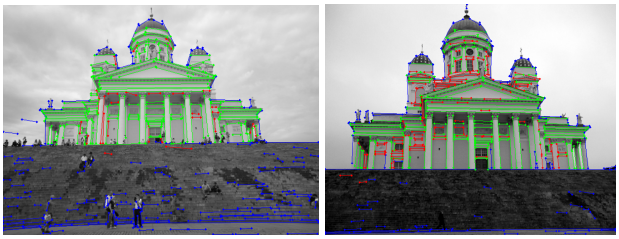
GlueStick: Robust Image Matching by Sticking Points and Lines Together Supplementary Material

In the following, we provide additional details regarding GlueStick, our point-line matcher. Appendix A offers a visualization of the generation of our line ground truth. Appendix B gives additional insights and ablation studies motivating our choices. Appendix C specifies some experimental details to reproduce our experiments and brings additional results. Appendix D shows matching results, as well as failure cases of our method. Finally, Appendix E provides visualizations of the attention for various kinds of nodes.

A. Ground Truth Generation

Designing a line matching ground truth (GT) is challenging, due to partial occlusions and lack of repeatability of line detectors. We provide here some visualizations of the GT generation process.

Fig. 1 shows an example of the ground truth between two images. Line segments can be either `MATCHED` (green), `UNMATCHED` (red), or `IGNORED` (blue). The latter case happens when the depth along the line is too uncertain or when its reprojection in the other image is occluded. The generation process is also illustrated in Fig. 2 for one pair of line segments. The advantage of the proposed method is that it recovers a large number of matches for each pair of images, providing a strong matching signal. In comparison to the method proposed in [1] which does robust 3D reconstruction of line segments, our method is faster and simpler. By avoiding the 3D line reconstruction step, we can train in larger scenes with potentially noisy depth, like the MegaDepth dataset [10].



(a) GT line assignations, shown as `MATCHED`, `UNMATCHED`, and `IGNORED`.

(b) Each color identifies a match $(i, j) \in \mathcal{M}^l$ of the GT.

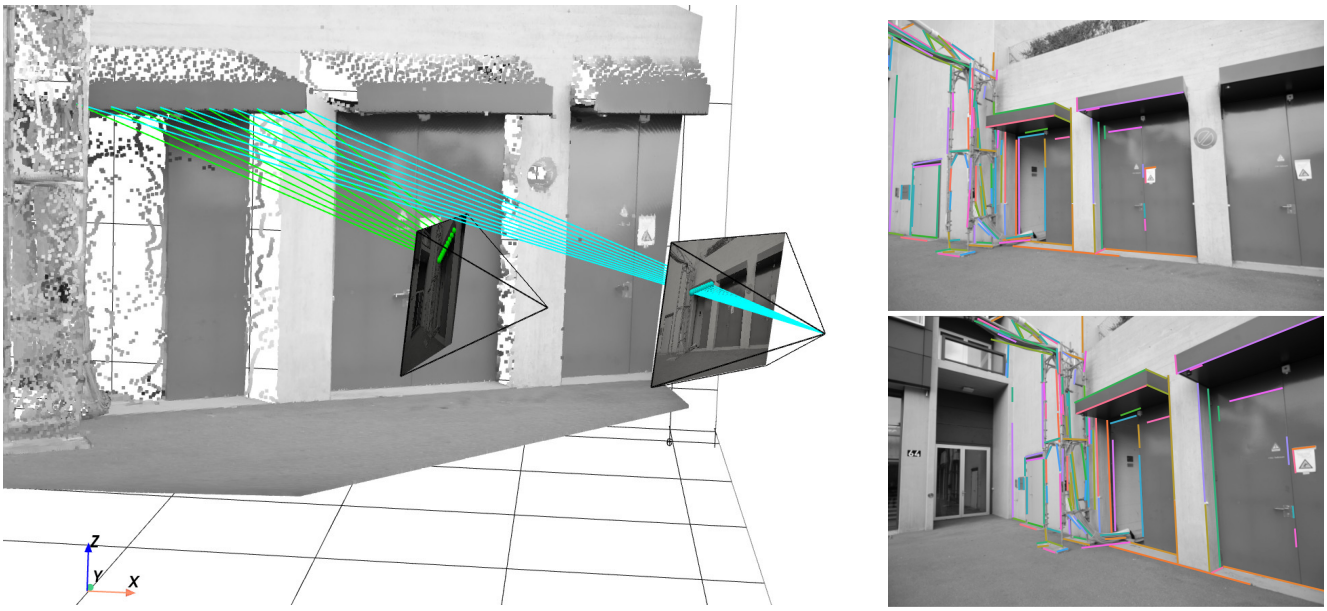
Figure 1: **Ground truth (GT) line assignations.** Examples of GT line matches. Note that blue lines are located in uncertain regions and depth discontinuities, and they are ignored during training.

B. Additional Insights on GlueStick

In this section, we give extra insights and motivations for our design choices.

Choice of the Line Segment Detector. In all our training and experiments, we used the Line Segment Detector (LSD) [7] to extract line segments. For a certain application, such as indoor wireframe parsing [8], learned methods largely overtake classic ones [5, 20, 23, 24]. However, learned methods struggle to generalize this power to other contexts, tasks, or types of images. For this reason, we have chosen LSD as the generic method to train GlueStick. Furthermore, we believe that our line pre-processing, turning an unordered set of lines into a connected graph, is beneficial to make the endpoints more repeatable across views, thus potentially making LSD more repeatable.

We ran a small experiment to compute the line repeatability of different line detectors on the HPatches [2] and ETH3D [18] datasets. We define line repeatability for a pair of images as the percentage between the number of line correspondences and the number of lines in the pair of images [11]. We establish line correspondences between images with the protocol defined in Sec. 3.4 of the main paper and Appendix A. Tab. 1 shows that the learned baseline F-Clip [5] obtains the highest repeatability, but detects few lines, due to the fact that it was trained on the ground truth lines of the Wireframe dataset [8]. On the contrary, LSD provides the best trade-off in terms of repeatability and



(a) 3D Visualization of a scene in ETH3D [18], with the depth associated with each point and the camera poses.

(b) GT Matching result

Figure 2: **Visualization of the ground truth (GT) generation.** To check if a pair of line segments $\mathbf{l}_i^A$ and $\mathbf{l}_j^B$ correspond to each other, we sample points along the segment: cyan points in the right image of (a). Points are lifted using depth and re-projected in the second image: green points in the left image of (a). We use the number of points that lie close to the segment in the second image to build each entry $\mathbf{C}_{i,j}^B$ of the cost matrix. Together with the reciprocal matrix $\mathbf{C}^A$ we define an assignation problem whose solution is our GT shown in (b).

