LaLaLoc++: Global Floor Plan Comprehension for Layout Localisation in Unvisited Environments

Henry Howard-Jenkins and Victor Adrian Prisacariu

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

Abstract. We present LaLaLoc++, a method for floor plan localisation in unvisited environments through latent representations of room layout. We perform localisation by aligning room layout inferred from a panorama image with the floor plan of a scene. To process a floor plan prior, previous methods required that the plan first be used to construct an explicit 3D representation of the scene. This process requires that assumptions be made about the scene geometry and can result in expensive steps becoming necessary, such as rendering. LaLaLoc++ instead introduces a global floor plan comprehension module that is able to efficiently infer structure densely and directly from the 2D plan, removing any need for explicit modelling or rendering. On the Structured3D dataset this module alone improves localisation accuracy by more than 31%, all while increasing throughput by an order of magnitude. Combined with the further addition of a transformer-based panorama embedding module, LaLaLoc++ improves accuracy over earlier methods by more than 37% with dramatically faster inference.

1 Introduction

Floor plans are ubiquitous in the built, indoor environment. For almost any modern building, these structural plans are stored by local government, builders, real estate agents and even as fire-safety maps and other guides to indoor environments. The prevalence of these documents across domains is not an historical fluke, but instead because they have a number of very useful characteristics. *Permanence*, while furniture and objects present within an environment may move or change, the structure represented in a floor plan remains the same, and therefore these documents are able to provide a description of the environment over a period of years and decades. *Expressiveness*, the structure depicted in a floor plan gives the reader a good basis for understanding that particular indoor environment: in a fire safety map, one is able to bridge the gap between their egocentric view and the plan to determine an exit route; when viewing a property online, one is able to imagine living in the space illustrated by the top-down plan. *Convenience*, these blueprints are extremely lightweight format for expressing this structure, allowing them to be easily stored and interpreted across physical and digital formats.

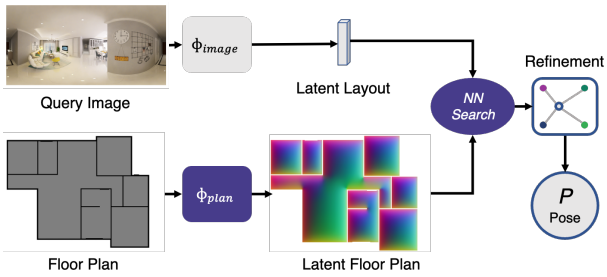

Fig. 1. An overview of our proposed method for floor plan localisation. LaLaLoc++ takes a 2D floor plan and infers 3D structure across the free space of the plan. To localise, an image branch infers layout structure from the query image. Position is predicted by aligning the image's latent structure to the globally inferred structure.

It is for these reasons that, in this paper, we turn our attention to direct interpretation of structural blueprints, specifically in the context of floor plan localisation. Visual relocalisation estimates the pose of a camera given some prior of an environment. Commonly in localisation methods, the prior takes the form of a built 3D structural point-map, *e.g.* [24], a series of stored images, *e.g.* [1,2], or in the weights in a neural network, *e.g.* [4]. Floor plan localisation, however, takes an architectural plan of the indoor environment as prior.

We present LaLaLoc++ to perform floor plan localisation directly from 2D plans. Previous methods for floor plan localisation relied on translation of the architectural floor plan into an explicit 3D geometry, either relying on the assumption of known [15], or manually-estimated [9] ceiling and camera heights to extrude the plan into 3D. LaLaLoc++, however, learns a general prior over 2D floor plans which allows us to implicitly hallucinate 3D structure across the entire plan in a single pass, and without the need for the intermediate conversion into an explicit 3D representation. This is depicted in Fig. 1.

The contributions of this paper can be summarised as follows: (*i*) We present a floor plan comprehension module that learns to implicitly hallucinate 3D visible structure from 2D floor plans. This removes the need for explicit extrusion with an assumed or known camera height. (*ii*) We demonstrate that, when combined with an improved RGB panorama embedding module, this method is a significant advance over previous work, both in terms of accuracy (over 37% and 29% improvement over LaLaLoc on Structured3D [34] and Zillow Indoor Dataset [9], respectively) and speed (35x speed up).

2 Related Work

Camera relocalisation has seen a vast collection of research progress, inspiring a diverse set of problem settings and methods to solve them. Generally, visual localisation seeks to make associations between query image and prior knowledge of the environment in order to predict the capture location. However, the type and construction requirements of this prior offers a variety of design choices.

