Hierarchical Visual Relocalization with Nearest View Synthesis from Feature Gaussian Splatting

Supplementary Material

Contents

A. Details of Adaptive Viewpoint Retrieval

The algorithm of the adaptive viewpoint retrieval strategy is illustrated in Algorithm 1.

Hyperparameters. The hyperparameters used in this retrieval process for indoor and outdoor scenes are summarized in Table I. These include the number of retrieved images k_1 for geometric verification in the coarse stage, the number of virtual views k_2 generated by pose perturbation, the number of candidate images k_3 verified in the fine stage, the perturbation ranges $(a°, b\,\text{m})$ for rotation and translation, and the inlier threshold $\mathcal{I}$ used during geometric verification. Moreover, the images retrieved in the coarse stage exhibit strong viewpoint redundancy. To avoid repeated verification on nearly identical views, we perform geometric verification only on every 10th retrieved image.

Table I. Hyperparameters in the adaptive retrieval process.

Scene Type	k_1	k_2	k_3	a	b	$\mathcal{I}$
Indoor	10	150	5	5	0.5	150
Outdoor	10	100	5	5	0.8	300

Perturbation sampling strategies We compare multiple perturbation strategies generated using different sampling distributions on the stairs scene. As shown in Table II, Normal and Random perturbations consistently outperform Uniform sampling in both accuracy and robustness. Normal sampling yields the best localization success rate, whereas Random provides a favorable trade-off between accuracy and runtime.

Range of perturbations. In Fig. I, we vary the angular perturbation a on the Stairs scene. The accuracy improves as a increases from $0°$ to $5°$, indicating that a moderate amount of angular perturbation helps the system escape degenerate initializations and yields better coarse pose candidates. However, excessively large perturbations (e.g., $a = 10°$) degrade performance, as they place the perturbed poses too far from the true solution. Fig. II presents the complementary study in which the translation perturbation b is varied

Table II. Comparison of Uniform, Normal, and Random perturbation strategies. Time denotes the average runtime required for the initial localization.

Distributions	**Avg. Err [cm/°]** ↓	R**@[5cm, 5°]** ↑	**Time (s)** ↓
Uniform	1.11/0.33	85.3	0.60
Normal	**1.03/0.30**	**92.8**	0.60
Random	**1.03/0.30**	91.9	**0.57**

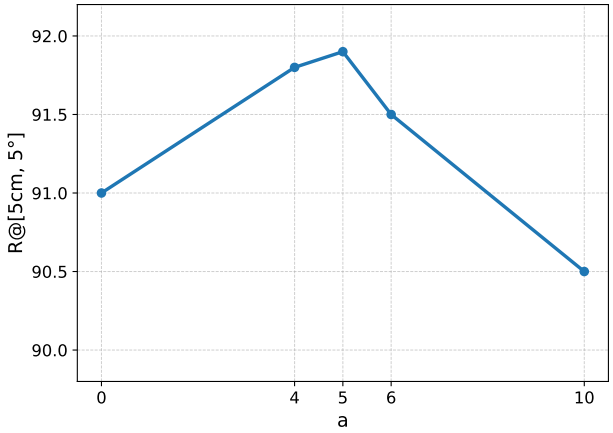

Figure I. The R@[5cm, 5°] accuracy under different perturbation angle values a, with the perturbation distance b fixed to 0.5 m.

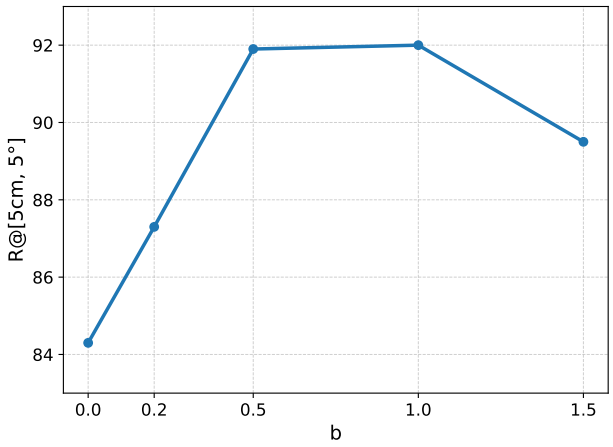

Figure II. The R@[5cm, 5°] accuracy under different perturbation distance values b, with the perturbation angle a fixed to 5°.