	HPatches		ETH3D	
	#Lines	Rep. (%)	#Lines	Rep. (%)
LSD [7]	307	68.72	578	47.49
ELSED [20]	243	64.66	464	48.03
HAWP [24]	366	60.86	420	38.69
SOLD^2 [13]	167	65.49	332	39.86
F-Clip [5]	139	72.91	444	50.90
LETR [23]	95	70.65	311	47.70

Table 1: **Line segment detection comparison.** We compare different line segment detectors in terms of their repeatability in the HPatches [2] and ETH3D [18] datasets.

number of lines. Thus, this good trade-off, as well as its low localization error and versatility, make LSD a very suitable choice for our approach.

We provide in Fig. 4 additional visualizations of line segments for two traditional methods: LSD [7] and ELSED [20], and two learned methods: SOLD^2 [13] and HAWP [24]. While traditional ones sometimes detect noisy lines (for example in the sky), learned ones are often biased towards their training set and do not generalize very well to different settings, such as outdoor images.

However, is important for GlueStick to generalize and perform well with other line segment detectors. In Fig. 3 we run GlueStick using either LSD or SOLD^2 [13] lines, and we evaluate the precision-recall of both methods on the ETH3D dataset [18]. The latter are generic lines extracted by a deep network, with strong repeatability and low localization error [13]. It can be seen from the precision-recall curves that 1) our GlueStick model trained on LSD lines is able to generalize to other lines such as SOLD^2 [13], and 2) the performance is slightly better with LSD lines. This is reasonable, since GlueStick was already trained on these lines. In summary, we chose LSD as base detector for downstream tasks since it remains one of the most accurate detector currently available, by directly relying on the image gradient at a sub-pixel level.

Effect of the Fine-tuning. Again in Fig. 3, we compare our final GlueStick model with its pre-trained version on homographies, *GlueStick - H*. The plot shows that pre-training on homographies is already sufficient to get very high performance on ETH3D - better than the previous state-of-the-art line matchers. Fine-tuning on MegaDepth [10] with real viewpoint changes can however further improve the robustness of our matcher, as demonstrated by the stronger performance of the final model.

Dependence on Point Matches. When jointly matching two kinds of features, one caveat is often that one type of

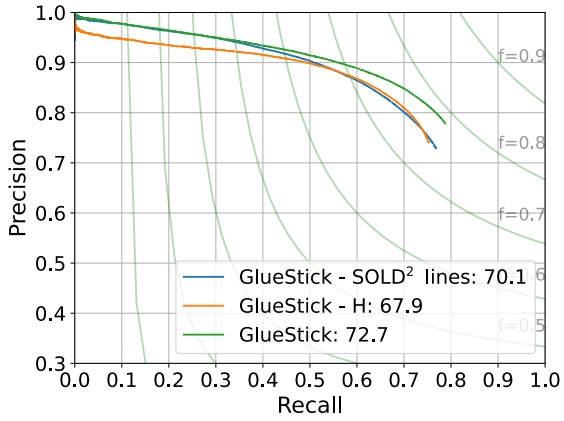


Figure 3: **Additional ablation study on the ETH3D dataset [18].** We report the precision-recall curve of the line matching, as well as Average Precision (AP) in the legend. Our final *GlueStick* model running with LSD [7] lines is compared to its pre-trained version on homographies (*GlueStick - H*), and the final model using SOLD^2 lines [13] (*GlueStick - SOLD^2 lines*).

feature takes the lead and the other relies mainly on the first one. While we know from SuperGlue [17] that point-only matching is already very strong on its own, we show here that our architecture is very robust to the absence of keypoints and that line-only matching is still possible. We ran an evaluation of the precision, recall, and average precision (AP) of the line matching on 1000 validation images of our homography dataset (images taken from the 1M distractor images of [14]), and tested different maximum numbers of keypoints per image. The results showed in Fig. 5, highlight that our line matching is extremely robust to the lack of keypoints. The precision remains indeed constant, and the recall and AP are decreasing by at most 5% when switching from 1000 keypoints to no keypoints. Thus, this study confirms that our matcher can be used in texture-less areas where no keypoints are present, and is still able to match lines with high accuracy.

Impact of the Line Length. While we adopted a line representation based on the endpoints, one may wonder whether GlueStick can handle very long lines, and how it performs with respect to the line length. We studied this on the ETH3D [18] by categorising lines into three categories of length (in pixels): *Short* ($[0, 50)$), *Medium* ($[50, 150)$), and *Long* ($[150, +\infty)$). The results are shown in Fig. 6. It can be seen that the best performance is obtained for long lines, showing that GlueStick is still able to match lines even without context in the middle of the line. This result is due to the fact that long lines are more stable across views, while short ones are often noisy and not very repeatable.

Robustness to Small Image Overlaps. The image overlap and scale changes between images can play a large role in matching. To study the effect of image overlap on GlueStick, we revisited our line matching experiment on the ETH3D dataset [18] and separated the pairs of images into three categories of image overlap: *Small* ($[0, 0.33)$), *Medium* ($[0.33, 0.66)$), and *Large* ($[0.66, 1]$). Overlap is defined as the proportion of pixels falling into the other image after reprojection. It is computed symmetrically between the two images, and the minimum of the two values is kept. Results are available in Fig. 7. While the performance naturally decreases with smaller overlaps, GlueStick maintains a strong performance on such hard cases.

C. Experimental Details

In this section, we first provide additional baselines to the experiment on ScanNet for homography and relative pose estimation. Secondly, we give details and visualizations of the pure rotation estimation between two image pairs, and its application to image stitching. Thirdly, we provide a comparison of methods on homography estimation on the HPatches dataset [2]. Finally, we give the full results of visual localization on the 7Scenes dataset [19], though the performance on most scenes is already saturated.

C.1. Relative Pose through Homographies

The experiment in Tab. 1 of the main paper was meant to evaluate the quality of homographies retrieved from points, lines or points+lines features. Homographies were evaluated by decomposing into the corresponding relative pose and evaluating the latter on the ScanNet dataset [4]. For the sake of completeness, we also report here the results that would be obtained with a more efficient method to obtain the relative pose: the 5-point algorithm to obtain the essential matrix [12], later decomposed as a relative pose. Tab. 2 demonstrates that the quality of relative poses retrieved through essential matrices is much higher than with homographies - as could be expected. GlueStick remains nevertheless the top performing method among all baselines. Note that we use here the outdoor models for all methods, for fairness reasons as GlueStick was only trained on outdoor data.

C.2. Pure Rotation Estimation

In this section, we describe the details of the pure rotation algorithm used to perform the experiment of Sec. 4.4.2 of the main paper. Inside a Hybrid RANSAC [3], we design minimal and least square solvers to estimate the rotation based on points, lines, or a combination of both.

