Supplementary Material for "Towards Accurate Active Camera Localization"

Qihang Fang[1*], Yingda Yin[2*], Qingnan Fan[3†], Fei Xia[4], Siyan Dong[1]
Sheng Wang[5], Jue Wang[3], Leonidas Guibas[4], Baoquan Chen[2†]

[1]Shandong University, [2]Peking University
[3]Tencent AI Lab, [4]Stanford University, [5]3vjia

The appendix provides the additional supplemental material that cannot be included in the main paper due to its page limit:

- Algorithm illustration
- More results.
- More analysis.
- More implementation details.
- More details of the ACL-synthetic/real datasets.

A Algorithm illustration

We summarize our proposed algorithm in Algorithm 1.

B More results

B.1 Comparison on the sparse data

The posed RGB-D stream is the basis for both passive and active localization. To further validate the robustness of our proposed algorithm, we discard half of the posed RGB-D stream as the sparse data for evaluation. The numerical comparisons with the best baselines (Camera-descent/Scene-descent) on both the sparse data and dense data (default setting in the main paper) are shown in Table 1. We observe that all the methods achieve worse results on the sparse data as expected, yet our approach still outperforms the other competitors.

B.2 Comparison on more real-world datasets

To further evaluate the compatibility of our method, we compare our approach and its best competitors on 10 scenes of the real-world Gibson V2 [7] and Replica [4] datasets besides ACL-synthetic/-real datasets in the main paper. Shown in Table 2, our results consistently outperforms the others.

B.3 More qualitative results

In Figure 1, we show the qualitative results of the intelligent behaviors of our algorithm on more test scenes.

* Equal contribution; ordered alphabetically.
† Corresponding authors.

Algorithm 1 The full pipeline of our algorithm

function PASSIVE LOC. MODULE(observation $I^{(t)}$, posed RGB-D stream $\{I^{(i)}_{basis}, C^{(i)}_{basis}\}^{m}_{i=1}$)
  if *initialization* **then**
    initialization ← false
    Adapt the passive localizer by posed RGB-D stream $\{I^{(i)}_{basis}, C^{(i)}_{basis}\}^{m}_{i=1}$
    Construct the scene model D_{scene} by fusing posed RGB-D stream $\{I^{(i)}_{basis}, C^{(i)}_{basis}\}^{m}_{i=1}$
  Current pose estimation $\widehat{C}^{(t)}$ ← Passive localizer($I^{(t)}$)
  return $\widehat{C}^{(t)}$
function ACTIVE LOC. MODULE(pose estimation $\widehat{C}^{(t)}$, scene model D_{scene})
  $M^{(t)}_{wd}, M^{(t)}_{cd}$ ← Scene uncertainty computation($\{\widehat{C}^{(t)}, D_{scene}\}$)
  $U^{(t)}_{cu}$ ← Camera uncertainty computation($\{\widehat{C}^{(t)}, D_{scene}\}$)
  Action $a^{(t)}$ ← Policy network($\{M^{(t)}_{wd}, M^{(t)}_{cd}\}$)
  return $U^{(t)}_{cu}$, $a^{(t)}$
procedure ENTIRE PIPELINE(posed RGB-D stream $\{I^{(i)}_{basis}, C^{(i)}_{basis}\}^{m}_{i=1}$, accuracy threshold λ_{cu})
  $t \leftarrow 0$
  $D_{scene} \leftarrow NULL$
  initialization ← true
  while $t <$ maximum step length **do**
    Obtain the current observation $I^{(t)}$
    $\widehat{C}^{(t)}$ ← PASSIVE LOC. MODULE($I^{(t)}, \{I^{(i)}_{basis}, C^{(i)}_{basis}\}^{m}_{i=1}$)
    $U^{(t)}_{cu}$, $a^{(t)}$ ← ACTIVE LOC. MODULE($\widehat{C}^{(t)}, D_{scene}$)
    if $U^{(t)}_{cu}$ is within λ_{cu} cm, λ_{cu} degrees **then**
      break
    Execute the action $a^{(t)}$
    $t \leftarrow t + 1$
  return $\widehat{C}^{(t)}$

C More analysis

We provide more analysis of the camera uncertainty component below.

The iterative closest point (ICP) approach is based on the general assumption that the two input point clouds are roughly aligned. When the estimated camera pose of the current frame is far from its ground truth, such as 20cm, 20°, the camera uncertainty component generated by ICP becomes unstable and not reliable to determine the adaptive stop condition. To be specific, following the experiment of "Analysis of camera uncertainty" in the main paper, we further summarize that when the estimated relative pose is within 20cm, 20° ($\lambda_{cu} = 20$), about 83.57% (3220/3853) samples are truly within 20cm, 20° compared to the ground truth, which is much smaller than 94.14% for 5cm, 5° ($\lambda_{cu} = 5$).