Table III. The number of images in the training and test sets for each scene.

7-Scenes	**Scenes**	***Chess***	***Fire***	***Heads***	***Office***	***Pumpkin***	***Kitchen***	***Stairs***	-	-	-	-	-
	Train	4000	2000	1000	6000	4000	7000	2000	-	-	-	-	-
	Test	2000	2000	1000	4000	2000	5000	1000	-	-	-	-	-
12-Scenes	**Scenes**	***Kitchen-1***	***Living-1***	***Bed***	***Kitchen-2***	***Living-2***	***Luke***	***Gates362***	***Gates381***	***Lounge***	***Manolis***	***5a***	***5b***
	Train	744	1035	890	782	731	1370	3540	2950	933	1623	1000	1391
	Test	357	493	244	230	359	624	386	1053	327	807	497	405
Cambridge	**Scenes**	***Court***	***College***	***Hospital***	***Church***	***Shop***	-	-	-	-	-	-	-
	Train	1531	1220	895	1487	231	-	-	-	-	-	-	-
	Test	760	343	182	530	103	-	-	-	-	-	-	-

Table IV. The rendered image quality on the test set for each scene.

7-Scenes	**Scenes**	***Chess***	***Fire***	***Heads***	***Office***	***Pumpkin***	***Kitchen***	***Stairs***	-	-	-	-	-
	PSNR	24.76	22.29	18.90	22.29	24.99	21.29	19.70	-	-	-	-	-
12-Scenes	**Scenes**	***Kitchen-1***	***Living-1***	***Bed***	***Kitchen-2***	***Living-2***	***Luke***	***Gates362***	***Gates381***	***Lounge***	***Manolis***	***5a***	***5b***
	PSNR	20.66	26.66	27.46	25.82	24.33	23.91	20.63	22.82	22.85	20.31	26.81	17.72
Cambridge	**Scenes**	***Court***	***College***	***Hospital***	***Church***	***Shop***	-	-	-	-	-	-	-
	PSNR	14.42	12.79	13.84	14.30	14.48	-	-	-	-	-	-	-

Table V. The relocalization accuracy for each scene on the 12-Scenes dataset.

Scenes	***Kitchen-1***	***Living-1***	***Bed***	***Kitchen-2***	***Living-2***	***Luke***	***Gates362***	***Gates381***	***Lounge***	***Manolis***	***5a***	***5b***
Avg. Err [cm/°] ↓	0.29/0.18	0.25/0.12	0.24/0.11	0.25/0.17	0.25/0.12	0.45/0.18	0.32/0.13	0.30/0.13	0.48/0.14	0.32/0.13	0.37/0.16	0.36/0.14
R**@[2cm, 2°]** ↑	98.3	100	99.6	100	100	97.3	99.7	98.8	96.9	93.1	96.6	86.9

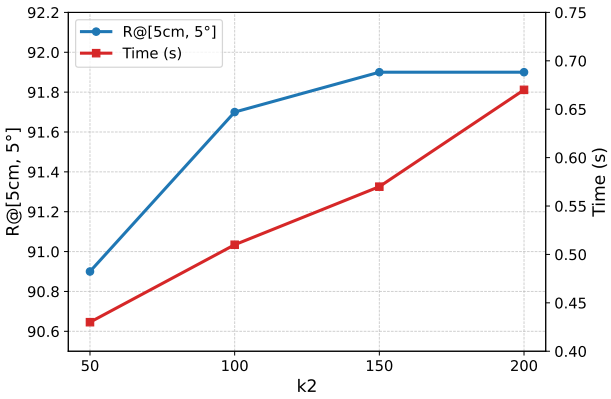

Figure III. The R@[5cm, 5°] accuracy and the average initial relocalization time under different values of k_2.

on the Stairs scene. Introducing a small positional perturbation (e.g., $b = 0.2\,\text{m}$) already produces a noticeable improvement, and the performance peaks at moderate magnitudes ($b = 0.5$–$1.0\,\text{m}$). When the perturbation becomes too large (e.g., $b = 1.5\,\text{m}$), the accuracy drops, as the perturbed viewpoints deviate excessively from plausible poses and reduce the reliability of the rendered feature cues. Based on these observations, we adopt $(a = 5°, b = 0.5\,\text{m})$ as the default configuration in the adaptive retrieval pipeline.