Images are cropped from the panorama images of SUN360 [22] and projected with a calibration matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$. Fig. 8 shows some examples. The relation between

(a) LSD [7] (b) ELSED [20] (c) SOLD^2[13] (d) HAWP [24]

Figure 4: **Comparison of line segment detectors.** Learned methods such as SOLD^2 [13] and HAWP [24] may not generalize well in all situations, such as outdoors. Traditional ones such as LSD [7] and ELSED [20] produce a lot of overlapping segments and sometimes noisy ones. We decided to use LSD for its high versatility and accuracy.

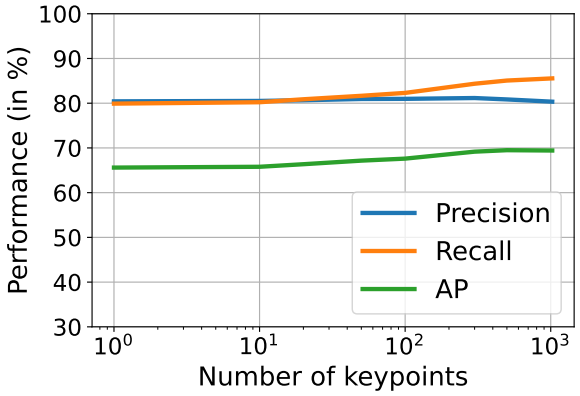


Figure 5: **Analysis of the dependence on keypoints.** We run GlueStick on 1000 image pairs warped by a homography (taken from the 1M distractor images of [14]), with varying numbers of keypoints, and report the precision, recall and Average Precision (AP) of the line matching. GlueStick robustly matches lines even when few or no keypoints are present.

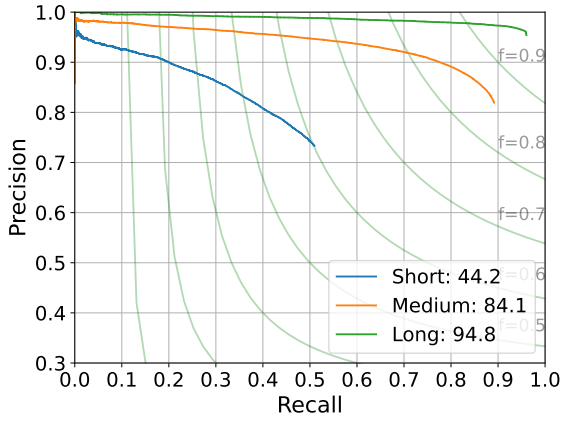


Figure 6: **Analysis of the impact of line length.** We run GlueStick on the ETH3D dataset [18] and evaluate separately the matching of Short, Medium, and Long lines. The best performance is obtained for long lines, as they are more stable than short ones. GlueStick can thus match long lines even with an endpoint representation.

both images is defined by:

$$\mathbf{x}^B = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{x}^A, \tag{1}$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix between the cameras. Thus, we can calibrate the features detected on each image, multiplying them by $\mathbf{K}^{-1}$. This way, we only have to robustly estimate the 3 Degrees-of-Freedom (DoF) of the rotation.

Point features are sampled uniformly, and lines are sampled proportionally to the square root of their length to give priority to larger lines. The probability of choosing one type of feature is proportional to its number of matches. For example, if a pair has 60 point matches and 40 line matches, the probability of choosing a point is 60% and 40% for lines.

The minimal solver randomly chooses 2 feature matches (point-point, point-line, or line-line) and estimates the rotation based on them. This can be seen as aligning 2 sets of 3D vectors. Homogeneous points are already 3D vectors going from the camera center to the plane $Z = 1$. To get a vector from a line segment with homogeneous endpoints $\mathbf{x}_s$ and $\mathbf{x}_e$ we use its line-plane normal $\mathbf{n} = \mathbf{x}_s \times \mathbf{x}_e$. To make the method invariant to the order of the endpoints, we force the normals of the segments to have a positive dot product.

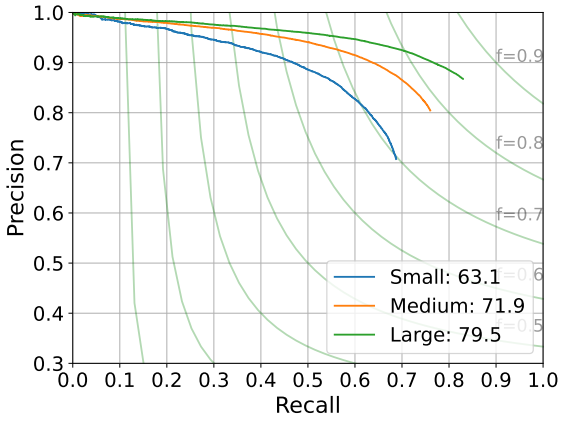


Figure 7: **Analysis of the impact of the image overlap.** We run GlueStick on the ETH3D dataset [18] and classify image pairs into three categories: Small, Medium, and Large overlap. While the performance of GlueStick decreases with smaller overlaps, it is able to maintain a high performance for all kinds of overlaps.

		Pose error (↓)	Pose AUC (↑)
Pose from H	SuperGlue (SG) [17]	18.1	15.6 / 29.8 / 39.4
	LoFTR [21]	16.8	15.8 / 30.9 / 41.4
	GlueStick	**14.1**	**19.3 / 35.4 / 46.0**
Pose from E	SuperGlue (SG) [17]	8.6	30.5 / 46.0 / 54.1
	LoFTR [21]	11.7	23.6 / 39.6 / 48.4
	GlueStick	**8.4**	**30.9 / 46.8 / 55.1**

Table 2: **Using essential matrices instead of homographies on ScanNet [4].** While our experiment on ScanNet is meant to evaluate homographies, we display here the results that would be obtained when using essential matrices to get a relative pose, instead of homographies. We report the median pose error in degrees, as well as the AUC at 10° / 20° / 30° error. Essential matrices are naturally more robust and obtain better results when evaluated on relative pose estimation.

	Rotation error (↓)		AUC (↑)					
	Avg	Med	0.25º	0.5º	1º	2º	5º	10º
LineTR [25]	60.37	10.80	0.239	0.307	0.366	0.414	0.455	0.474
LBD [26]	19.57	0.054	0.593	0.681	0.736	0.768	0.790	0.799
SOLD^2 [13]	23.74	0.308	0.232	0.384	0.515	0.609	0.682	0.713
L2D2 [1]	18.31	0.056	0.578	0.667	0.725	0.765	0.795	0.808
GlueStick-L	8.79	0.070	0.579	0.701	0.780	0.830	0.869	0.885
SG [17]	**0.135**	0.052	0.730	0.860	0.929	0.964	0.985	**0.992**
GlueStick-P	0.230	0.052	0.729	0.860	0.928	0.963	0.984	0.991
PL-Loc [25]	0.338	0.050	0.733	0.859	0.927	0.961	0.982	0.989
GlueStick-PL	0.327	**0.039**	**0.789**	**0.890**	**0.943**	**0.970**	**0.986**	0.991

Table 3: **Pure rotation estimation on SUN360 [22].** We estimate a rotation based on point-only, line-only, or points+lines matches. We report the average and median rotation error in degrees, as well as the Area Under the Curve (AUC) at 0.25 / 0.5 / 1 / 2 / 5 / 10 degrees error.

