

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Acharya, D;Roy, SS;Khoshelham, K;Winter, S

Title:

A recurrent deep network for estimating the pose of real indoor images from synthetic image sequences

Date:

2020-10-01

Citation:

Acharya, D., Roy, S. S., Khoshelham, K. & Winter, S. (2020). A recurrent deep network for estimating the pose of real indoor images from synthetic image sequences. Sensors Switzerland, 20 (19), pp.1-20. <https://doi.org/10.3390/s20195492>.

Persistent Link:

<https://hdl.handle.net/11343/251374>

License:

CC BY

Article

A Recurrent Deep Network for Estimating the Pose of Real Indoor Images from Synthetic Image Sequences

Debaditya Acharya ^{1,2,*} , Sesa Singha Roy ³ , Kouros Khoshelham ¹ and Stephan Winter ¹ ¹ Department of Infrastructure Engineering, The University of Melbourne, Parkville, Victoria 3010, Australia; k.khoshelham@unimelb.edu.au (K.K.); winter@unimelb.edu.au (S.W.)² Department of Manufacturing, Materials and Mechatronics, RMIT University, Carlton, Victoria 3053, Australia³ Institute for Sustainable Industries and Livable Cities, Victoria University, Werribee, Victoria 3030, Australia; sesa.singharoy@live.vu.edu.au

* Correspondence: acharyad@student.unimelb.edu.au

† Current address: Room B310, Building 175, The University of Melbourne, Parkville, Victoria 3010, Australia.

Received: 25 August 2020; Accepted: 22 September 2020; Published: 25 September 2020



Abstract: Recently, deep convolutional neural networks (CNN) have become popular for indoor visual localisation, where the networks learn to regress the camera pose from images directly. However, these approaches perform a 3D image-based reconstruction of the indoor spaces beforehand to determine camera poses, which is a challenge for large indoor spaces. Synthetic images derived from 3D indoor models have been used to eliminate the requirement of 3D reconstruction. A limitation of the approach is the low accuracy that occurs as a result of estimating the pose of each image frame independently. In this article, a visual localisation approach is proposed that exploits the spatio-temporal information from synthetic image sequences to improve localisation accuracy. A deep Bayesian recurrent CNN is fine-tuned using synthetic image sequences obtained from a building information model (BIM) to regress the pose of real image sequences. The results of the experiments indicate that the proposed approach estimates a smoother trajectory with smaller inter-frame error as compared to existing methods. The achievable accuracy with the proposed approach is 1.6 m, which is an improvement of approximately thirty per cent compared to the existing approaches. A Keras implementation can be found in our Github repository.

Keywords: indoor localisation; camera pose regression; 3D building models; long short term memory

1. Introduction

Deep convolutional neural networks (CNNs) have been successfully used to perform localisation using single images without the need of an initial location [1–4]. These approaches consist of regressing the camera pose (location and rotation) of a single image with deep CNNs that have been fine-tuned using labelled real images. The labels (known poses) are generated from a 3D reconstruction of the indoor space using images, which is usually performed using the structure-from-motion (SfM) approach [5]. The SfM approach involves creating a database of overlapping images of the whole indoor space, which presents challenges for large indoor environments. This challenge limits the wide applicability of the CNN-based camera pose regression approaches [6].

Building information model (BIM)-PoseNet [7] and Bayesian BIM-PoseNet [8] eliminate the requirement of 3D image-based reconstruction using a texture-less 3D model of the indoor space. These approaches fine-tune a deep CNN using synthetically rendered images from a building information model (BIM), to estimate the pose of real images. However, the camera pose for each image frame is estimated without considering the spatial dependency between the frames. The per-frame error varies largely, thus resulting in high variance of distances between two consecutive frames.

Therefore, a relevant question is whether the performance of approaches that use synthetic images for visual localisation, such as [7,8], can be improved by incorporating spatio-temporal constraint between consecutive images. In addition, the uncertainty of camera pose plays a vital role in the estimation, as it provides the confidence of the estimations in the absence of ground-truth. Ideally, localisation errors should be correlated with the estimated uncertainties. The uncertainty of camera pose has been modelled using synthetic images [8]. However, that approach estimates the localisation uncertainty from a single image without considering the spatio-temporal continuity of image sequences. The pose estimates vary largely for consecutive images resulting in a jagged trajectory. Therefore, we explore another relevant question as to whether the uncertainty of camera pose can be modelled by using a sequence of synthetic images.

In this article, we propose Recurrent BIM-PoseNet (A Keras implementation can be found at <https://github.com/debaditya-unimelb/RecurrentBIM-PoseNet>), a deep Bayesian recurrent CNN that utilises synthetic image sequences obtained from a BIM, thereby eliminating the requirement of 3D reconstruction of the indoor spaces. At the test time, the camera poses of a sequence of real image is regressed. Our method takes advantage of temporal dependencies between consecutive images in a sequence to model the uncertainty, reduce the pose estimation error and generate a smoother trajectory. By smoothness, we mean having consistent inter-frame distances so that there are no sudden jumps between two consecutive camera poses.