Therefore, a natural question to ask is, when evaluating the camera pose in a coarse level, such as 20cm, 20°, what is the best parameter value (λ_{cu}) to determine the adaptive stop condition for the highest camera pose accuracy? In Table 3, we compare the numerical results of our algorithm trained with different parameter values ($\lambda_{cu} = 5/20$) and evaluated on the coarse-scale accuracy (20cm, 20°). We observe that the camera pose accuracy is much worse with $\lambda_{cu} = 20$, which validates the parameter selection λ_{cu} for the camera uncertainty component.

Table 1. Numerical results on both the dense and sparse data.

		ACL-synthetic		ACL-real	
Data	Methods	Acc (%)	#steps	Acc (%)	#steps
Dense	Camera-descent (t+1)	61.55	22.90	61.40	26.85
	Camera-descent (t+2)	55.30	22.60	59.20	25.78
	Scene-descent	57.65	18.56	54.20	16.87
	Ours	**83.05**	17.33	**82.40**	17.90
Sparse	Camera-descent (t+1)	55.45	24.17	49.60	31.62
	Camera-descent (t+2)	55.05	28.15	52.40	37.35
	Scene-descent	19.90	6.12	41.40	22.71
	Ours	**82.00**	20.52	**76.40**	22.54

Table 2. Numerical results on GibsonV2 and Replica datasets.

	GibsonV2		Replica	
Methods	Acc (%)	#steps	Acc (%)	#steps
Camera-descent (t+1)	57.60	23.51	67.80	19.04
Camera-descent (t+2)	51.60	25.42	69.80	26.13
Scene-descent	56.20	16.16	62.80	14.60
Ours	**75.00**	15.27	**86.80**	13.30

D More implementation details

D.1 Policy network

The policy network takes the scene uncertainty component as input and generates the probability of the three actions defined by the action space. The camera-driven scene map is represented as a 3-channel 2D map $M_{cd}^{(t)}$, which can be easily consumed by the convolution operation. We employ a convolution neural network of 6 convolution layers (32-64-128-128-256-256) and 1 linear layer (64) to extract the global feature ($\mathbb{R}^{64}$). Each convolution layer is of kernel size 3x3 and followed by a batch normalization layer and a max pooling layer of stride 2. The world-driven scene map is represented as a 6-channel point cloud $M_{wd}^{(t)}$. Inspired by the popular point cloud processing network PointNet [2], we employ a three-layer pointwise MLP (64-128-64) followed by a max pooling layer to extract its global feature ($\mathbb{R}^{64}$). Finally, by concatenating all the extracted features, we use a three-layer MLP (64-16-3) to predict the probability of the three predefined actions.

D.2 Noise perturbation on the action space

To simulate robotic agents in a real-world condition, the action does not lead to perfect execution, hence we add the Gaussian noise to each action. To be

Table 3. Numerical results of our algorithm trained with different parameter values ($\lambda_{cu} = 5, 20$) on the ACL-synthetic dataset.

Uncertainty parameters	Accuracy (20cm, 20°)
$\lambda_{cu} = 5$	85.92
$\lambda_{cu} = 20$	49.40

specific, if the agent turns left or right, the Gaussian noise of standard deviation (6) will be added to the rotation angle $\theta^{(t)}$ of mean value (20); if the agent moves forward, the Gaussian noise of standard deviation (5) will be added on the 2D positions $x^{(t)}, y^{(t)}$ of mean value (30). The positions are measured in centimeters, and the rotation angle is measured in degrees. Note the standard deviation (σ) is actually very large considering 31.74% of sampled noises are beyond σ for the Gaussian distribution.

D.3 Implimentation details

In our experiment, we employ the Adam [1] to optimize the network weights with the initial learning rate of 3×10^{-4}. Some hyper-parameters: $N_{cd} = 12, N_{wd\_r} = 1000, N_{wd\_p} = 2^{14} = 16384, N_f = 5, X = 256, Y = 256$.

E More details of the ACL-synthetic/real datasets

The posed RGB-D stream in the existing camera localization datasets [3,5,6] is usually obtained by scanning the environment with handheld sensors by human operators, hence does not always cover the complete scene model. We design the posed RGB-D stream in our dataset to simulate this effect. Directly visualizing the trajectory of the posed RGB-D stream in the scene is not intuitive as it would lose the orientation information of the camera pose, instead we choose to visualize the scene model reconstructed from the posed RGB-D stream to showcase how much scene region is covered by the posed RGB-D stream.