A large portion of methods offer prior construction that is generalisable across scenes. Structure-based methods [20,19,24] establish correspondences by matching points between the query image and an explicit 3D representation built using a method such as SfM [13,26]. Retrieval-based methods [25,2,1,11] primarily estimate pose by matching the query image to its most similar counterpart in a database. On the other hand, other methods require that a new prior be learned for each environment in which localisation is performed, usually learning an implicit representation of the scene within neural network weights. Absolute pose regression methods [17,16,8] train a deep neural network to directly regress the camera pose. Scene-coordinate regression methods [27,30,4,5,6] densely regress global 3D coordinates across image locations. These generalisable and scene-specific methods share a common attribute, which is that, although the form their prior takes differs, they require a training sequence of images in order to construct or learn it. In this paper, however, we approach the task of floor plan localisation, which takes only a floor plan as its prior. Therefore, the need for a training set of images to enable localisation in a new environment is alleviated.

Like other forms of visual localisation, floor plan-based methods can take many forms. The task has seen particular interest in the context of robotics, where methods operate by forming a room layout observation model between a location in the floor plan and query by extracting layout edges [3,32,29], layout corners [14] or aggregating depth [33] from which an observation likelihood can be estimated. Further, if depth information is available, scan-matching techniques [22] estimate an alignment between the observed depth and the floor plan. These methods are generally not standalone, instead relying on multiple sequential measurements used as hypothesis weightings for a Monte Carlo Localisation [10] frameworks. In this paper, however, we approach the task as defined by LaLaLoc [15]. This localisation formulation, unlike previous works, aims to localise a panorama image within a floor plan without depth information, motion or time-coherency cues, and without assuming a good initialisation. Concurrent to this work, Min *et al.* [21] also perform floor plan localisation in the same vein. Cruz *et al.* [9] construct a similar layout-based localisation task for panorama registration, however, they aim to localise with respect to other panoramas, rather than in an unvisited location.

LaLaLoc [15] learns a latent space to capture room layout [18]. Localisation is performed as retrieval between a database of embedded layouts from sparsely sampled poses within the floor plan and the embedding of the panorama image. This is followed by a gradient-based optimisation of pose to minimise the distance between layout and query embeddings, enabled through differentiable rendering. However, the model of the scene used by LaLaLoc had to be created by extruding the floor plan in the z-dimension, assuming a known camera and ceiling height. In addition, LaLaLoc's formulation requiring rendering across the plan for retrieval, and multiple (differentiable) rendering steps for pose refinement is slow. LASER [21] addressed these limitations by rendering embeddings individually using a codebook scheme. In this work, we instead propose dense and global structural comprehension of the 2D plan. In doing so, LaLaLoc++

is able to dramatically improve accuracy and inference speed over LaLaLoc, all while removing the need for height assumptions at inference time.

3 Method

LaLaLoc++ consists of two parallel branches which operate across data modalities. An overview of its architecture is depicted in Fig. 1. The floor plan comprehension module, Φ_{plan} , infers the layout structure that is visible across a 2D floor plan, computing a latent vector at every location in the $x - y$ plane that covers the scene. The image embedding module, Φ_{image} , on the other hand, aims to capture the layout that is visible within a cluttered, RGB panorama image. They map to a shared latent space such that a panorama captured at a location and embedded by Φ_{image} should have the same latent vector as that location sampled from Φ_{plan} 's output. Therefore, LaLaLoc++ performs localisation of a query panorama by finding the location in the floor plan that most accurately matches the visible room layout, with matching cost being computed in the shared layout latent space. In the following, we detail the task, the form that our architecture takes, how it is used to localise, and how it is trained.

3.1 Floor Plan Localisation

We follow the floor plan localisation task as used in LaLaLoc [15]. This is the task of performing localisation of a panorama within a scene which has not been visited previously, given only a floor plan as prior. Specifically, we predict the 2 DoF camera pose to an $x - y$ location in the 2D floor plan assuming a known orientation, an assumption we explore in Sec. 5.6. We also detail some tacit assumptions arising from dataset characteristics in the supplementary material. The floor plan consists of a 2D image which denotes an empty scene consisting of only walls, floors and ceilings.

3.2 Φ_{plan} : Global Latent Floor Plan

Given a 2D floor plan of an environment, we wish to infer the architectural structure of the scene at any given location within it. In order to achieve a similar objective, LaLaLoc [15] formed an explicit representation of the scene in the form of a mesh using a known ceiling height. This explicit geometry was then rendered as a depth image at a given location from which the implicit layout representation could be computed by LaLaLoc's Φ_{layout} . While shown to be effective, this method proves to be slow, requiring multiple rendering steps and forward passes in order to form only a sparse map of layout across the floor plan, and also requires that camera and ceiling heights are known.

Instead, with LaLaLoc++ we propose that the implicit representation of layout can be inferred directly from the floor plan itself. This formulation allows us to walk well-trodden ground in architecture design for fast, efficient, and *dense* inference of layout across the floor plan.

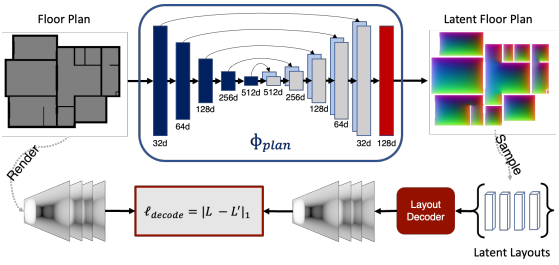

Fig. 2. Φ_{plan} , the floor plan comprehension network and its training routine. The network takes as input a 2D plan of the scene and outputs a dense grid of features which capture the 3D structure visible at their respective locations. Feature maps are depicted with spatial dimensions unrolled for simplicity, and channel dimensions are listed.

