LaLaLoc++: Supplementary Material

Henry Howard-Jenkins and Victor Adrian Prisacariu

Active Vision Laboratory, University of Oxford, UK
{henryhj,victor}@robots.ox.ac.uk

1 Introduction

This document provides further implementation details for LaLaLoc++ (Sec. 2), as well as some additional experiments and visualisations (Sec. 3).

2 Implementation

2.1 Floor Plan Localisation: Tacit Assumptions

Due to the characteristics of the datasets used, namely Structured3D [5] and the Zillow Indoor Dataset [1], there are some additional tacit assumptions about the structure of the indoor scenes within which localisation is performed.

Firstly, 3D room geometry is generated through extrusion of 2D polygons. Therefore, this results in all walls being vertical. In addition, it means that all floors and ceilings consist of a single horizontal plane each. Secondly, camera heights fall within a range of 1.4-1.8m in Structured3D, and within a range of 0.7-2.1m in the Zillow Indoor Dataset.

2.2 Training Schedule

On the Structured3D dataset, Φ_{plan} is trained for a total of 100 epochs through SGD. The batch size is set to 16. The initial learning rate is set to 0.05, with momentum at 0.9 and weight decay of 10^{-4}. The learning rate is scaled by a factor of 0.1 after 50 and 75 epochs.

Φ_{image} is trained for 200 epochs, again through SGD and with a batch size of 64. 0.1 is taken as the initial learning rate, and again with a momentum of 0.9 and weight decay of 10^{-4}. After 100 and 150 epochs the learning rate is scaled by a factor of 0.1.

When training on the Zillow Indoor Dataset, the total number of epochs and the learning rate decay milestones are each increased by a factor of $1.5\times$ to account for there being fewer training scenes.

2.3 Position Refinement

We perform bilinear position refinement using an Adam [2] optimiser. The initial learning rate is set to 0.01 and is scaled by a factor of 0.5 when the loss plateaus below a threshold of 0.05 and a patience of 10 iterations. The refinement is considered to have converged after 20 steps with a threshold on the reduction in cost set to 0.001, or after 150 steps have elapsed.

Table 1. Computational complexity for floor plan comprehension in each scene. We note the complexity of a single forward pass ("FLOPs/Item"), and the average number of passes required during training and testing for each of the methods ("#Items"), all per scene. The size of the floor plan and the number of testing layout inputs are each computed as the median over Structured3D val split

Method	FLOPs/Item	Training		Testing	
		# Items	Total FLOPs	# Items	Total FLOPs
LaLaLoc (0.5m)	1.1G	21	23.9G	332	378.5G
LaLaLoc++	40.0G	1	40.0G	1	40.0G

Table 2. Generalisation of Φ_{plan} moving from synthetic (Structured3D) to real (Zillow Indoor Dataset). Both are evaluated on the Zillow Indoor Dataset

Φ_{plan}	Localisation Accuracy				
Train Dataset	Med.	<1cm	<5cm	<10cm	<1m
Structured3D	9.9	2.0%	29.1%	50.2%	79.1%
Zillow Indoor	9.3	2.5%	32.1%	51.7%	80.7%

3 Additional Experiments and Visualisation

3.1 Computational Complexity

In Sec. 5.1 of the main paper, we noted that LaLaLoc required many forward passes of its Φ_{layout} to form the reference grid of layout embeddings. Here, we quantify this by computing the complexity of LaLaLoc++'s Φ_{plan} against Φ_{layout} from LaLaLoc, with results listed in Tab. 1. Although a single pass of Φ_{plan} is much more complex than a single pass of Φ_{layout} , Φ_{layout} requires many forward passes to compute descriptors during testing, whereas Φ_{plan} only requires a single pass for each scene. This results in a very significant computational saving.

3.2 Generalisation Between Datasets

We explore the generalisation of Φ_{plan} across the synthetic-to-real domain gap in Tab. 2. We take Φ_{plan} trained on Structured3D and apply it to the Zillow Indoor Dataset, Φ_{image} is retrained from scratch on the new dataset. The resulting model is able to achieve near parity in its accuracy. This demonstrates that the Φ_{plan} is able to generalise well to the more complex scene geometry present in the Zillow Indoor Dataset. However, we found poor performance in experiments where both modules are kept frozen across the datasets. In addition to the challenges of synthetic-to-real, the scene data representation is not completely similar in each dataset, resulting in a difference in the rasterisation of input 2D floor plans. We hypothesise that this is a significant contributor to the performance drop.

Table 3. Comparison of the image embedding architecture that is used. R50 + Transformer is the proposed formulation where the image feature map is refined with a transformer encoder For further comparison, we also include some panorama-specific backbones, HoHoNet and SliceNet

			Localisation Accuracy			
Φ_{image} Backbone	# Params.	Med.	<1cm	<5cm	<10cm	<1m
R50 + Transformer	24.1M	5.2	**5.4%**	48.8%	72.3%	92.3%
SliceNet Encoder	69.2M	4.7	**5.4%**	52.0%	73.7%	92.6%
HoHoNet Encoder	31.4M	**4.6**	**5.4%**	**54.1%**	**75.5%**	**93.2%**

3.3 Image Module and Alternative Backbones.

In this section, we explore how the panorama-specific network designs can be used to further improve localisation accuracy with LaLaLoc++. To do so, we compare Φ_{image} formulations with backbones inspired by recent work on layout and monocular depth estimation from panorama images, specifically HoHoNet [4] and SliceNet [3]. These networks both work similarly to exploit the structure of gravity-aligned panoramas by operating mostly in the vertical dimension alone. This produces features that have been flattened vertically, but maintain the input horizontal resolution.

