LASER: LAtent SpacE Rendering for 2D Visual Localization Appendix

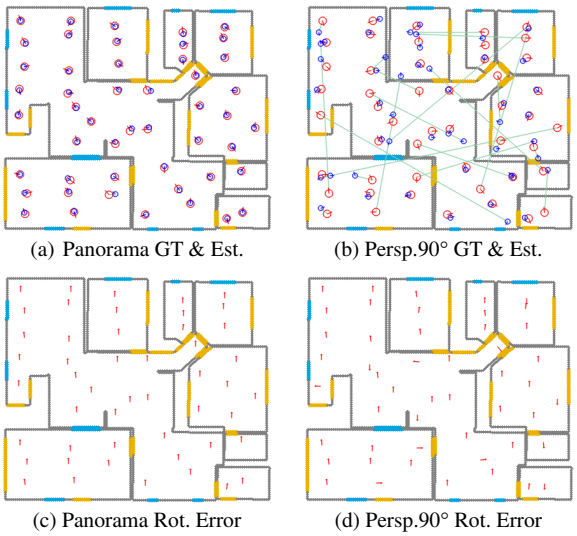
Zhixiang Min[1] Naji Khosravan[2] Zachary Bessinger[2] Manjunath Narayana[2]
Sing Bing Kang[2] Enrique Dunn[1] Ivaylo Boyadzhiev[2]
[1]Stevens Institute of Technology [2]Zillow Group

1. Supplementary

We include additional experiments in §1.1, §1.2, §1.3, §1.4, §1.5, §1.6, §1.7. Additional implementation details are in §1.8. More experiment justifications are in §1.9.

1.1. Localization Error Visualization

To visualize the localization error, we picked a floor map that has a high density of panoramas from ZInD. Such high density samples will allow us to analyze the results of our framework based on a wide range of imagery locations within the room geometry. Fig.1 shows localization errors for both panoramas and perspective crops from those panoramas. As can be seen, our framework has a consistent performance with panorama queries, showing minimal bias to sampling location of within the rooms (i.e. room center/corner) or to room attributes (i.e. size/shape of the room). For perspective queries, certain amount of failure cases emerges due to the ambiguities. The rotation error visualization shows that the failure cases are usually subject to the canonical rotation errors (i.e. $-90°$, $90°$, $180°$), indicating a non-random failure pattern (i.e. failed from ambiguities) within the Manhattan world map.



(a) Panorama GT & Est. (b) Persp.90° GT & Est.

(c) Panorama Rot. Error (d) Persp.90° Rot. Error

Figure 1. **Localization Error Visualization.** (Top) GT and estimation visualization. Red oriented-circles indicate GTs, and blue are estimations. Corresponding estimations are connected with green lines. (Bottom) Rotation error are visualized with arrowed lines, where straight-up lines indicates zero rotation error.

1.2. Robustness to Map Noise

To measure robustness of our framework to the noise in maps, we add Gaussian noise with different variances to the map points' location. In this setting, other input features of map points are kept the same. We did not train or fine-tune the models with the noisy maps. Our results show our method is robust to a reasonable level of noise (i.e. std=0.1m) while the performance gradually degrades with the increasing noise levels. The proposed method still produces a 67% 1m recall for panorama queries and 30% for persp-90° queries with noise levels as high as std=0.5m, where the map is largely corrupted.

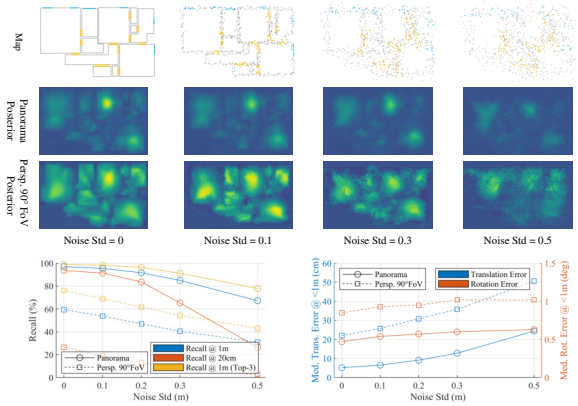


Figure 2. **Robustness to map noise.** (Top) Maps with different Gaussian noise levels. (Bottom) Recall/accuracy under different noise levels.

