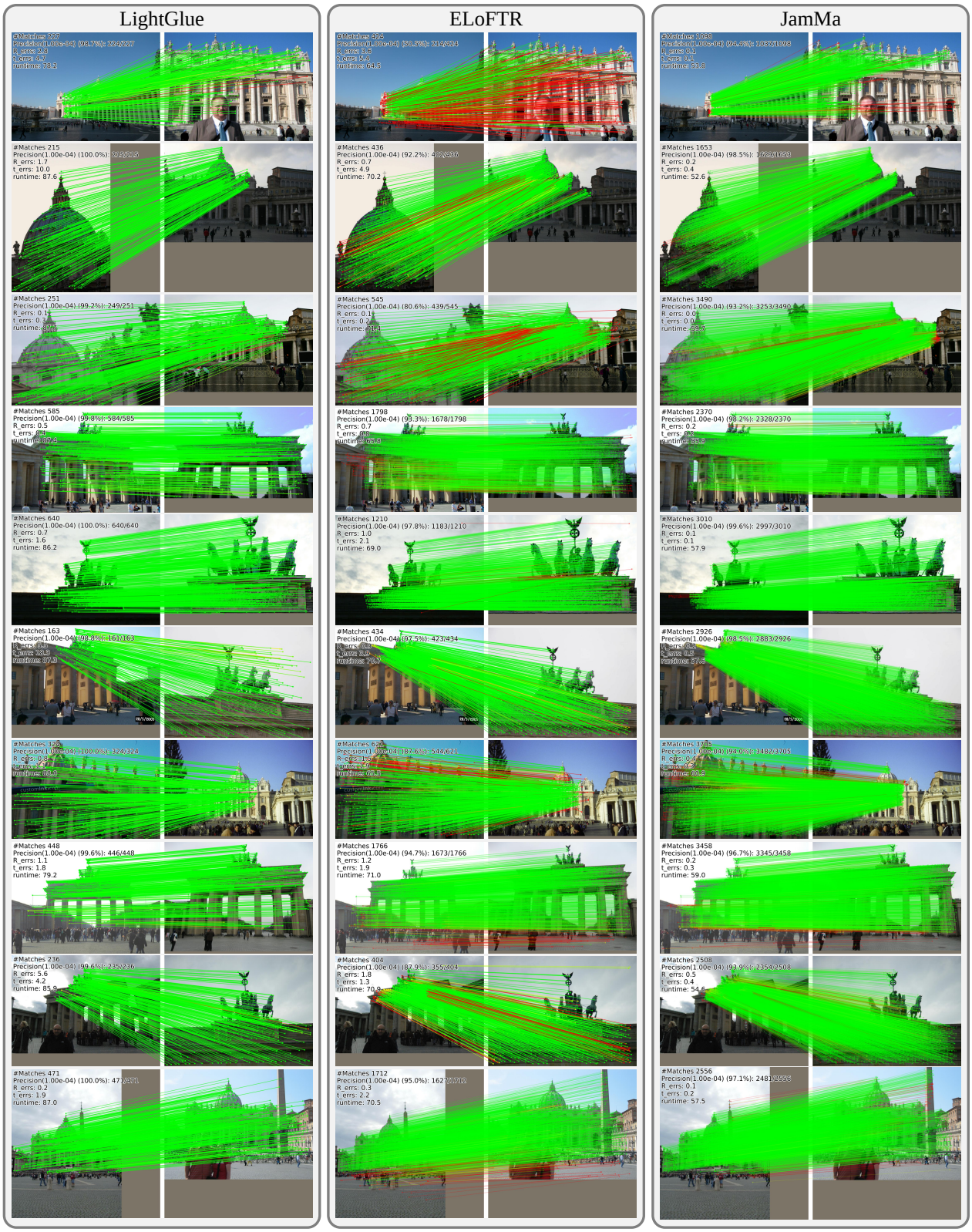

Figure S9. **Comparison of qualitative results.** The reported metrics include precision with an epipolar error threshold of 1×10^{-4}, rotation and translation errors in pose estimation, and runtime.