This simple heuristic works as long as the sought rotation is less than 180°. Therefore, we can obtain a 3D vector from any of the types of features. For two (or more) correspondences, the optimal rotation can then be found with SVD using the Kabsch algorithm [9].

We provide a more extensive table of results than in the main paper in Tab. 3, as well as visual examples of the point and line matches obtained by GlueStick, and the resulting image stitching output in Fig. 8.

C.3. Homography Estimation in HPatches

HPatches [2] is one of the most frequently used datasets to evaluate image matching. It contains 108 sequences where the scene contains only one dominant plane, with 6 images per sequence. Each sequence has either illumination of viewpoint changes. Similarly as in [17], we compute a homography from point and/or line correspondences and RANSAC, and compute the Area Under the Curve (AUC) of the reprojection error of the four image corners. We report the results for thresholds 3 / 5 / 10 pixels. We also compute the precision and recall of the ground truth (GT) matches obtained by the GT homography.

We report the results in Tab. 4 and Fig. 9. HPatches is clearly saturated and the precision/recall metrics are already very high for point-based methods. Note the strong performance of GlueStick on line matching, with an increase by nearly 10% in precision compared to the previous state of the art. Regarding point-based methods and homography scores, GlueStick obtains a very similar performance as the previous point matchers SuperGlue [17] and LoFTR [21], and ranks first (with a very small margin) in terms of homography estimation. In addition to the fact that HPatches is saturated, it contains also very few structural lines that could have been useful to refine the homography fitting. Thus, the improvement brought by line segments is not significant here.

		AUC (↑)			Points (↑)		Lines (↑)	
		3px	5px	10px	P	R	P	R
L	L2D2 [1]	43.73	55.98	69.39	-	-	55.55	38.76
	SOLD^2 [13]	26.60	36.41	48.82	-	-	80.57	77.43
	LineTR [25]	42.90	55.74	69.26	-	-	78.78	58.74
	LBD [26]	46.82	59.13	71.82	-	-	82.73	56.38
	GlueStick-L	46.61	61.45	76.32	-	-	**90.27**	76.69
P	SuperGlue [17]	66.21	77.77	88.05	**98.85**	97.44	-	-
	LoFTR [21]	66.15	75.28	84.54	97.60	**99.38**	-	-
	GlueStick-P	65.88	77.41	87.72	**98.85**	97.08	-	-
P+L	PL-Loc [25]	60.03	71.44	83.08	90.80	77.60	80.33	50.35
	GlueStick-PL	**66.88**	**78.14**	**88.12**	98.00	94.86	89.54	**80.44**

Table 4: **Homography estimation in HPatches [2].** We report the Area Under the Curve (AUC) of the cumulative error curve generated by the re-projection error of the four image corners at different thresholds (3px, 5px, 10px), as well as the precision (P) and recall (R) of the matches.

C.4. Visual Localization on the Full 7Scenes Dataset

As stated in the main paper, 7Scenes [19] is a rather small-scale dataset for visual localization, which is already largely saturated for point-based methods. Adding line segments into the pipeline can improve the results only in a few scenes such as Fire, Office and mostly on Stairs. We demonstrate this in Tab. 5, where we used the same setup as described in the main paper. While all methods obtain close results on such a saturated dataset, GlueStick is slightly ahead of the baselines, and largely outperforms them on the most challenging scene, Stairs.