1.3. Robustness to Map Sampling Intervals

We test the performance of our method under different map sampling intervals. Fig.3 shows that our method consistently gains better performance with denser map sampling. Such improvements becomes marginal for finer intervals than 10cm, which suggests 10cm as a good balance between the computational cost and localization accuracy.

1.4. Robustness to Camera Roll/Pitch

Fig.4 shows the effect of noisy (i.e. non-zero) camera pitch/roll to the performance of our method with perspective queries. To this end, we added variations to the camera pitch/roll sampled from a Gaussian distribution with different variances when cropping perspective images from panoramas. Such variations are not used for training/fine-tuning our model. Our experiments show that pitch/roll variations have more impact on rotation error while having a less impacts on translation.

1.5. Performance to Data Attributes

In this section we study effects of two attributes: furnishing-level and query image saliency.

Furnishing level. Table.5(Left) shows the performance of our method in the presence of different furniture levels on S3D dataset. As can be seen the performance of our method is only slightly impacted by furniture.

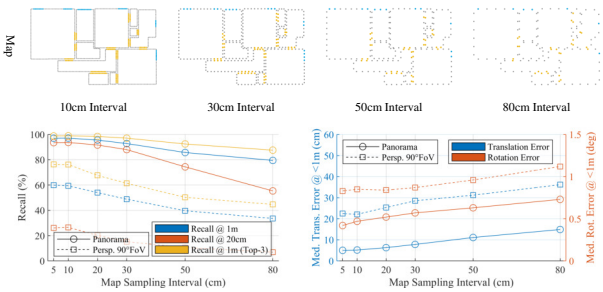


Figure 3. **Robustness to map sampling intervals.** (Top) Visual examples of different map sampling intervals. (Bottom) Recall/accuracy w.r.t. different map sampling intervals.

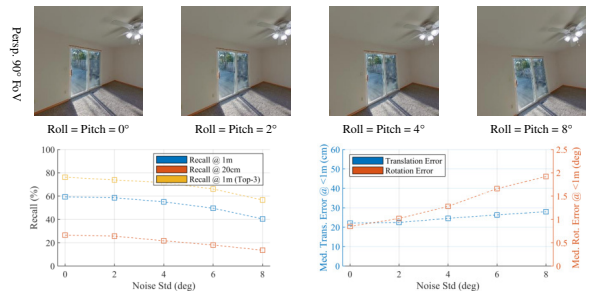


Figure 4. **Robustness to camera pitch/roll (persp-90°).** (Top) Visual examples of perspective images with different roll/pitch. (Bottom) Recall/accuracy under different roll/pitch.

Furnishing	Recall @ 1m (%) Panorama	Persp. 90° FoV Random	Persp. 90° FoV Saliency
	-		
Empty	96.23	58.40	75.19
Simple	95.82	56.92	69.12
Full	95.52	55.80	64.17

Figure 5. **Performance of data attributes.** (Left) Performance under furnishing-levels and photo saliency. (Right) Saliency-aware sampling estimates column-wise saliency value for panorama and crops the perspective image from the region with highest saliency response.

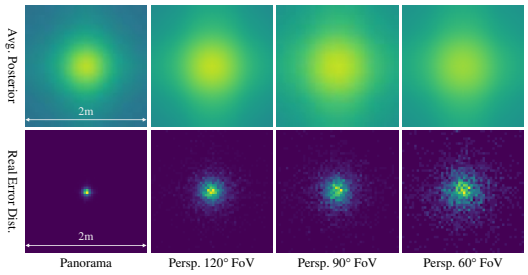


Figure 6. **Uncertainty estimation v.s. real error distribution.** The uncertainty estimation is shown with averaged posterior map crops around GT location. The real translation error distribution is shown in 2D histograms where GT locates at centers.

Query Image Saliency. To study the influence of content richness in the query images, we introduce a saliency-aware image cropping as shown in Fig.5(Right). While the perspective images in our experiments are cropped from the dataset panoramas, this saliency-aware strategy crops perspective images with a yaw-angle that gives the highest saliency response under the spectral residual approach [3]. Contrary to the default random cropping, the saliency-aware strategy simulates human photography behavior of taking photos with rich contents. Photos with higher saliency effectively avoids capturing less-informative photos (e.g. empty walls) that improves the recall largely. When the room is less furnished, the improvements become higher since the saliency has a higher chance to capture the localization landmarks such as windows/doors that are helpful for localization instead of interior/furniture.