Number of perturbations. A larger k_2 provides more pose-perturbed virtual views, increasing the chance of finding a viewpoint closer to the query. As shown in Fig. III, the recall R@[5cm, 5°] on the Stairs scene improves rapidly when k_2 increases from 50 to 100, and marginally from 100 to 150, while no further gain is observed at $k_2 = 200$. In contrast, the initialization time grows approximately linearly with k_2, since more virtual views are rendered and verified. Considering this trade-off between accuracy and efficiency, we adopt $k_2 = 150$ for indoor scene.

Visualization. Fig. IV provides a qualitative visualization of the Adaptive Viewpoint Retrieval process. The coarse retrieval stage selects reference images with limited viewpoint overlap due to the sparse image observations, resulting in only a small number of geometrically consistent matches. After synthesizing pose-perturbed virtual views, the fine retrieval stage identifies reference images that are significantly closer to the query viewpoint. This leads to a substantial increase in co-visible regions and a much larger set of verified inliers, as illustrated by the dense green correspondences on the right.

B. Scene Metadata

We report the number of training and test images for each scene in the 7-Scenes, 12-Scenes, and Cambridge Landmarks datasets in Table III. Moreover, we apply sky and dynamic-object masks to exclude these regions from supervision, preventing their instability from degrading the FGS training process in the Cambridge Landmarks dataset.

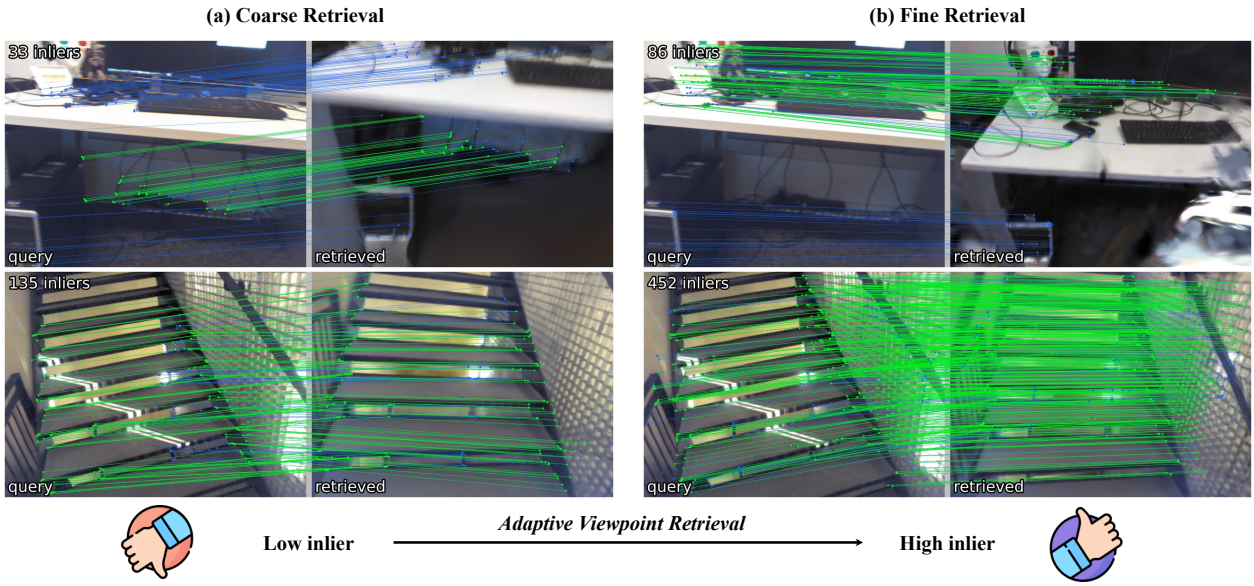

Figure IV. Qualitative visualization of the Adaptive Viewpoint Retrieval process. Green lines denote the inlier correspondences that pass geometric verification, whereas the other matches are visualized in blue. (a) Coarse retrieval often yields candidates with limited co-visibility and low inlier counts. (b) By synthesizing pose-perturbed virtual views and performing fine retrieval, our method significantly increases co-visibility, resulting in substantially more inliers.