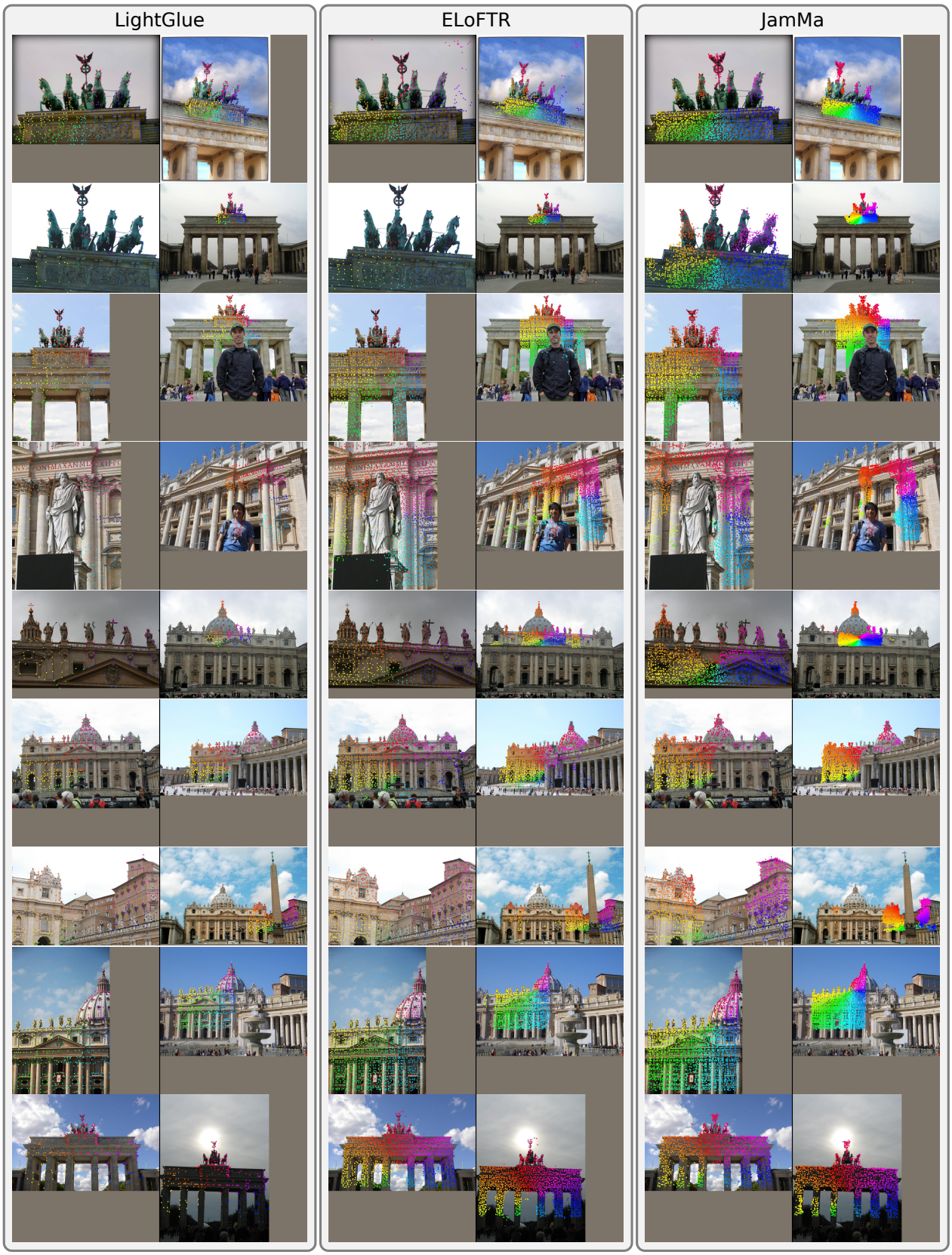

Figure S10. **Additional qualitative comparisons in outdoor scenes.** The matched points are visualized as the same color.