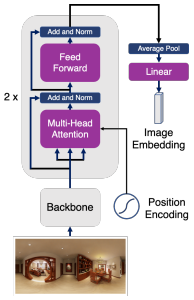

Fig. 3. Φ_{image} , the transformer-based image embedding module. A ResNet backbone is used and its output is fed into a transformer encoder architecture. The resulting features are pooled and projected to get the final embedding.

Specifically, Φ_{plan} takes the form of a UNet [23] inspired encoder-decoder structure. It takes as input a 2D structural floor plan and outputs a dense feature map. The vector at each location is a latent representation of the layout visible there. Therefore, in just a single forward pass we are able to build a dense expression of layout within a plan. This structure is depicted at the top of Fig. 2.

3.3 Φ_{image} : Image Embedding with Long-range Attention

While forming an expressive latent space for layout inferred from a floor plan was shown to be readily achievable to a high degree of effectiveness in LaLaLoc [15], the difficulty of (implicitly) predicting layout from RGB panoramas proved to be a significant source of error. We therefore propose an improved image embedding module based on a transformer encoder [31]. This module allows Φ_{image} to better capture long-range structural patterns present in the query panorama image.

We implement a transformer encoder to refine the feature-map computed by the image branch backbone, which is a ResNet50 network [12]. The features outputted by the backbone are linearly projected from 2048d to 128d. These projected features are then fed into the transformer encoder, with 2 encoder blocks before finally being pooled and projected to form the image embedding for a panorama. The multi-headed self-attention present in the encoder blocks encourages long-range attention. This architecture is visualised in Fig. 3.

3.4 Localisation

At inference time, a dense reference grid of latent layout vectors are computed by passing the 2D floor plan. An initial estimate of position is computed as the location corresponding to the nearest latent layout in this grid. However, since this is still a spatially discretised grid across the scene, we introduce further

alignment in the form of pose optimisation allowing for a continuous estimate of position. We describe this more concretely in the following.

We denote the floor plan encoded by Φ_{plan} as $\mathbf{G}$, and the query latent layout computed by Φ_{image} as f. The initial position estimate, p_0, is taken as whole pixel location, *i.e.*$p_0 = (i_0, j_0) \in \mathbb{Z}^2$, with the lowest matching cost $|f - \mathbf{G}[p_0]|_2$, where $[\cdot]$ is a look-up function. This location is used to initialise the refinement.

We then perform the continuous refinement of pose through interpolation of the local neighbourhood of latent layout representations in the encoded floor plan. Position is initialised at the nearest location in the feature grid. During refinement, the position estimate is relaxed to allow sub-pixel locations, $p_r \in \mathbb{R}^2$, and $[\cdot]$ is extended to use sub-pixel interpolation. The refined pose is iteratively updated to minimise the matching cost in a gradient-based optimisation scheme,

$$\min_{p_r} |f - \mathbf{G}[p_r]|_2. \tag{1}$$

3.5 Training

LaLaLoc++ follows an analogous training philosophy to that proposed in LaLaLoc [15] where the network is trained in two separate stages. In the first stage, we train Φ_{plan} alone. Since Φ_{plan} takes a form of explicit structure as input, its training is used to establish an expressive latent space for capturing the room layout at any given location in the scene. This latent space is then frozen and, in the second stage, Φ_{image} is trained to map images to this fixed representation space.

Φ_{plan} : Learning the Latent Space. We train Φ_{plan} with a single loss, where the layout embedding at a given location is passed through a simple layout depth prediction decoder and compared with a rendered layout at the same location:

$$\ell_{decode} = |L' - L|_1, \tag{2}$$

where L' is the predicted layout, and L is the ground-truth rendering at a given location. During training, we use known camera and ceiling heights for the rendering of the target layout, however, the floor plan comprehension module is forced to infer the structure without these parameters. Therefore, it must learn a generalised prior for the hallucination of 3D structure from the top-down 2D input. Illustration of how this loss is computed can be found in Fig. 2.

Φ_{image} : Mapping to the Latent Space. The training of Φ_{image} takes the form of a simple strategy, which reflects its role in the localisation method. Given a training batch of panorama images, their respective poses, and a floor plan, the global latent floor plan is computed by Φ_{plan} and sampled at the panorama poses. The panorama images are then embedded by Φ_{image} , and a loss is applied to the difference between the embeddings produced by Φ_{image} , f, and those sampled from Φ_{plan} 's output, g, which is defined as follows:

$$\ell_{L2} = |f - g|_2. \tag{3}$$

4 Implementation Details

Here, we describe the specific nature of the task evaluated in the experiments section of the paper. We start with the datasets, followed by the configuration of our method. Additional details can be found in the supplementary material.