1.6. Uncertainty Qualitative Study

We study the goodness of uncertainty estimation by comparing it with the empirical error distribution. As shown in Fig.6, with increasingly challenging (i.e. less FoV) query images, the variance of localization error grows larger. This is successfully reflected in our posterior estimation, where the model becomes less certain on the GT location and the predicted likelihoods diffuse to a larger region.

1.7. Map retrieval.

In this experiment, we test if we can retrieve the map that an arbitrary query panorama belongs to from the map set. We randomly pick one panorama from each map, and exhaustively match the panoramas to all maps. We record the maximum score of each matched posterior map in an affinity matrix. As shown in in Fig.7, the result affinity matrix has a clear diagonal with high response values, which reflects the high recall rate of our method. The non-diagonal elements of the matrix has a row-dependent pattern, that reflects the 'commonness' of the room in the query image. We also sorted the map with the number of rooms they have. Maps with less rooms have lower scores for all queries on average, which is reflected with its dimmer column in the matrix. The right histogram show the rank of GT floor maps, where our model has 70% and 50% top-1 accuracy on ZInD and S3D for retrieving the correct map from more than 200 maps. This reflects a good expected performance if the map is in a very large scale (e.g. shopping mall).

1.8. Implementation Details.

Training details. Our training uses Adam optimizer [6] on batch size of 8. During the training, maps are randomly sampled to a fixed 2048 points, which approximately gives a 5cm average sampling interval for both ZInD and S3D. The map point coordinates are scaled up to meter and normalized to have zero mean. We apply random rotation transforms to all map points as training augmentation. A PointNet [8] without feature transform is used for encoding the map to rendering codebooks. The image branch processes the feature maps from the last layer of a ResNet50 [2] into circular features. The ResNet is initialized with pretrained weights on ImageNet [7]. The refinement branch uses two 1D convolution layers with circular padding followed by a fully-connected layer. The 1D convolution layers both have kernel size of 3 and stride of 1.

Hard negative sampling. A uniform random sampling strategy in the 3D camera pose space most likely yields easy negative samples that does not efficiently contribute to the training. Hence, for training efficiency, we split the 100 negative samples evenly to three groups. We let the first group have random rotation and translation, second group have gt rotation and random translation, third group have random rotation and gt translation. This sampling strategy is an augmentation for efficiently sample hard negative samples for distance and incident-angle codebooks respectively. Specifically, the second group (gt rotation, random translation) has a higher probability to hit the samples that are difficult to distinguish using the incident-angle codebook (i.e. same rotations yield similar incident-angle observations in Manhattan world map). This forces the network utilize the distance codebooks to distinguish those samples. Similarity, the third group (random rotation, gt translation) are used

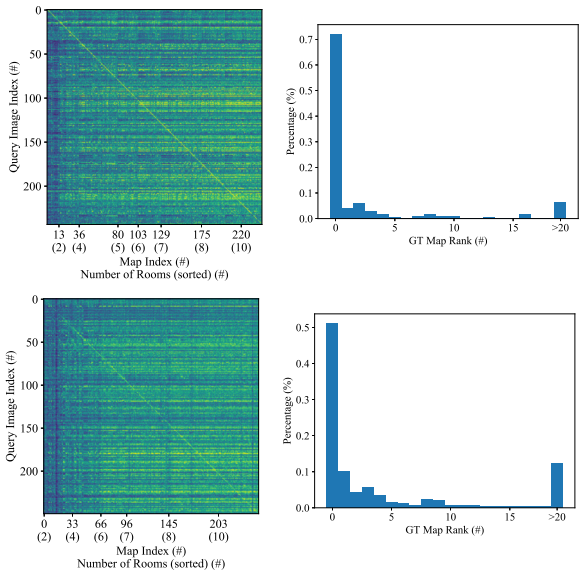


Figure 7. **Map retrieval.** (Top: ZInD) (Bottom: S3D) We randomly pick one panorama from each map, and exhaustively match the panoramas to all maps. The left affinity matrix record the maximum score of each posterior map for each matching, while the floor map are sorted with number of rooms in the map. The right histogram show the rank of GT floor maps.

to augment incident-angle codebooks (i.e. with same translations, similar distance observations can be achieved when rotations are sampled close to gt). Note we exclude the third group that has gt translation from the context loss, since the circular feature context defined in Eq.13 of the main paper is agnostic to rotation.