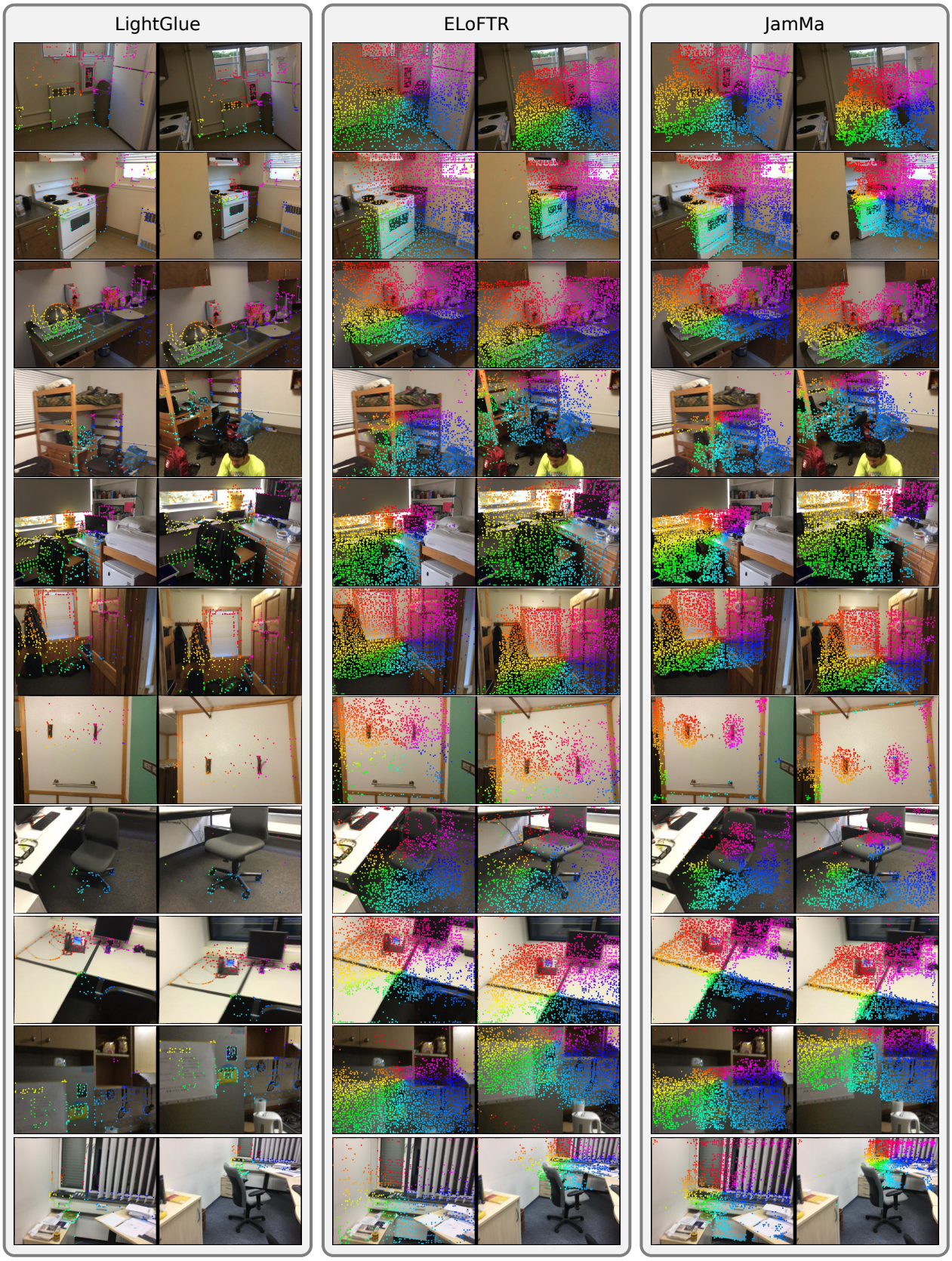

Figure S11. **Additional qualitative comparisons in indoor scenes.** The matched points are visualized as the same color.