4.1 Datasets

The bulk of evaluation experiments are performed on the Structured3D dataset [34], which consists of 3,500 photorealisticly-rendered synthetic indoor scenes. We follow the split of data used in LaLaLoc [15], which itself follows the dataset's predefined split with some scenes excluded due to corrupted data. This leaves 2979/246/249 scenes for training/validation/testing, respectively. Every scene comprises of multiple rooms, each containing a single panorama image which is used as a query. Since there are no reference images captured from within the same scene, we label each scene as *unvisited*, as there is no RGB data that can be leveraged for training or database building and therefore the method must rely on floor plan data alone. There are three furniture configurations, empty, simple and full, as well as three lighting configurations for each scene, cold, warm and raw. During training, we randomly sample a furniture and lighting configuration at each iteration. At test time, we perform evaluation in the full and warm setting. This is the most difficult furniture configuration as there are the most possible distractors from layout present in the image.

We also explore how this method extends to real data by performing evaluation on the recently released Zillow Indoor Dataset [9]. We processes the dataset to match the problem setting on the Structured3D dataset, namely considering localisation across a single floor and alignment of panorama images. This results in 1957/244/252 scenes for training/validation/testing, respectively. However, the dataset only provides a single lighting and furniture configuration for each panorama. The remaining implementation details apply across both Structured3D and the Zillow Indoor Dataset.

Floor plans. We represent the 2D floor plans as image. Each pixel location can take three possible values: 1 represents a wall, 0.5 is used to denote free-space within the floor plan, and 0 is assigned to free space outside of the floor plan. To ensure that absolute scale is preserved in the creation of these plans, we use a consistent scale across all scenes in the dataset. It is set such that one image pixel covers 40mm of space in reality. This value was set heuristically as a balance between keeping the overall size of the floor plan manageable, and ensuring that all walls are still discernible. At the most compact representation in Φ_{plan} , the spatial resolution of the feature map is 1/32th of the original height and width, we zero-pad the floor plans so that both height and width are divisible by 32.

4.2 Network Architecture

As depicted in Fig. 2, Φ_{plan} consists of 5 down-sample and 5 up-sample layers. These layers each consist of a convolutional block which contains a 3x3 convolu-

tion, a batch-norm and a ReLU activation repeated twice. In the down-sample block, the convolutional block is followed by a 2x2 max-pooling layer. In the up-sample block it is preceded by a 2x2 bilinear up-sampling of the feature map. This up-sampled feature map is then concatenated with the feature map from the corresponding down-block (taken before pooling), to form skip connections. The output of the final up-sampling layer is passed through a final 3x3 linear convolution to form the final latent floor plan. This latent floor plan keeps the same spatial resolution as the original input floor plan with a 128d descriptor for each location which are normalised. The layout decoder used in training takes an identical structure to that used in [15].

Φ_{image} takes the form of a ResNet50 backbone. In its simplest formulation, the feature map outputted by this backbone is pooled to a single vector, then linearly projected to 128d and normalised. However, in our transformer formulation, a transformer encoder forms an intermediate refinement of the image feature map before pooling and projecting. More specifically, a 2D sinusoidal positional encoding [31] is generated for each spatial location in the feature map. The feature map is then flattened, linearly projected to 128d, and fed into a vanilla transformer encoder [31] with two repeated blocks. There are 8 attention heads, and the feed-forward hidden dimension is set to 256 dimensions. We apply the positional encoding at the attention layers, rather than input, following [7]. The encoded features are then pooled and projected once more before being normalised to form the image embedding.

5 Experiments

In this section, we perform experimental evaluation of LaLaLoc++. Initially, we explore the performance of the method as a whole before exploring the constituent components and their impact on the final accuracy. Finally, we perform ablation experiments to validate the design of these components. Unless otherwise stated, experiments are performed on the Structured3D [34] dataset.

5.1 Floor Plan Localisation

We first perform localisation on the test split of the Structured3D [34] dataset. We compare the performance of LaLaLoc++ to three other methods, LaLaLoc

Table 1. Localisation performance of LaLaLoc++ compared against three other floor plan localisation methods on Structured3D

Method	Localisation Accuracy				
	Median (cm)	<1cm	<5cm	<10cm	<1m
ICP	21.8	**9.5%**	26.4%	35.6%	68.5%
HorizonNet [28] + Loc	9.1	3.3%	29.3%	53.4%	77.4%
LaLaLoc [15]	8.3	3.6%	32.0%	58.0%	87.5%
LaLaLoc++	**5.2**	5.4%	**48.8%**	**72.3%**	**92.3%**