C. Mapping Quality

We report the PSNR of the FGS models trained within our SplatHLoc framework for each scene in Table IV. In indoor scenes, the PSNR typically reaches around 20 dB, while in outdoor scenes it is closer to 15 dB. This trend is consistent with the characteristics of the datasets: indoor environments contain more stable lighting conditions and richer geometric structures, enabling the FGS model to fit the scene appearance more accurately. In contrast, outdoor scenes exhibit stronger illumination changes, dynamic elements, and larger viewpoint variations, making high-fidelity rendering more challenging. Moreover, the Cambridge Landmarks dataset provides fewer training images for each scene. Nevertheless, the obtained PSNR values demonstrate that the trained FGS maps retain sufficient photometric fidelity to support robust retrieval and matching within our pipeline.

D. Additional Experimental Results

12-Scenes. We report the relocalization results of our SplatHLoc on each scene of the 12-Scenes dataset in Table V. Across all scenes, our method achieves low translation and rotation errors, with most scenes exhibiting average errors below 0.3 cm and 0.15°. The corresponding recall under the stringent R@[2 cm, 2°] metric reaches nearly 100% on the majority of scenes, demonstrating the robustness of SplatHLoc in challenging indoor environments.

Map size. Our method requires storing additional VPR features of database images during the mapping process. In Table 4 (main text), we only compared the size of the FGS map. For the chess scene, storing the VPR features requires 62.50 MB. Nonetheless, the total map size of our method for the chess scene remains less than half that of STDLoc.

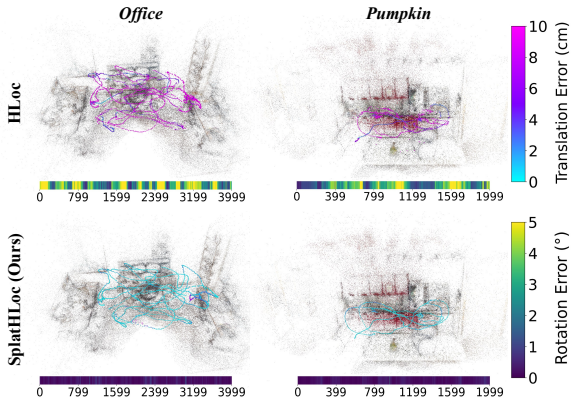

Figure V. Additional comparison of camera pose estimation errors for the 7-Scenes dataset.

E. Additional Visualization

Relocalization error. Fig. V shows a comparison of the camera pose estimation errors in the *Office* and *Pumpkin* scenes. The error maps further show that SplatHLoc reduces outlier predictions and maintains lower errors across the entire sequence, demonstrating superior robustness in challenging indoor scenes. For trajectory points exceeding the upper error limit (10cm, 2°), the color is set to the value corresponding to that limit.

Fig. VII provides qualitative visualizations of the relo-

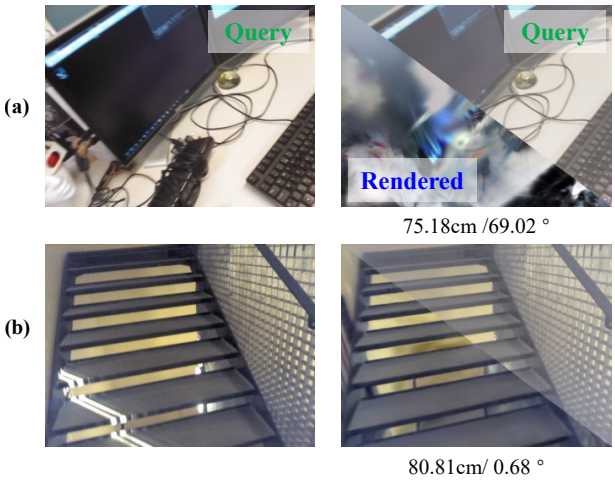

Figure VI. Visualization of the failure cases. (a) The first row illustrates a failure case caused by poor rendering quality of the Gaussian map near sparsely observed viewpoints. (b) The second row shows a failure induced by repetitive structures in the scene.

calization errors across all scenes in the Cambridge Landmarks dataset. Across Court, College, Hospital, Shop, and Church, the rendered views closely align with the corresponding query images, demonstrating that SplatHLoc achieves highly accurate pose estimation even in large-scale outdoor environments.

In Fig. VIII, we visualize the relocalization errors for each scene in the 7-Scenes dataset. Across all scenes, SplatHLoc consistently produces well-aligned query–rendered pairs, demonstrating the robustness and reliability of our approach under varying levels of texture, clutter, and viewpoint change.