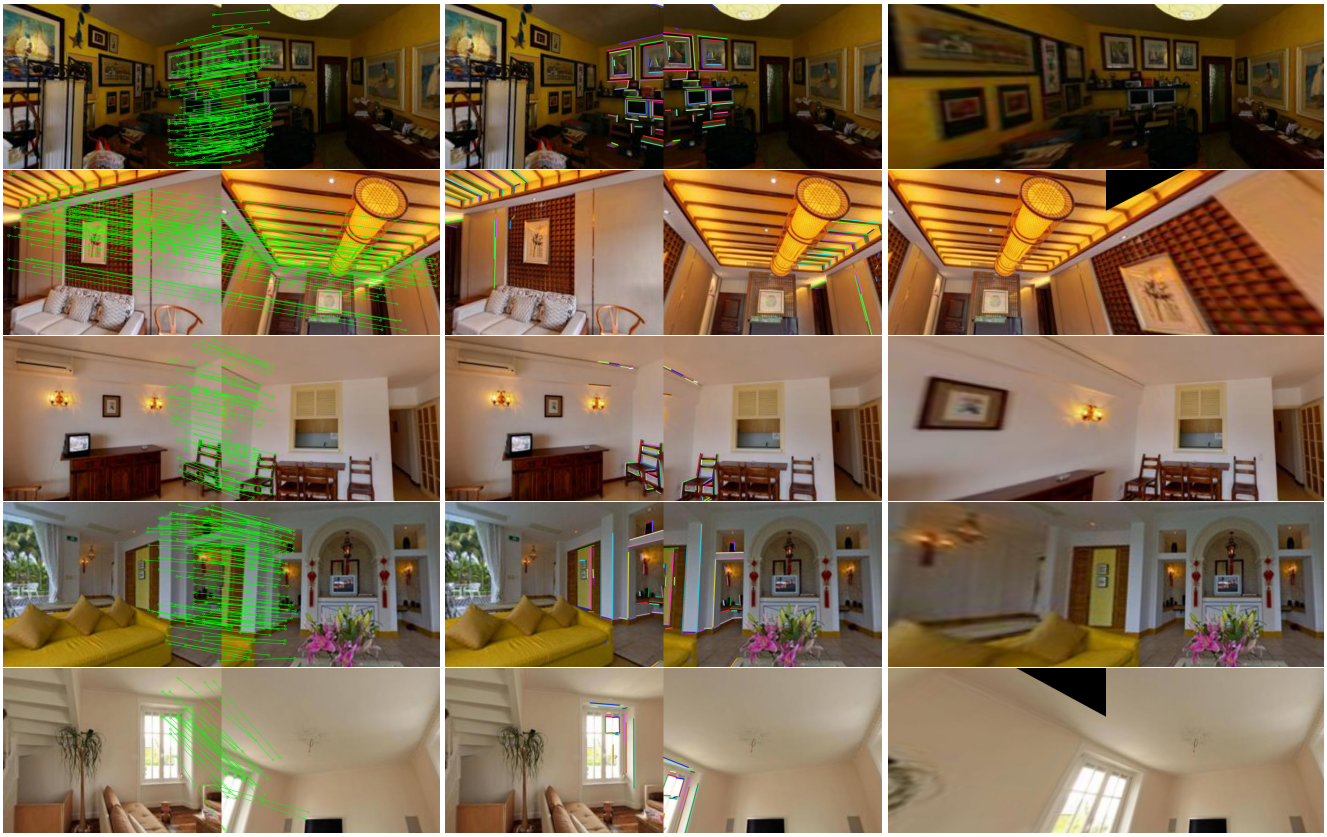
D. Qualitative Examples

D.1. Feature Matches

Fig. 10 displays some examples of line matching on the ETH3D dataset [18]. We plot in green the correct matches and in red the incorrect ones. Thanks to its spatial reasoning in the GNN and context-awareness, GlueStick is consistently matching more lines and with a higher precision than previous works. This is in particular true for scenes with repeated structures, such as the one in the right column, where the descriptors of SOLD^2[13] and L2D2[1] do not have context from neighboring lines, and can only match a few lines.

D.2. Visualization of the Camera Pose Estimation

We visualize the reprojection of points and lines on the scene Stairs of the 7Scenes dataset [19] in Fig. 11. We plot in green the points and lines that were originally detected in 2D, and re-project in red the corresponding 3D features using the estimated camera pose. The reprojections of GlueStick are almost perfectly aligned compared to the ones of hloc [16, 15], highlighting the quality of the poses retrieved



(a) Point matches (b) Line matches (c) Image stitching results

Figure 8: **Examples of GlueStick matches on image pairs of SUN360 [22].** We provide the point and line matches, as well as the stitching of the two images using the resulting matches.

	Points			Points + Lines				
	SuperGlue [17]	LoFTR [21]	GlueStick - P	SOLD^2 [13]	LineTR [25]	L2D2 [1]	SG + Endpts	GlueStick - PL
Chess	**2.4** / 0.81 / 94.5	2.5 / 0.86 / 93.8	**2.4** / **0.80** / 94.3	**2.4** / 0.82 / 94.4	**2.4** / 0.81 / 94.5	**2.4** / 0.83 / 94.5	**2.4** / 0.82 / **94.6**	**2.4** / 0.82 / 94.5
Fire	1.9 / 0.76 / 96.4	1.7 / **0.66** / 96.8	2.0 / 0.78 / 96.6	**1.6** / 0.69 / 96.8	**1.6** / 0.69 / 97.0	**1.6** / 0.69 / 96.4	1.7 / 0.69 / 97.2	1.7 / 0.69 / **97.4**
Heads	1.1 / 0.74 / 99.0	1.1 / 0.78 / 98.2	1.1 / 0.74 / 99.2	**1.0** / **0.72** / **99.4**	1.1 / 0.75 / 99.1	**1.0** / 0.73 / 99.3	1.1 / 0.74 / 99.2	**1.0** / 0.74 / **99.4**
Office	2.7 / 0.83 / 83.9	2.7 / 0.83 / 82.0	2.7 / 0.83 / 83.6	**2.6** / 0.80 / **84.7**	**2.6** / **0.79** / 84.4	**2.6** / 0.81 / 83.9	**2.6** / **0.79** / 84.4	**2.6** / **0.79** / 84.6
Pumpkin	4.0 / 1.05 / 62.0	**3.9** / 1.12 / **62.4**	**3.9** / **1.04** / 62.2	4.0 / 1.07 / 60.2	4.0 / 1.08 / 61.5	4.0 / 1.05 / 61.3	4.0 / 1.06 / 61.2	4.0 / 1.06 / 61.5
Red kitchen	3.3 / **1.12** / 72.5	3.3 / 1.14 / **73.8**	3.3 / **1.12** / 72.8	**3.2** / 1.15 / 72.6	3.3 / 1.15 / 72.6	**3.2** / 1.14 / 72.9	**3.2** / 1.14 / 72.8	**3.2** / 1.13 / 73.0
Stairs	4.7 / 1.25 / 53.4	4.4 / 0.95 / 53.9	4.4 / 1.21 / 55.4	3.2 / 0.83 / 75.8	3.7 / 1.02 / 66.6	4.1 / 1.15 / 55.8	3.1 / 0.81 / 75.6	**2.9** / **0.79** / **79.7**
Total	2.9 / 0.94 / 80.2	2.8 / 0.91 / 80.1	2.8 / 0.93 / 80.6	2.6 / 0.87 / 83.4	2.7 / 0.90 / 82.2	2.7 / 0.91 / 80.6	2.6 / **0.86** / 83.6	**2.5** / **0.86** / **84.3**

Table 5: **Visual localization on the full 7Scenes dataset [19].** We report the median translation error (cm) / median rotation error (deg) / pose AUC at a 5 cm / 5 deg threshold. Most scenes are already saturated for point methods, and lines can hardly make a difference.

by our method.

D.3. Failure Cases and Limitations

While jointly matching keypoints and lines in the same matching network helps disambiguating many challenging scenarios, GlueStick may still underperform in some scenarios. We list in the following some limitations of our method, and report some failure cases.

Limitations. Currently, the main performance bottleneck of GlueStick lies in the line segment detection. While this field has seen great advances in recent years, existing line detectors are still not as repeatable and accurate as point features, making the line matching more challenging. Partially occluded lines are also a potential issue for GlueStick, as it represents lines with their two endpoints. However, we observed a surprisingly good robustness of GlueStick to

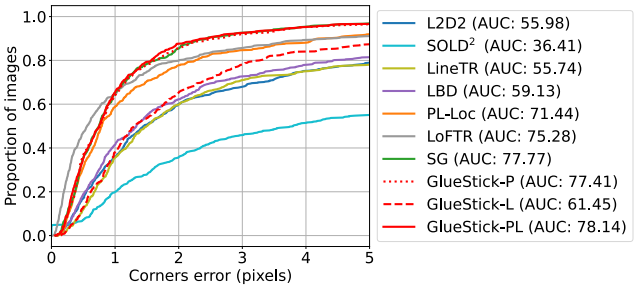


Figure 9: **HPatches [2] cumulative error curve.** We report the percentage of images where the homography is correctly predicted for various pixel error thresholds.

partially occluded lines, probably thanks to the neighboring points and lines that are not occluded. Note that it is also possible to equip GlueStick with a similar mechanism as in SOLD^2 [13], by sampling several points along the line segments, and matching them with the Needleman-Wunsch algorithm. We tried this option and observed a small increase in performance (notably in areas with occluded lines), but at the cost of higher running time. Therefore, we did not incorporate this feature in our final method.