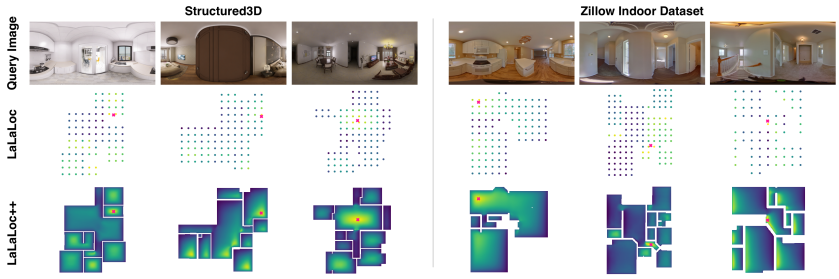

Fig. 4. Qualitative comparison of the floor plan comprehension differences between LaLaLoc++ and LaLaLoc. Visualised as a plot of the latent distance between the query image and embeddings across the floor plan in each of the two methods. While LaLaLoc++'s Φ_{plan} is able to infer layout densely within the scene, LaLaLoc instead just has a sparse sampling (0.5m in each direction). The difference in the resulting understanding of the environment is stark.

Method	Localisation Accuracy				
	Med.	<1cm	<5cm	<10cm	<1m
LaLaLoc	13.1	1.3%	19.9%	40.9%	77.6%
LaLaLoc++	**9.3**	**2.5%**	**32.1%**	**51.7%**	**80.7%**

Table 2. Localisation performance of LaLaLoc++ compared to LaLaLoc with real data on the Zillow Indoor Dataset.

Method	Inference Time (s)		
	Embedding & Retrieval	Refinement (Rendering)	Total
2D-ICP	-	-	20.1
LaLaLoc w/o VDR	0.05	2.33 (0.85)	2.38
LaLaLoc	0.05	4.60 (3.34)	4.65
LaLaLoc++	**0.01**	**0.12**	**0.13**

Table 3. Inference time comparison on a single Nvidia Titan RTX GPU. Times are displayed as: Total time (Render time).

itself, as well as the 2D-ICP and HorizonNet [28]-based baseline presented in LaLaLoc [15]. Results are listed in Tab. 1.

As can be seen in Tab. 1, LaLaLoc++ is able to localise in these unseen environments significantly more accurately than the original LaLaLoc. The median localisation accuracy represents a 37% improvement over LaLaLoc. When considering the accuracy thresholds, it is notable that LaLaLoc++ not only is able to localise more images to the fine-grained accuracy thresholds, but is also able to localise more images to within 1m (87.5% *v.s.* 92.3%). This suggests that LaLaLoc++ reduces the frequency of catastrophic failure to localise. We visualise some failure modes in the supplementary material.

We further investigate how this performance extends beyond synthetic data into the real world captures on the Zillow Indoor Dataset [9]. We report these results in Tab. 2. Inline with results on Structured3D, LaLaLoc++ offers significantly improved accuracy over LaLaLoc (29% increase). The real world layout distributions appear to pose a harder challenge, with more complex floor plan geometry. However, it does show that the method can extend to real data.

Figure 4 illustrates the contrast in prior-forming between LaLaLoc and LaLaLoc++. The dense prediction of Φ_{plan} is able to far more expressively capture the layout across the floor plan, whereas the sparse strategy to form reference embeddings

Table 4. Contribution of latent floor plan training to the final localisation accuracy. Identical image branch architecture is used in each scenario. *Mimic* is used to refer to a training scheme where Φ_{plan} is trained to copy LaLaLoc's layout branch output

	Localisation Accuracy				
Method	Med.	<1cm	<5cm	<10cm	<1m
LaLaLoc	8.3	3.6%	32.0%	58.0%	87.5%
Mimic	8.4	1.8%	27.3%	56.8%	88.5%
Decoder Loss	**5.7**	**4.9%**	**45.0%**	**69.4%**	**92.0%**

in LaLaLoc is comparatively low in fidelity. Although LaLaLoc could sample reference poses more densely, inferring at the same level of density with LaLaLoc quickly become intractable.

In Tab. 3, we compare the time complexities of LaLaLoc++ against LaLaLoc. LaLaLoc++ offers more than a 35x speed up over LaLaLoc. This is predominantly for two reasons. Firstly, LaLaLoc++'s direct inference from 2D plans means that expensive rendering steps can be removed. Notably, this also includes the expensive backwards passes in the differentiable rendering optimisation scheme. Secondly, LaLaLoc required a forward pass through a ResNet18 [12] for each rendered layout. Although this could be performed in batches, depending on the density of sampling the total number of items could reach hundreds. On the other hand, LaLaLoc++ allows dense inference of structure in a single forward pass, which proves to be significantly faster.

5.2 Global Floor Plan Comprehension

To evaluate the contribution of our proposed global floor plan processing, we perform localisation using various configurations of Φ_{plan} all with equivalent image branch architectures. The image branch architecture is chosen to be identical to the one proposed in LaLaLoc. The first comparison is to LaLaLoc's layout branch and its scheme of sparse floor plan rendering and embedding. We then train our proposed global floor plan module as a direct substitution for the layout embedder in LaLaLoc. Therefore, the floor plan network is trained to mimic the output of LaLaLoc's layout branch, and the original image branch is used without any modification or fine-tuning. We call this "Mimic". Finally, we compare to our proposed method, where the Φ_{plan} is trained with the layout decoding loss, is frozen, and then Φ_{image} is trained to map panoramas to the learned space.