Failure Cases. Fig. VI presents representative failure cases observed during evaluation. In case (a), the failure arises from poor rendering quality near sparsely observed viewpoints in the training set. When the Gaussian map does not sufficiently cover a region, the synthesized image deviates significantly from the true appearance, leading to ultimately incorrect pose estimation. Case (b) shows a different type of failure induced by highly repetitive structures in the scene. In such environments, distinct locations may produce visually similar local patterns, causing the feature matching stage to converge to an incorrect alignment despite low rotational ambiguity. These cases highlight challenging scenarios where even high-quality novel view synthesis may be insufficient for accurate relocalization.

F. Gaussian Rasterization in FGS

We adopt the differentiable rasterization procedure of 3D Gaussian Splatting. Each Gaussian primitive is parameterized as $\mathcal{G}_i = \{\mathbf{x}_i, \mathbf{q}_i, \mathbf{s}_i, \alpha_i, \mathbf{c}_i, \mathbf{f}_i\}$, where $\mathbf{x}_i \in \mathbb{R}^3$ denotes the center, $\mathbf{q}_i$ the rotation, $\mathbf{s}_i$ the anisotropic scale, α_i the base opacity, $\mathbf{c}_i$ the color, and $\mathbf{f}_i \in \mathbb{R}^d$ the feature vector.

3D covariance in world coordinates. The intrinsic 3D covariance of Gaussian i in the world coordinate frame is

$$\Sigma^{3D}_{i,\text{world}} = R(\mathbf{q}_i)\,\text{diag}(\mathbf{s}_i^2)\,R(\mathbf{q}_i)^\top, \tag{I}$$

where $R(\mathbf{q}_i)$ is the rotation matrix associated with quaternion $\mathbf{q}_i$.

Transformation to the camera frame. Given the viewing transformation $W \in \text{SO}(3)$ (the rotation of world-to-camera pose), the 3D covariance in the camera coordinate system becomes

$$\Sigma^{3D}_{i,\text{cam}} = W\,\Sigma^{3D}_{i,\text{world}}\,W^\top. \tag{II}$$

Projection to the image plane. Under the camera projection $\pi : \mathbb{R}^3 \to \mathbb{R}^2$, the corresponding 2D covariance is obtained by linearizing π at the Gaussian center:

$$\Sigma^{2D}_i = J_i\,\Sigma^{3D}_{i,\text{cam}}\,J_i^\top, \qquad J_i = \left.\frac{\partial \pi(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_i}. \tag{III}$$

Projected kernel and opacity. The 2D Gaussian kernel on the image plane is

$$G_i(\mathbf{u}) = \exp\!\left(-\tfrac{1}{2}(\mathbf{u}-\mu_i)^\top(\Sigma^{2D}_i)^{-1}(\mathbf{u}-\mu_i)\right), \tag{IV}$$

where $\mu_i = \pi(\mathbf{x}_i)$. The per-pixel opacity is

$$\alpha_i(\mathbf{u}) = \alpha_i\,G_i(\mathbf{u}). \tag{V}$$

Volume compositing. Given Gaussians sorted in front-to-back order, the accumulated transmittance is

$$T_i = \prod_{j<i}(1-\alpha_j). \tag{VI}$$

The rendered color is obtained via alpha compositing:

$$\mathbf{C}(\mathbf{u}) = \sum_{i\in\mathcal{N}(\mathbf{u})} \mathbf{c}_i\,\alpha_i(\mathbf{u})\,T_i(\mathbf{u}), \tag{VII}$$

where $\mathcal{N}(\mathbf{u})$ denotes the Gaussians influencing pixel $\mathbf{u}$.

Depth and feature rendering. Depth rendering replaces the color with the camera-space depth z_i:

$$\mathbf{D}(\mathbf{u}) = \sum_{i\in\mathcal{N}(\mathbf{u})} z_i\,\alpha_i(\mathbf{u})\,T_i(\mathbf{u}). \tag{VIII}$$

Feature rendering follows the similar form:

$$\mathbf{F}(\mathbf{u}) = \text{norm}\left(\sum_{i\in\mathcal{N}(\mathbf{u})} \text{norm}(\mathbf{f}_i)\,\alpha_i(\mathbf{u})\,T_i(\mathbf{u})\right), \tag{IX}$$

where $\text{norm}(\cdot)$ denotes the L2 normalization operation.