We illustrate the textured meshes of both the complete and reconstructed scene models for the ACL-synthetic and ACL-real datasets in Figure 2, 3 and 4. Their related statistics are shown in Table 4.

Table 4. Scene statistics of the ACL-synthetic and ACL-real dataset. We summarize the number of scenes, scene area, max scene area, min scene area and the number of frames in the RGB-D sequences. The unit for all areas is m^2. The Area and #frames metrics are averaged over all the scenes involved.

Scene		#scenes	Area	Max area	Min area	#frames
ACL-synthetic	Train split	15	37.89	49.40	25	58.00
	Test split	20	43.17	75.00	26.9	54.45
ACL-real	Test split	5	64.82	98.28	23.62	88.40
All		40	43.90	98.28	23.62	60.03

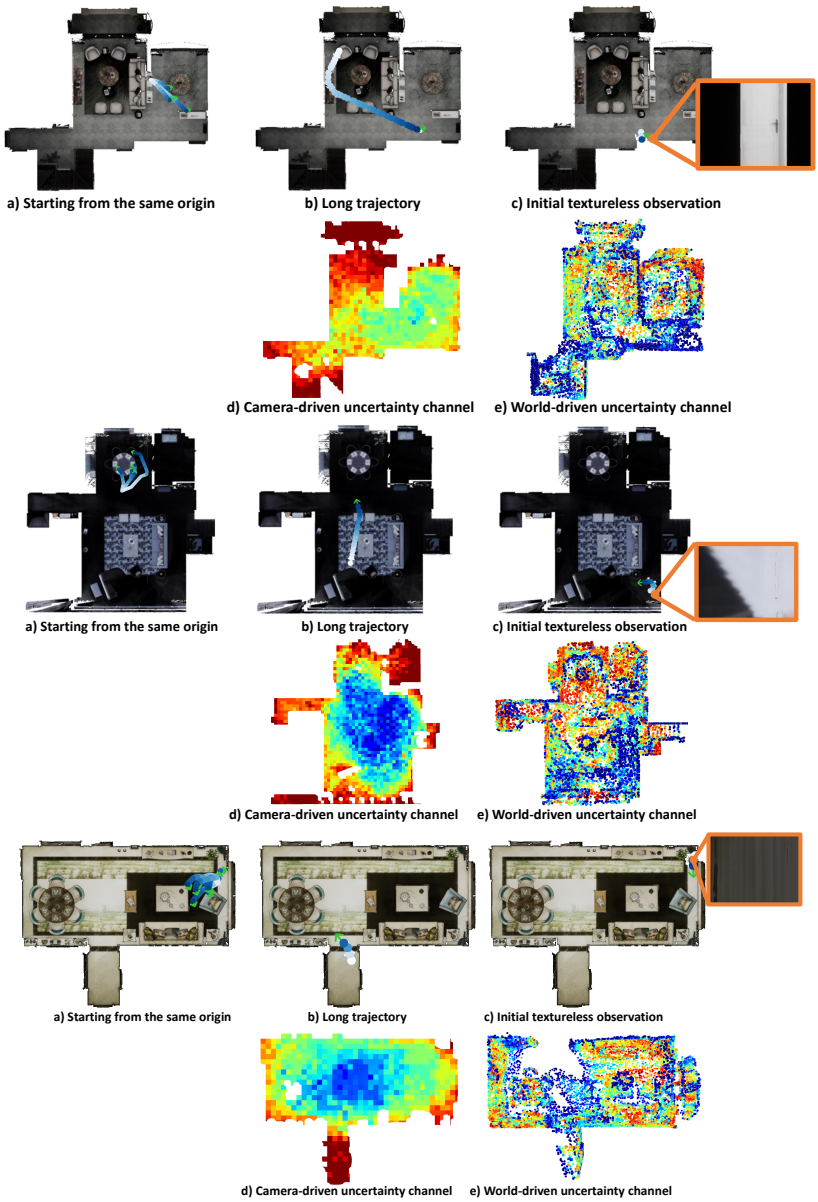

Fig. 1. Qualitative results of the intelligent behaviors learned by our algorithm.