Another issue is that points and lines are still detected with different methods for now. Thus, three networks / algorithms need to be run to detect and describe keypoints, detect lines, and finally match them. Jointly detecting and describing points and lines would be an interesting future direction of research. Furthermore, the extraction of discrete features such as points and lines is usually non-differentiable, such that one cannot get a fully differentiable pipeline going from the feature extraction to their matching. Enabling such end-to-end training could potentially make features better specialized for matching.

Finally, our current supervision requires ground truth correspondences of points and lines across images (usually obtained through reprojection with depth and camera poses). Other supervision signals such as epipolar constraints and using two-view geometry would be an interesting direction of improvement in the future.

Failure cases. We display in Fig. 12 a few examples of scenarios where GlueStick may still fail or underperform. First, in scenes with repeated patterns, GlueStick is able to find a consistent matching, but can be displaced by one pattern if there is no additional hint to disambiguate the transform between the two images. This is for example the case on 7Scenes Stairs [19], when the camera is only seeing several steps, and it is unclear which step should be matching with which one in the other image.

Secondly, GlueStick has not been trained for large rotations beyond $45°$ and often fails in these scenarios. The pretraining with homographies was done with rotations lower than $45°$, and real viewpoint changes are rarely with such rotations. Nonetheless, a simple fix is to rotate one of the two images by $0°, 90°, 180°$ and $270°$, match it with the other image, and keep the best matching among the four.

Finally, a challenging scenario happens when the images have low texture in combination with symmetric structures. The former makes visual descriptors less reliable, while the latter makes it harder to disambiguate matches from the spatial context. The performance is then degraded in such situations. Having access to sequential data and feature tracking may help solving such cases.

E. Attention visualization

We display the attention for some nodes in Fig. 13. This visualization is obtained by taking the attention matrix at various cross layers, averaging it across all heads, and taking the top 20 activated nodes. Green lines are used for nodes with connectivity greater than 0 (i.e. line endpoints), and cyan for nodes that are isolated keypoints. It can be seen from the left column that keypoint attention is leveraging the line structure to look for the right points along the line. In the right column, we can see that line endpoints can benefit from both keypoint and line endpoint attention. The attention is initially looking broadly at the image, before gradually focusing on the corresponding node in the other image. Thus, both points and line endpoints can complement each other to disambiguate the matching process.

References

[1] Hichem Abdellali, Robert Frohlich, Viktor Vilagos, and Zoltan Kato. L2D2: learnable line detector and descriptor. In *IEEE International Conference on 3D Vision (3DV)*, 2021. 1, 6, 7, 9

[2] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 1, 2, 3, 6, 8

[3] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid camera pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 3, 5

[5] Xili Dai, Haigang Gong, Shuai Wu, Xiaojun Yuan, and Ma Yi. Fully convolutional line parsing. *Neurocomputing*, 506:1–11, 2022. 1, 2

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 10

[7] Rafael Grompone von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 32(4):722–732, 2010. 1, 2, 3, 4

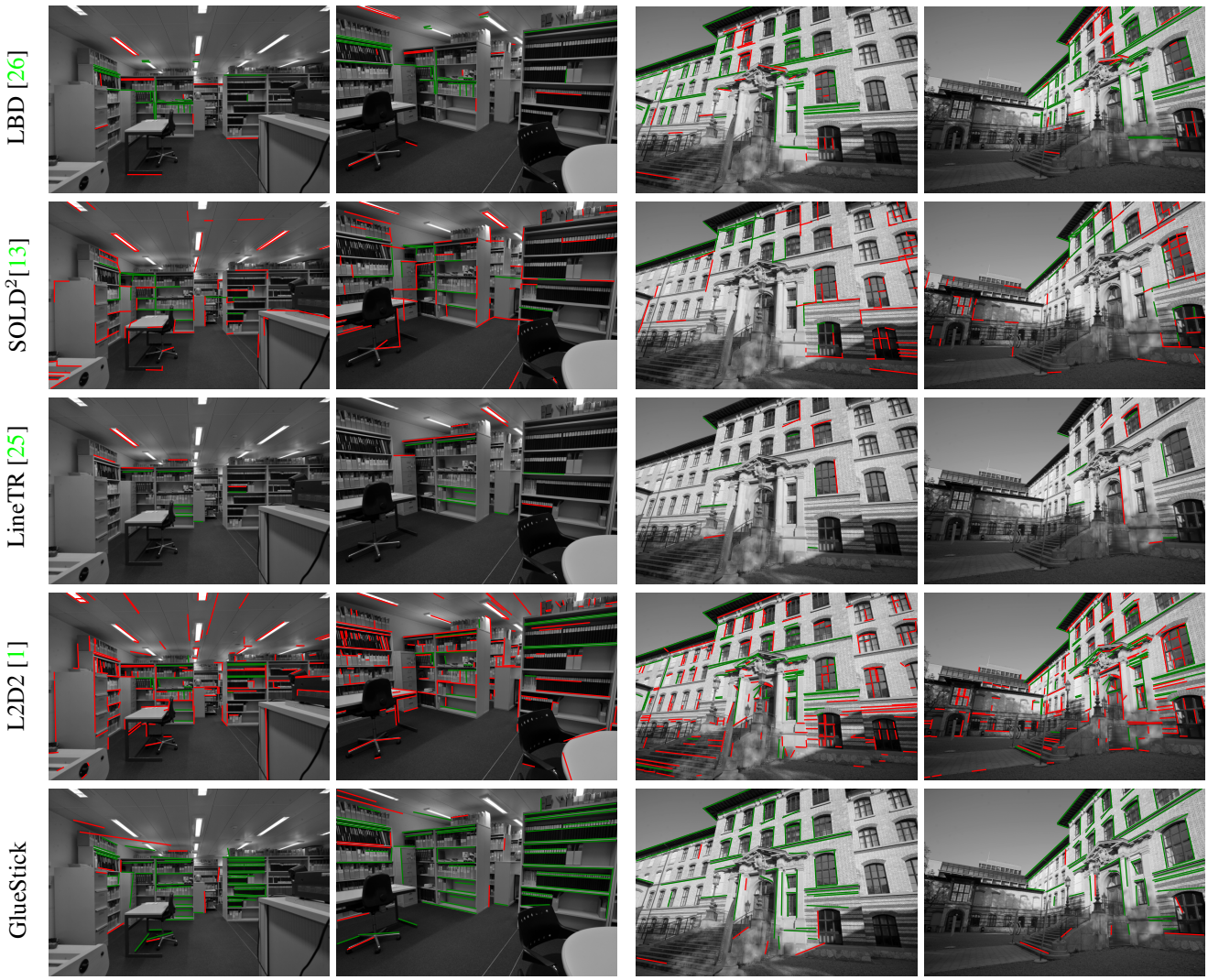


Figure 10: **Line matches examples on ETH3D [18].** We display correct line matches in green and incorrect ones in red for several state-of-the-art line matchers.