Differentiability. All operations—covariance transformation, projection, kernel evaluation, opacity accumulation, and compositing—are differentiable with respect to $\{\mathbf{x}_i, \mathbf{q}_i, \mathbf{s}_i, \alpha_i, \mathbf{c}_i, \mathbf{f}_i\}$, enabling the end-to-end optimization of all Gaussian attributes.

Figure VII. Qualitative visualization of the relocalization errors on the Cambridge Landmarks dataset. For each image pair, the upper-left triangle shows the query image, while the lower-right triangle displays the rendered image obtained from the final estimated pose. Better visual alignment between the two indicates that our SplatHLoc achieves more accurate relocalization.

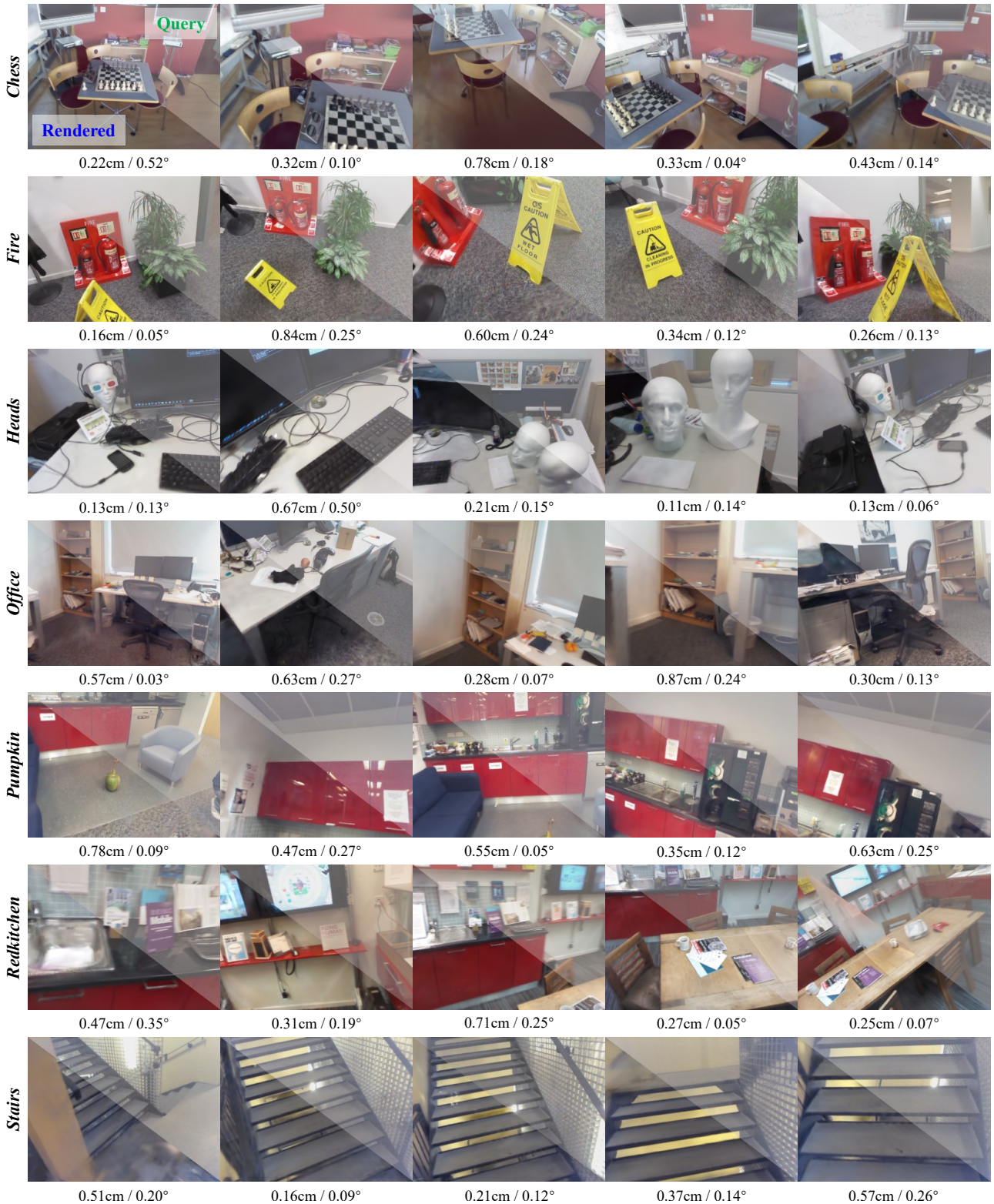

Figure VIII. Qualitative visualization of the relocalization errors on the 7-Scenes dataset. For each image pair, the upper-left triangle shows the query image, while the lower-right triangle displays the rendered image obtained from the final estimated pose. Better visual alignment between the two indicates that our SplatHLoc achieves more accurate relocalization.

Algorithm 1 Adaptive Coarse-to-Fine Viewpoint Retrieval

Require: Query image I_q, Gaussian map $\mathcal{G}$, VPR model $\mathcal{V}$, sparse matcher $\mathcal{M}_{\text{sparse}}$,
training images $\{I_t\}$ with poses $\{\mathbf{T}_t\}$,
retrieval hyperparameters $(k_1, k_2, k_3, a, b, \mathcal{I})$.
Ensure: Final candidate image I_c and its pose $\mathbf{T}^\star$ for I_q.

// Coarse viewpoint retrieval
Extract global descriptor $\mathbf{v}_q = \mathcal{V}(I_q)$.
Compute descriptors $\{\mathbf{v}_t\}$ for all training images.
Retrieve top-k_1 neighbors $\{I_c^1, \ldots, I_c^{k_1}\}$ of I_q.
Initialize $N^\star \leftarrow 0$, $\mathbf{T}^\star \leftarrow \varnothing$, $i^\star \leftarrow -1$.