Results for these experiments can be found in Tab. 4. Worth considering first is the results for the "Mimic" configuration. This experimental setup can be considered a test of whether 3D structure can be directly inferred from the 2D layout to a similar degree of expressiveness as the original LaLaLoc's render and embed methodology. The results show that the module is in fact able to nearly match the original performance of the network for which it is substituting. This is particularly notable as the 2D comprehension module is not given height

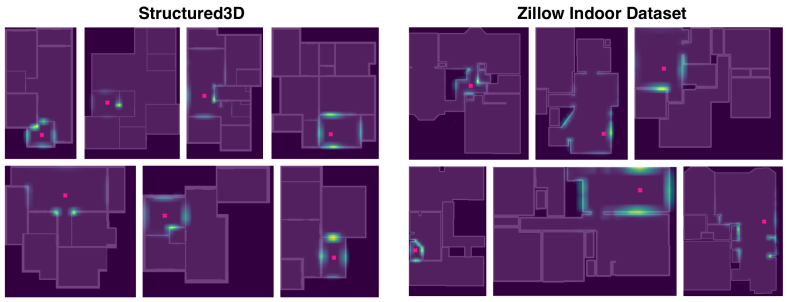

Fig. 5. Investigation into floor plan saliency for inferring layout at a given location from a global floor plan. The query location is marked with a pink cross, and computed saliency is overlaid onto the floor plan. The floor plan comprehension module demonstrably learns to differentiate between visible and invisible structure for the query locations when computing their descriptor.

information (neither ceiling nor camera) and instead must learn a generalised 2D to 3D room structure prior which holds for these unseen environments. As such, there is an inherent degree of inaccuracy expected from this formulation as it must replicate a method while dealing with significantly more ambiguity.

When trained to form its own layout latent space in the "Decoder Loss" setting, the performance improvement over LaLaLoc is dramatic. This is likely because the latent space for layouts can be constructed in a way compatible with the structure that can be determined from a 2D floor plan. In the supplementary material, to aid in the intuition of the learned space, we visualise the output of the decoder. In addition, the dense estimation of layout may provide a better setting for pose refinement than the Latent Pose Optimisation proposed in LaLaLoc, which relies on differentiable rendering. We investigate this in Sec. 5.3.

Floor Plan Saliency. In the explicit rendering formulation used in LaLaLoc, the structure rendered at any given point in the scene has the same level of context as the query image with which you wish to compare it to, *i.e.* only the visible structure from that location. However, when processing the entire floor plan, the receptive field of the network for structural inference at any given location quickly moves beyond what is strictly visible from that location. If global information is leveraged to form the layout descriptor from the plan, it is likely that the image branch will not be able to map to the same latent space.

To investigate this, we plot saliency of floor plan structure in computation of the descriptor at any given location. The saliency is computed by masking a sliding window across the floor plan such that any walls in this window are instead assigned as free-space. We compare how a descriptor at a chosen location in the plan changes against the original from an unmasked plan via L2 distance. This distance is assigned to the centre point of the mask to form a saliency map.

Table 5. Exploration of the Latent Floor plan resolution on localisation. *Init.* refers to the median accuracy of the initialisation for refinement, reported in cm

Floor plan Subsampling	Init.	Med.	Localisation Accuracy <1cm	<5cm	<10cm	<1m
LaLaLoc LPO	22.5	10.5	2.0%	27.4%	48.4%	87.5%
LaLaLoc	11.0	8.3	3.6%	32.0%	58.0%	87.5%
8x	28.4	6.5	4.7%	41.3%	64.3%	87.2%
4x	15.1	5.3	5.7%	47.7%	71.4%	92.1%
2x	8.8	5.3	5.3%	48.0%	69.9%	92.1%
Full-size	6.1	5.2	5.4%	48.8%	72.3%	92.3%

Some examples of these saliency maps are plotted in Fig. 5. It is apparent that the floor plan module does not unduly exploit global structure, but instead the structure local to the test point is the most salient regions of the plan.

5.3 Sub-pixel Refinement

In this section, we wish to investigate the performance properties of the proposed sub-pixel floor plan optimisation process. We perform refinement initialised with retrieval results from increasingly sparse representations of the floor plan. In each of these experiments the floor plan is embedded at full-resolution, but the resulting feature maps are sub-sampled by factors of 2. This has the effect of testing the refinement performance with wider baseline initialisation. For each level of sub-sampling, we re-train the image branch. We also include results for LaLaLoc in its operation mode where the latent pose optimisation is the only pose refinement after retrieval, thus skipping the local retrieval normally performed, and in the configuration with local re-sampling and continuous refinement.