[8] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 1

[9] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 6

[10] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 1, 2

[11] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and L Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vis. (IJCV)*, 65:43–72, 2005. 1

[12] David Nistér. An efficient solution to the five-point relative pose problem. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2003. 3

[13] Rémi Pautrat, Juan-Ting Lin, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. SOLD^2: Self-supervised occlusion-aware line description and detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021. 2, 3, 4, 6, 7, 8, 9

[14] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3, 5

[15] Paul-Edouard Sarlin. Visual localization made easy with hloc. https://github.com/cvg/Hierarchical-Localization. 6, 10

[16] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchi-

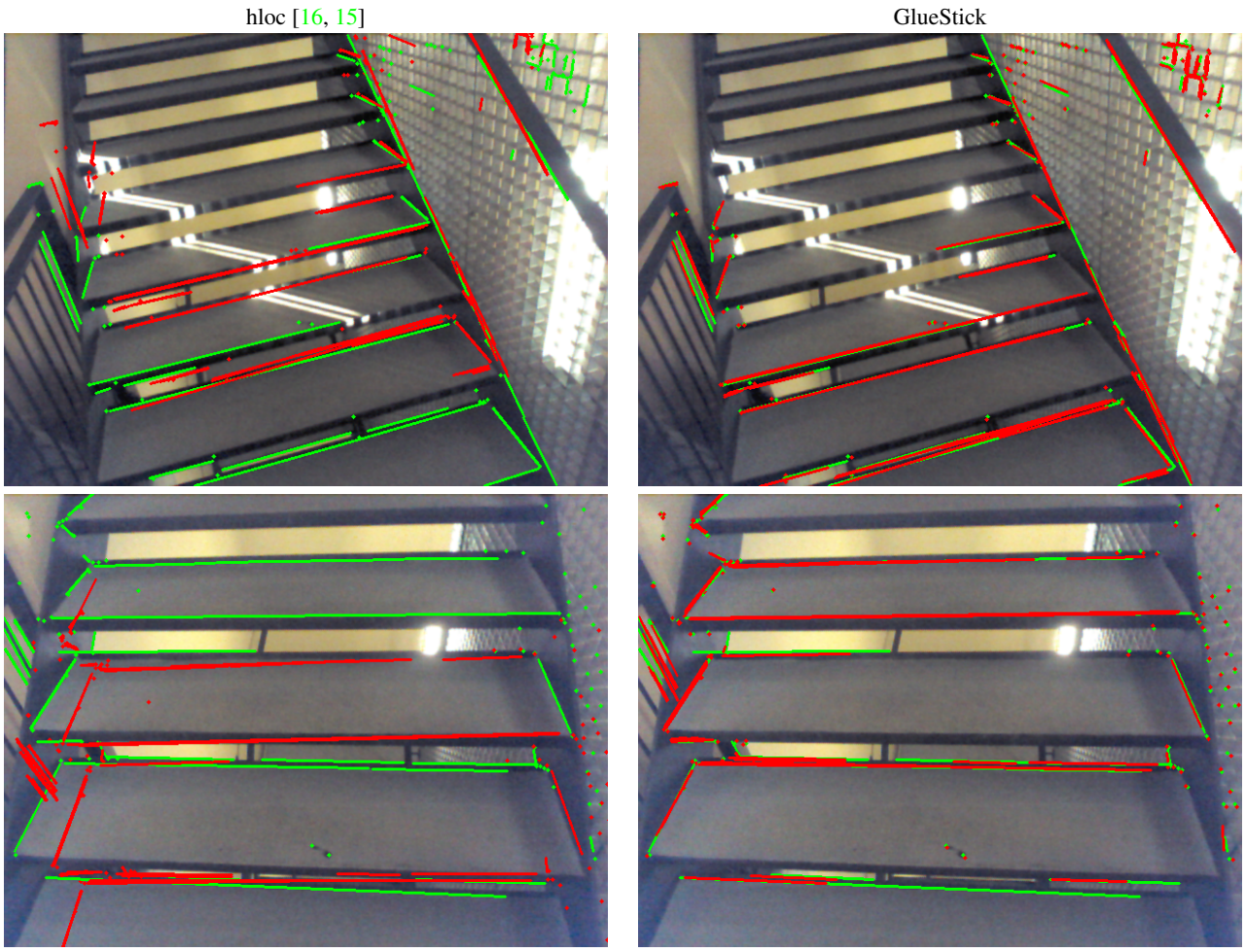


Figure 11: **Camera pose estimation visualizations.** We compare the originally detected keypoints and lines (in green) with the re-projected points and lines using the estimated camera pose (in red), for hloc [16, 15] with SuperPoint [6] + SuperGlue [17] and our method. The reprojections of GlueStick align almost perfectly with the 2D detections, showing a high quality estimated pose.

cal localization at large scale. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 6, 10

[17] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020. 3, 5, 6, 7, 10

[18] Thomas Schops, Johannes L. Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 1, 2, 3, 5, 6, 9

[19] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013. 3, 6, 7, 8

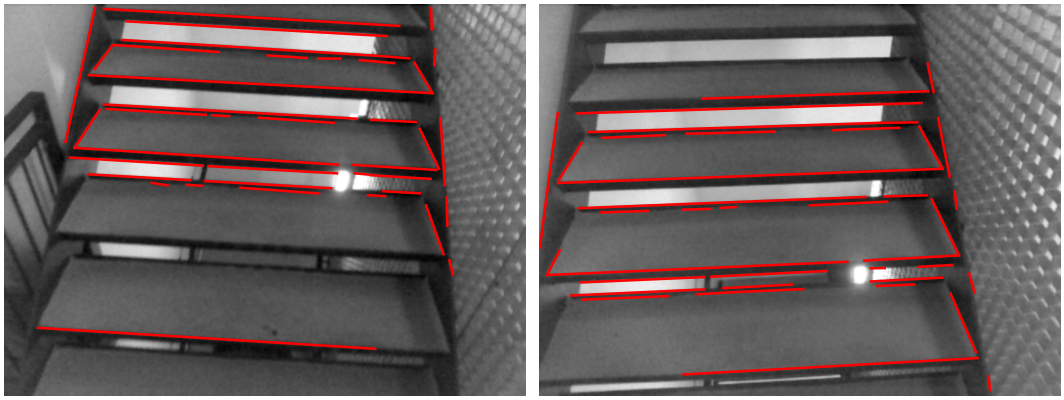
[20] Iago Suárez, José M Buenaposada, and Luis Baumela. ELSED: Enhanced line segment drawing. *Pattern Recognition*, 127:108619, 2022. 1, 2, 4