for $i = 1$ to k_1 **do**
  $I_c \leftarrow I_c^i$; $\mathbf{T}_c \leftarrow$ pose of I_c from $\{\mathbf{T}_t\}$.
  matches $\leftarrow \mathcal{M}_{\text{sparse}}(I_q, I_c)$.
  Estimate inliers $\mathcal{I}_c$.
  $N \leftarrow |\mathcal{I}_c|$.
  if $N > N^\star$ **then**
    $N^\star \leftarrow N$, $\mathbf{T}^\star \leftarrow \hat{\mathbf{T}}$, $i^\star \leftarrow i$.
  end if
  if $N^\star \geq \mathcal{I}$ **then**
    break // sufficient inliers, stop coarse search
  end if
end for
$I_c^c \leftarrow I_c^{i^\star}$ // best coarse candidate image
// Fine retrieval via virtual novel viewpoints
$I_c^f \leftarrow I_c^c$ // by default, fall back to the coarse candidate
if $N^\star < \mathcal{I}$ **then**
  Generate k_2 perturbed poses $\{\mathbf{T}_v^j\}_{j=1}^{k_2}$ around $\mathbf{T}^\star$ within $(a^\circ,\ b\ \text{m})$.
  for $j = 1$ to k_2 **do**
    Render virtual view $I_v^j = \text{Render}(\mathcal{G}, \mathbf{T}_v^j)$.
    Extract descriptor $\mathbf{v}_v^j = \mathcal{V}(I_v^j)$.
  end for
  Build a VPR database with $\{\mathbf{v}_v^j\}_{j=1}^{k_2}$.
  Retrieve top-k_3 virtual views $\{I_v^{j_1}, \ldots, I_v^{j_{k_3}}\}$ for I_q.
  Initialize $j^\star \leftarrow -1$.
  for each index $j \in \{j_1, \ldots, j_{k_3}\}$ **do**
    matches $\leftarrow \mathcal{M}_{\text{sparse}}(I_q, I_v^j)$.
    Estimate pose inliers $\mathcal{I}_v$.
    $N \leftarrow |\mathcal{I}_v|$.
    if $N > N^\star$ **then**
      $N^\star \leftarrow N$, $\mathbf{T}^\star \leftarrow \hat{\mathbf{T}}$, $j^\star \leftarrow j$.
    end if
  end for
  if $j^\star \geq 0$ **then**
    $I_c^f \leftarrow I_v^{j^\star}$ // best virtual candidate image
  end if
end if
$I_c \leftarrow I_c^f$ // final candidate image
return $I_c, \mathbf{T}^\star$.