Rendering Algorithm. To render a circular feature of a camera pose hypothesis, we first project all the map points onto the circular feature segments, where the map point features are determined during the projection. We consider the closest map point for each segment as visible, and discard other invisible (i.e. occluded) map points. Based on the distance between the projected locations and the segment centers, we linearly interpolate the feature of each segment with the features from its two closest map points. For rendering fidelity, we first render a higher resolution circular feature with 64 segments, then average pool it to $V = 16$ segments.

1.9. Experiment Justifications.

ZInD vs S3D. We did most ablation experiments on the real-world ZInD dataset for its realistic house layouts and diverse photo capture locations, while S3D only has single capture per room and bias more to room center.

Shared codebook. In main paper Table.2, where we show performance when replacing the PointNet with a fixed shared codebook. In this experiment, we learn a shared

codebook for all map points and separate codebook offsets for each semantic label. For semantically-labelled map points, we add the offset codebook of its semantic label to the shared codebook as the map points' final codebook. This offset codebook scheme is for balancing the training progress due to the unbalanced number of semantically-labelled map points. With the presence of PointNet, the framework has a global scope that extracts a global descriptor of the map. The global descriptor is used implicitly by the PointNet to fine-tune different codebooks for individual map points. Thus the performance slightly drops without PointNet. However, a shared codebook may have its own preferred applications. For instance, a shared codebook can save memory when storing codebooks for each map point is memory-consuming with a large scale map.

Equirectangular vs Perspective. In main paper Fig.7(a), with same horizontal FoV, our models exhibits better translation accuracy with equirectangular images but better rotation accuracy with perspective images. This is because in equirectangular images, the 180° vertical-FoV captures the room layout edges of floor/ceiling that are good cues for learning better distance estimation. Similarly, under perspective projection, the rotation is well expressed in the slope of room layout edges (i.e. horizontal edges are flattened when yaw-angle rotation is aligned with walls).

MCL baseline. The MCL [1] with simulated ground-truth LiDAR input defines a strong baseline for non-semantic map input. For panorama, we simulate a 72-rays LiDAR. For perspective images, number of rays are reduced according to FoV (e.g. 18 rays for 90° FoV). We use the same likelihood model (i.e. Gaussian disturbance) as in [1]. With same amount of samples, our method (w/o semantics) performs similarly to MCL in recall, where their posterior maps are also similar as shown in Fig.4 in main paper.

LaLaLoc baseline. LaLaLoc [4] is a learning-based MCL framework for panorama localization. LaLaLoc renders and encodes a panorama layout depth map for individual camera hypothesis. LaLaLoc requires the query panorama to have known rotation, thus suffer less from symmetric ambiguities. To visualize the posterior map from non-grid sampling of LaLaLoc, we linear interpolate a convex polygon from its samples, resulted in a black triangle at left-bottom corner. With the high fidelity depth map rendering and CNN (i.e. ResNet18) encoder, the sampling process for LaLaLoc is very expensive as shown in Table.3 in the main paper.

PfNet baseline. PfNet [5] uses spatially transformed map under top-down view (i.e. bird's-eye view) to represent camera pose hypothesises. PfNet extracts feature maps for the query image and the sampled map images, where their likelihoods are estimated with CNNs taking input of stacked feature maps. Compared to LaLaLoc, PfNet has relatively light-weight rendering process and network architecture, resulted in its faster sampling rate. However, given PfNet framework was originally designed for sequential time updating (i.e. particle filter), it does not perform very nice with single query given the large domain difference (i.e. view-point) of its rendering (affine top-down view) to the observation (perspective front view).

References

[1] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 2, pages 1322–1328. IEEE, 1999. 4

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[3] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2007. 2

[4] Henry Howard-Jenkins, Jose-Raul Ruiz-Sarmiento, and Victor Adrian Prisacariu. Lalaloc: Latent layout localisation in dynamic, unvisited environments. *arXiv preprint arXiv:2104.09169*, 2021. 4

[5] Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *Conference on robot learning*, pages 169–178. PMLR, 2018. 4

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 3

[8] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3