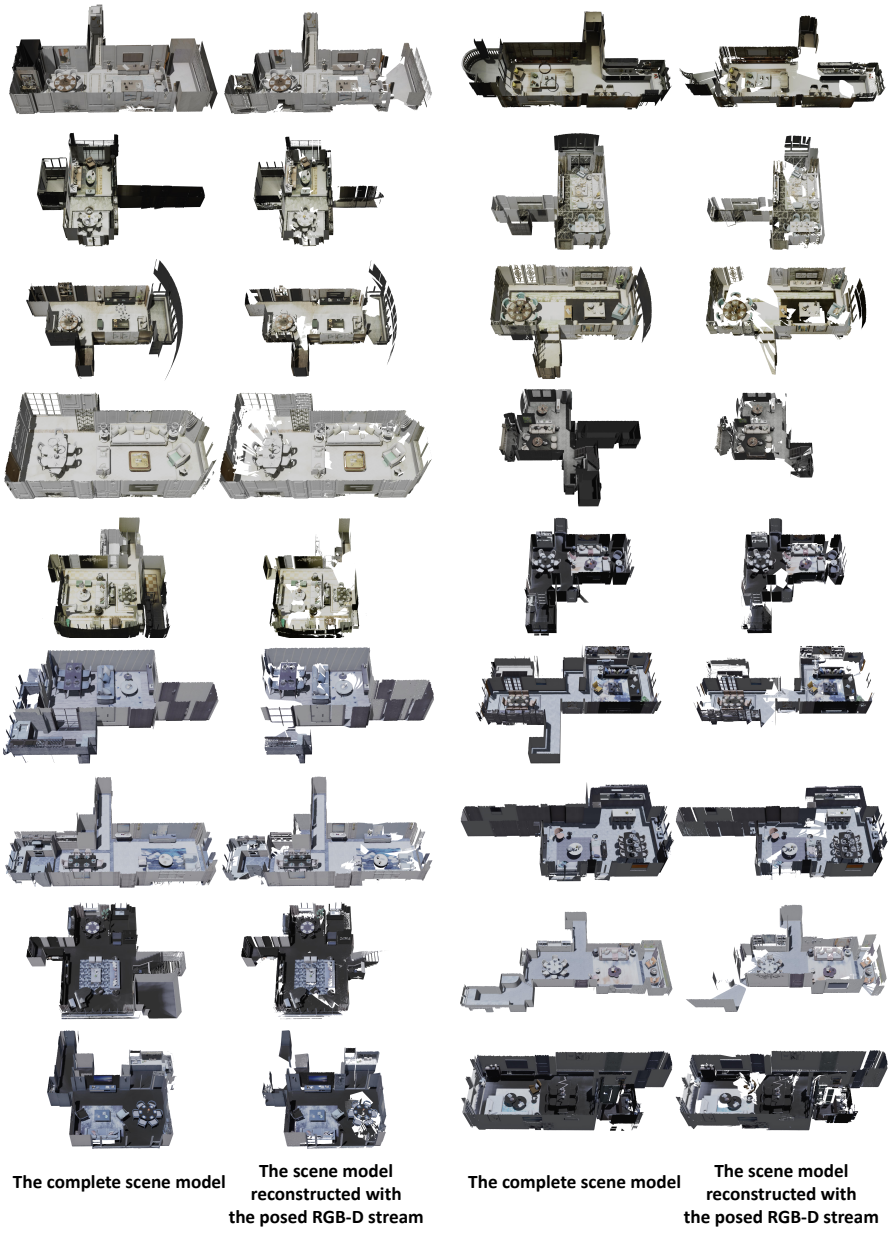

Fig. 2. Visualization of the ACL-synthetic dataset.