The results for this are listed in Tab. 5. The median error of the refinement initialisation is included for each method and formulation. It can be seen that LaLaLoc++, and specifically its continuous pose refinement through interpolation, is able to recover extremely well from increasingly poor initialisation. Even subsampling the latent floor plan by 8x, LaLaLoc++ still outperforms LaLaLoc by 1.8cm, despite the initialisation being nearly 3x worse.

5.4 Image Transformer

We show a localisation performance comparison between the image embedding structure consisting of a simple feature backbone, pooling and projection and our proposed transformer formulation. Results are listed in Tab. 6. As can be seen, the transformer leads to a nearly 9% improvement in localisation accuracy over the baseline. This result suggests that the long-range attentional mechanisms introduced by the transformer encoder allow the embedding module to better capture layout from the panorama image.

Table 6. Comparison of the image embedding architecture that is used. ResNet50 denotes a simple backbone followed by average pooling and a linear projection. R50 + Transformer is the proposed formulation where the image feature map is refined with a transformer encoder

Method	Localisation Accuracy				
	Med.	<1cm	<5cm	<10cm	<1m
ResNet50	5.7	4.9%	45.0%	69.4%	92.0%
R50 + Transformer	**5.2**	**5.4%**	**48.8%**	**72.3%**	**92.3%**

Φ_{plan}	Localisation Accuracy				
# Layers	Med.	<1cm	<5cm	<10cm	<1m
3	12.9	2.5%	27.4%	44.8%	70.0%
4	6.0	4.0%	44.0%	65.4%	90.5%
5*	**5.2**	**5.4%**	**48.8%**	**72.3%**	**92.3%**
6	5.3	5.1%	48.0%	70.4%	92.1%

Table 7. Localisation with differing numbers of down and up-sampling layers in Φ_{plan} . * denotes the chosen configuration.

Φ_{image}	Localisation Accuracy				
# Blocks	Med.	<1cm	<5cm	<10cm	<1m
1	5.9	3.7%	43.5%	69.2%	**92.3%**
2*	**5.2**	5.4%	**48.8%**	**72.3%**	**92.3%**
3	5.6	**5.7%**	45.2%	68.7%	92.0%
4	5.7	4.8%	45.1%	69.3%	92.2%

Table 8. Localisation as the number of transformer encoder blocks in Φ_{image} is varied. * denotes the chosen configuration.

5.5 Ablation Experiments

We validate LaLaLoc++'s architecture through the ablation experiments discussed here. First, we see how the localisation accuracy is impacted by the number of down and up-sample layers in Φ_{plan} . We then vary the number number of encoder blocks used in Φ_{image} . Results for each of these experiments are listed in Tabs. 7 and 8, respectively. In Tab. 8 it is notable that, although the chosen configuration performs almost identically to the configuration with 6 down-sample and up-sample layers, the chosen configuration has significantly fewer trainable parameters (24.1M *vs.* 63.3M). Therefore, the configuration with 5 down-sample and up-sample layers is the most appropriate.

5.6 Rotational Ambiguity

In this section, we conduct experiments extending LaLaLoc++ to also predict orientation of the query panorama.

To estimate the unknown orientation of the query panorama, we rectify the image using the prepossessing step as described in [35]. Through vanishing point detection, this estimates a semi-canonical alignment of the panorama. We consider this orientation, as well as 3 other candidates formed by successive 90° rotations. LaLaLoc++'s retrieval stage is performed on each of these candidate rotations. The predicted position and orientation being given by the lowest computed embedding distance across the rotations. These are then used to initialise the sub-pixel refinement stage to further optimise position.

We list results from the Zillow Indoor Dataset in Tab. 9. Rotation prediction significantly increases the difficulty of the task. However, analysis of the rotational errors (Fig. 6) suggests that the vanishing point alignment is generally

Method	Localisation Success <5cm/2°	<10cm/5°	<20cm/10°
LaLaLoc	19.9%	40.9%	61.6%
LaLaLoc++	32.1%	51.7%	67.5%
LaLaLoc++ (Top 1)	16.4%	28.0%	36.0%
LaLaLoc++ (Top 2)	27.9%	45.6%	58.5%

Table 9. Localisation with and without an orientation prior. Methods which assume a known rotation are *italicised*.

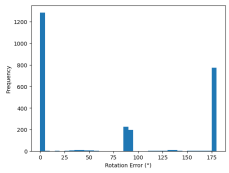

Fig. 6. Histogram of rotation error frequency on the Zillow Indoor Dataset.

accurate, but that there is ambiguity when multiple orientations can be considered, with multiple plausible poses. Therefore, we also consider a strategy where the *Top 1* orientation is computed as before, but the orientation with the second lowest predicted distance is also kept, labelled *Top 2* in Tab. 9. Localisation is considered successful if either hypothesis is within the tolerated error.