[21] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021. 5, 6, 7

[22] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012. 3, 6, 7

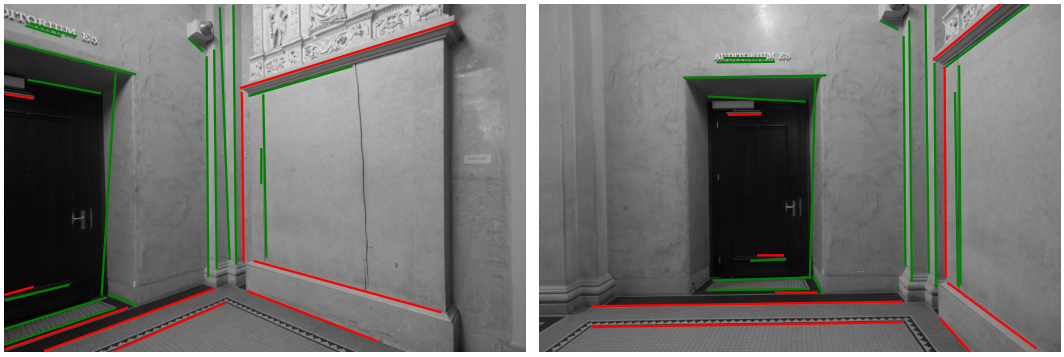
[23] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4257–4266, 2021. 1, 2

[24] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Gui-Song Xia, Liangpei Zhang, and Philip H.S. Torr. Holistically-attracted wireframe parsing. In *IEEE Conf. Comput. Vis.*



(a) Repeated patterns: GlueStick finds consistent line matches, but displaced by one step.

(b) GlueStick is not trained on large rotations.



(c) The lack of texture / symmetric structures make visual descriptors / spatial descriptors less reliable.

Figure 12: **Failure cases.** We display correct line matches in green and wrong ones in red. GlueStick may still fail or underperform in some situations, such as (a) perfectly repeated patterns that are hard to disambiguate, (b) large rotation (e.g. $> 45°$), and (c) lack of texture and symmetric structures.

Pattern Recog. (CVPR), 2020. 1, 2, 4 9

[25] Sungho Yoon and Ayoung Kim. Line as a visual sentence: Context-aware line descriptor for visual localization. *IEEE Robotics and Automation Letters (RAL)*, 6(4):8726–8733, 2021. 6, 7, 9

[26] Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7):794–805, 2013. 6,

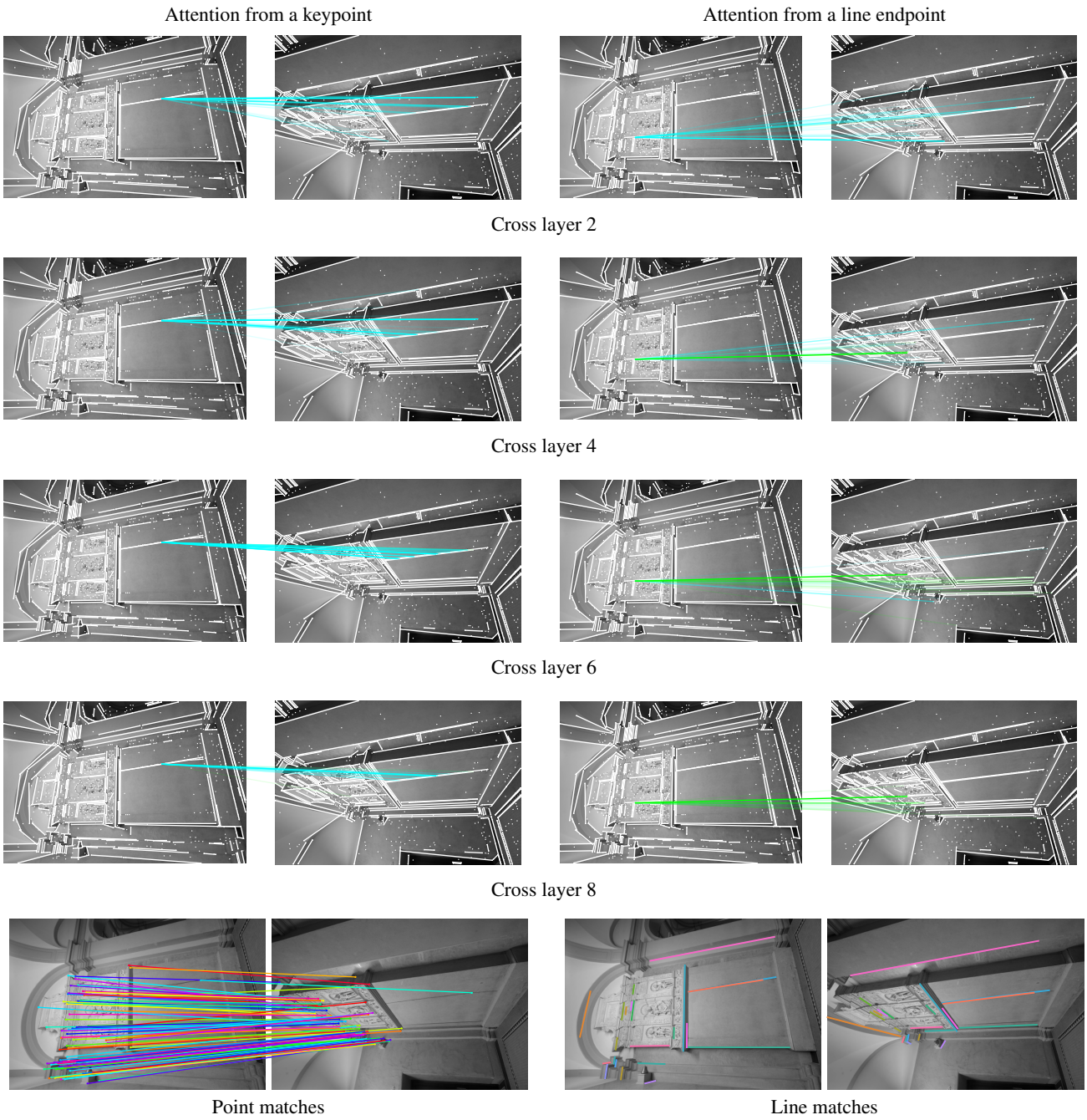


Figure 13: **Cross attention visualization.** We plot in the first column the attention from a keypoint and in the right column the attention of a line endpoint, for various layers in the Graph Neural Network. We compute here the average attention across all heads and keep the top 20 activated nodes. More opaque lines means higher attention, green matches are connected to a line endpoint in the second image, and cyan matches are connected to an isolated keypoint. The last row pictures the final matches.