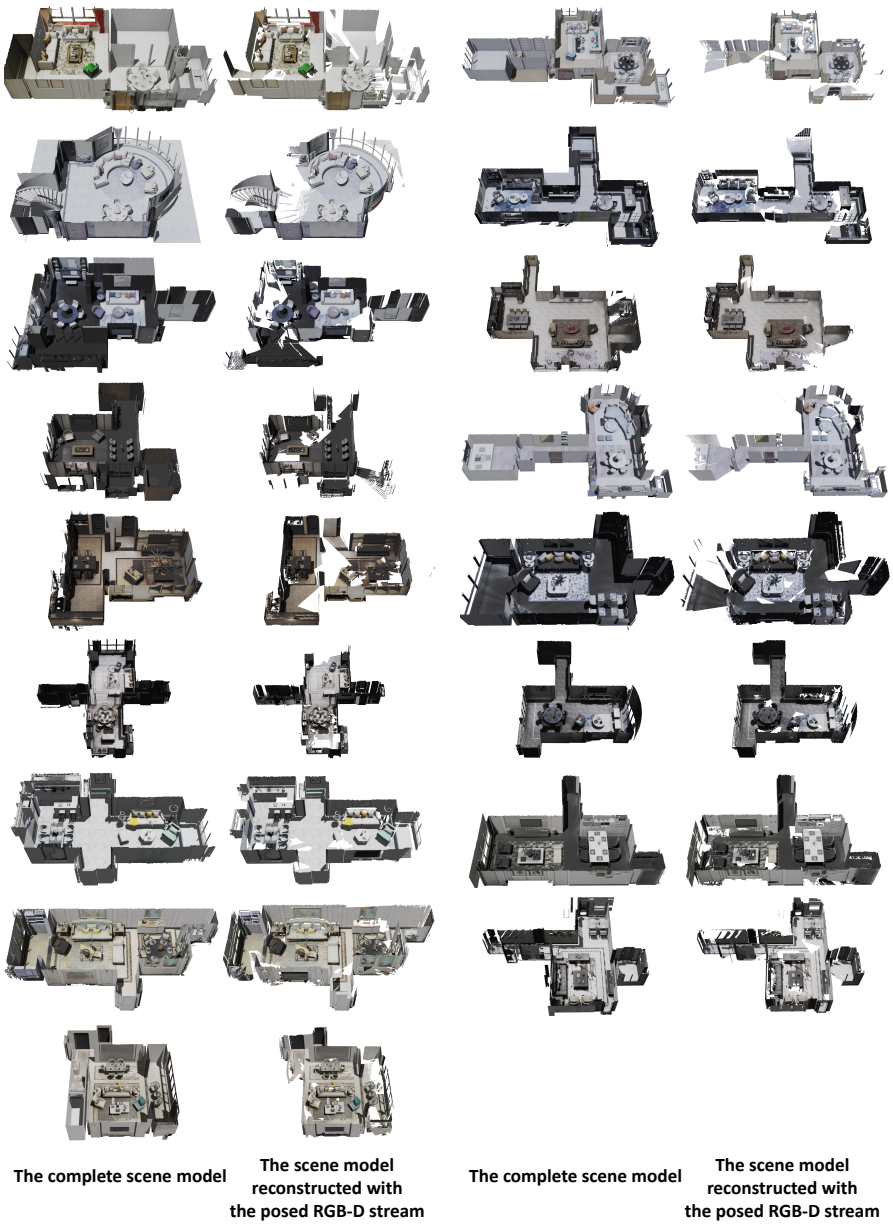

Fig. 3. Visualization of the ACL-synthetic dataset.

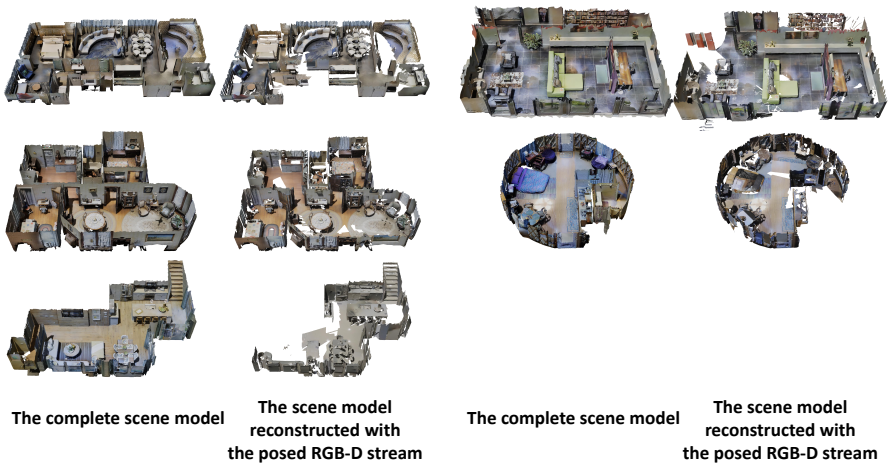

Fig. 4. Visualization of the ACL-real dataset.

References

1. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 4
2. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017) 3
3. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2930–2937 (2013) 4
4. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019) 1
5. Valentin, J., Dai, A., Nießner, M., Kohli, P., Torr, P., Izadi, S., Keskin, C.: Learning to navigate the energy landscape. In: 2016 Fourth International Conference on 3D Vision (3DV). pp. 323–332. IEEE (2016) 4
6. Wald, J., Sattler, T., Golodetz, S., Cavallari, T., Tombari, F.: Beyond controlled environments: 3d camera re-localization in changing indoor scenes. arXiv preprint arXiv:2008.02004 (2020) 4
7. Xia, F., Zamir, A.R., He, Z., Sax, A., Malik, J., Savarese, S.: Gibson env: Real-world perception for embodied agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9068–9079 (2018) 1