The results show that this generation of multiple pose hypotheses allows LaLaLoc++ to largely recover accuracy. And this setup actually outperforms LaLaLoc [15] for the strict thresholds, despite LaLaLoc using a known orientation. However, disambiguation of these hypotheses would require additional architectural information, such as windows and doors.

6 Discussion

Through experimental evaluation, LaLaLoc++ has been shown to outperform the previous floor plan localisation methods tested. However, floor plan localisation itself has some inherent limitations. For example, it is trivial to imagine buildings with many near structurally identical rooms. Even within the correct room, rotation can cause symmetries which leave two poses plausible, as explored in the previous section. In these scenarios, localisation through layout structure alone will fail due to the level of ambiguity. Despite this, and for the reasons stated in Sec. 1, we believe floor plan localisation is a promising and worthwhile avenue for research. In practical application, localisation methods are seldom used in isolation, and floor plan localisation is particularly useful as a strong prior from which more accurate methods can be initialised or verified, especially as it is designed for indoors where GPS is significantly less reliable.

7 Conclusion

In this paper, we have presented LaLaLoc++, a method for localisation in unseen environments with leveraging a floor plan as its only prior. We have demonstrated that a representative space of 3D layout structure can be inferred directly from 2D floor plans, and that doing so yields significant accuracy and performance improvements over previous methods for this task. We therefore show that localisation can be performed effectively in unseen environments by leveraging data that is near universal in the built world.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)
2. Balntas, V., Li, S., Prisacariu, V.: Relocnet: Continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 751–767 (2018)
3. Boniardi, F., Valada, A., Mohan, R., Caselitz, T., Burgard, W.: Robot localization in floor plans using a room layout edge extraction network. arXiv preprint arXiv:1903.01804 (2019)
4. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C.: Dsac-differentiable ransac for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6684–6692 (2017)
5. Brachmann, E., Rother, C.: Learning less is more-6d camera localization via 3d surface regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4654–4662 (2018)
6. Brachmann, E., Rother, C.: Expert sample consensus applied to camera relocalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7525–7534 (2019)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
8. Chen, S., Wang, Z., Prisacariu, V.: Direct-posenet: Absolute pose regression with photometric consistency. arXiv preprint arXiv:2104.04073 (2021)
9. 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)
10. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte carlo localization for mobile robots. In: Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C). vol. 2, pp. 1322–1328. IEEE (1999)
11. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: Camnet: Coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2871–2880 (2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Heinly, J., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3287–3295 (2015)
14. Hile, H., Borriello, G.: Positioning and orientation in indoor environments using camera phones. IEEE Computer Graphics and Applications **28**(4), 32–39 (2008)
15. Howard-Jenkins, H., Ruiz-Sarmiento, J.R., Prisacariu, V.A.: Lalaloc: Latent layout localisation in dynamic, unvisited environments. arXiv preprint arXiv:2104.09169 (2021)
16. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5974–5983 (2017)

17. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE international conference on computer vision. pp. 2938–2946 (2015)
18. Kim, S., Seo, M., Laptev, I., Cho, M., Kwak, S.: Deep metric learning beyond binary supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2288–2297 (2019)
19. Lim, H., Sinha, S.N., Cohen, M.F., Uyttendaele, M.: Real-time image-based 6-dof localization in large-scale environments. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 1043–1050. IEEE (2012)
20. Liu, L., Li, H., Dai, Y.: Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2372–2381 (2017)
21. Min, Z., Khosravan, N., Bessinger, Z., Narayana, M., Kang, S.B., Dunn, E., Boyadzhiev, I.: Laser: Latent space rendering for 2d visual localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11122–11131 (2022)
22. Pomerleau, F., Colas, F., Siegwart, R.: A review of point cloud registration algorithms for mobile robotics. Foundations and Trends in Robotics **4**(1), 1–104 (2015)
23. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
24. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12716–12725 (2019)
25. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–7. IEEE (2007)
26. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
27. 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)
28. Sun, C., Hsiao, C.W., Sun, M., Chen, H.T.: Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1047–1056 (2019)
29. Unicomb, J., Ranasinghe, R., Dantanarayana, L., Dissanayake, G.: A monocular indoor localiser based on an extended kalman filter and edge images from a convolutional neural network. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–9. IEEE (2018)
30. Valentin, J., Nießner, M., Shotton, J., Fitzgibbon, A., Izadi, S., Torr, P.H.: Exploiting uncertainty in regression forests for accurate camera relocalization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4400–4408 (2015)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)

32. Wang, S., Fidler, S., Urtasun, R.: Lost shopping! monocular localization in large indoor spaces. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2695–2703 (2015)
33. Winterhalter, W., Fleckenstein, F., Steder, B., Spinello, L., Burgard, W.: Accurate indoor localization for rgb-d smartphones and tablets given 2d floor plans. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3138–3143. IEEE (2015)
34. 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)
35. Zou, C., Colburn, A., Shan, Q., Hoiem, D.: Layoutnet: Reconstructing the 3d room layout from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2051–2059 (2018)