For the HoHoNet-based Φ_{image} , we extract the panorama's Latent Horizontal Features with dimension (128×512), *i.e.* with a 128d descriptor for each horizontal column in the original input panorama. We then compute a single 128d embedding with a 2-layer MLP operating across the horizontal axis. We use the ResNet34 variant of the network as this was reported to outperform other variations for the task of layout estimation. For the SliceNet-based Φ_{image} , we use the ResNet50 variant. Image features are taken as the output of the LSTM module and have dimension (1024×512), representing 512 vertical slices each 1024d. We again employ a 2-layer MLP across the horizontal axis to produce a single 1024d descriptor, which is linearly projected to 128d.

Results are listed in Tab. 3, where it can be seen that LaLaLoc++ can achieve some further gain in accuracy by leveraging these backbones that more explicitly exploit the characteristics of gravity-aligned panoramas.

3.4 Failure Modes

We provide qualitative visualisation of localisation failures with LaLaLoc++ in Fig. 1. These instances of failed localisation seem to generally fall into two main modes. In the first mode, the pose is predicted to be in an incorrect room, this is an example of layout ambiguity across the floor plan. However, in these scenarios the alignment does frequently still appear plausible. In the second mode of failure, the pose is predicted to be in the correct room, but the layout is then misaligned. By visualising these cases, we see that room edges and corners are often mistakenly aligned to the edges of room clutter, such as tables and counter-tops.

Fig. 1. Qualitative visualisation of failure modes on the Structured3D dataset. *Left:* failures where the wrong room is retrieved entirely. *Right:* misalignment of the layout to the image.

3.5 Layout Decoder Output

In this section, we provide additional visualisation of the layout decoder output. Figure 2 illustrates the 2D to 3D hallucination that is learned by LaLaLoc++ during the floor plan embedding training stage. It can be seen that the layouts captured by the floor plan encoder generally reflect the structure well. The wall/floor or ceiling boundary seems to be the largest source of error in many cases, as is seen in the top three rows. This is to be expected as LaLaLoc++ has to learn a general prior for heights from 2D plans.

It is also apparent that some finer layout detail is lost, but the prediction is still very representative of the overall structure, as depicted in the bottom three rows. Interestingly, when considered with the saliency plots in Fig. 5 from the main paper, the type of structure that is missed in decoding still contributes to the descriptor as evidenced by its saliency. This would suggest that this type of error is due to the limited decoder design. However, we emphasise that this paper is not targeting room layout estimation. This layout decoder offers a simple inductive bias and its output is only computed during training of the embedding space. These results only serve to improve intuition about how LaLaLoc++ infers 3D structure.

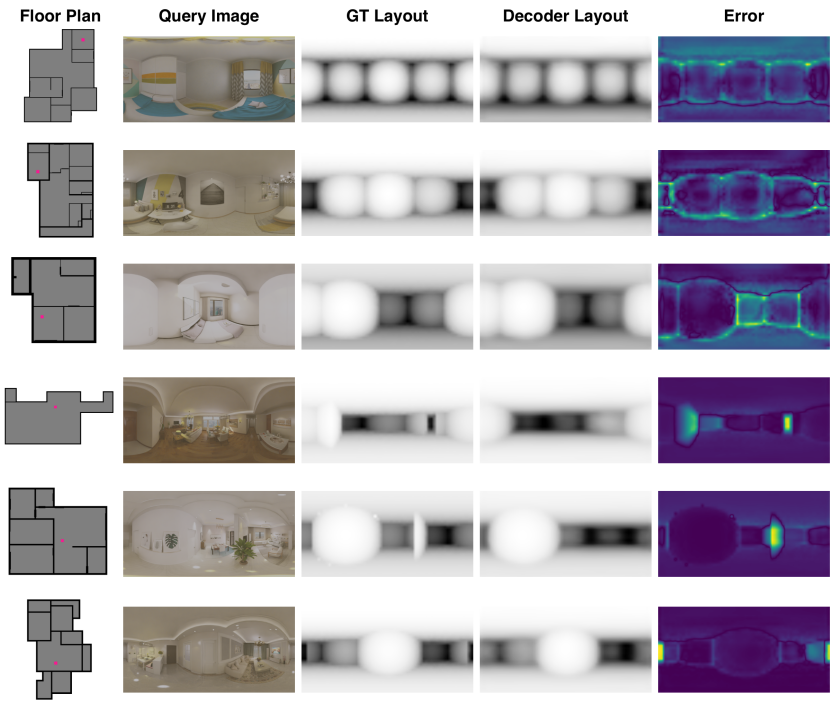

Fig. 2. Qualitative layout predictions from the decoder, helping to illustrate the 2D to 3D hallucination of layout. *Left to right:* floor plan with query location marked in pink; query image; ground truth layout; decoded layout; L1 error. Performed on the Structured3D dataset.

References

1. Cruz, S., Hutchcroft, W., Li, Y., Khosravan, N., Boyadzhiev, I., Kang, S.B.: Zillow indoor dataset: Annotated floor plans with 360deg panoramas and 3d room layouts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2133–2143 (2021)
2. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
3. Pintore, G., Agus, M., Almansa, E., Schneider, J., Gobbetti, E.: Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11536–11545 (2021)
4. Sun, C., Sun, M., Chen, H.T.: Hohonet: 360 indoor holistic understanding with latent horizontal features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2573–2582 (2021)
5. Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., Zhou, Z.: Structured3d: A large photo-realistic dataset for structured 3d modeling. arXiv preprint arXiv:1908.00222 **2**(7) (2019)