We show that the pose estimations by Recurrent BIM-PoseNet result in a smoother trajectory with smaller inter-frame errors as compared to the state-of-the-art approaches. Additionally, the estimated uncertainties show correlation with the localisation errors, suggesting the suitability of the approach for modelling uncertainty. Moreover, we show that the proposed method does not need any colour or texture information to perform localisation. The main contributions are:

1. We improve the localisation accuracy of pose regression networks that use synthetic images to estimate the camera pose of real images. The spatio-temporal constraint of image sequences is utilised to improve accuracy and to generate a smoother trajectory.
2. The uncertainty of camera pose estimation is modelled by sampling from a sliding window of image sequences. We show that the modelled uncertainty shows correlation with the errors.

Section 2 reviews the visual approaches to indoor localisation. In Section 3, the theory and methodology are explained, which is followed by experiments and results in Section 4. Section 5 concludes the outcomes of the research and the possible future directions.

2. Background and Related Work

The primary limitation of many computer vision approaches, such as SLAM [9], visual odometry [10] and 3D model-based tracking [11] is the requirement of an initial location. This initial location is often derived from image-based retrieval approaches. The image-based retrieval approaches that provide the initial camera pose can be classified into three categories [12].

2.1. Image-Based Retrieval Approaches

2.1.1. Using Point Features

The first category includes approaches (such as [13]) that match the point features with an existing database of features, like 3D point clouds. These approaches estimate the camera pose directly using classical photogrammetry techniques like P3P [14], where the 3D information of each point feature is retrieved from the point clouds. The main limiting factor for these approaches is the dependency on the point clouds that are usually generated from SfM approaches, thus requiring a 3D reconstruction of the indoor space in advance to localisation.

2.1.2. Depth-Based Approaches

The second category includes depth-based approaches [15,16]. These approaches use depth cameras (or RGB-D cameras) to assign a location to each pixel of the image by comparing with a pre-existing depth map [15]. Cavallari et al. [17] adapt offline-trained regression forests to regress location in new indoor spaces, by dynamically updating the learnt model from a few training examples of the new space. Brachmann et al. [18] propose a differentiable RANSAC (DSAC) framework that uses RGB-D images to train a CNN that predicts the scene coordinates. Subsequently, the DSAC pipeline was optimised resulting in DSAC++ where state-of-the-art accuracies are reported [16]. These approaches depend on the 3D depth map of the indoor space from depth cameras, thus limiting their applicability only with depth cameras.

2.1.3. CNN-Based Approaches

The third category includes approaches that use deep CNNs for single image camera pose regression, like PoseNet [1]. These approaches fine-tune pre-trained networks using annotated real images obtained from SfM approaches to estimate the camera pose of an input image. The works of [3,19] improve the pose regression accuracy with a new geometric loss function and data augmentation method, respectively. Walch et al. [4] propose the use of a CNN-LSTM architecture to perform a structural dimensionality reduction of image features derived from the CNN to improve accuracy. Uncertainty has been modelled for CNN pose regression with Bayesian PoseNet [2] where dropout is used to draw Monte Carlo samples. The CNN-based pose regression approaches estimate the camera pose independently and do not exploit the valuable constraint of spatio-temporal smoothness. The pose estimates of such approaches lead to cases where the inter-frame distances are larger than the camera motion. Clark et al. [20] propose VidLoc, a recurrent network architecture to smooth the estimated trajectory and model the uncertainty of camera pose estimates.

The major limitation of using the deep CNN for camera pose regression is the requirement of large number of annotated real images. In practice, capturing thousands of overlapping images and estimating their corresponding poses by SfM approaches is challenging for large indoor environments. A potential solution to eliminate this requirement is to use photo-realistic synthetic images generated from an available 3D model.

2.2. Use of Synthetic Images

Jian et al. [19] use a 3D model reconstructed from real images to improve the camera pose regression, by generating synthetic images. A coarse visual localisation is performed using images and a BIM [21], where the authors compare the real and the synthetic images on the basis of features obtained from a CNN using cosine similarity. The authors classify the real image to its nearest synthetic image of known location and orientation.

BIM-PoseNet [7] utilises synthetic images obtained from a texture-less 3D model to train a network that estimates the camera poses of real images. The authors achieve an accuracy under 2 m by representing the real and the synthetic images as edge gradient magnitude (gradmag). Subsequently, the uncertainty of pose estimation of real images was modelled using synthetic images by Bayesian BIM-PoseNet [8]. However, the estimated camera poses of these approaches are less precise, compared to the approaches using real images. Further, the estimated poses show a high variation between consecutive images due to the missing spatio-temporal constraint between the consecutive images.

2.3. Limitations of Current Approaches

A limitation of [13,19,20] is the requirement of 3D reconstruction by SfM methods. The drawback of the works of [17,18] is the dependency on RGB-D cameras, thus deeming it unsuitable for most of the smartphone cameras. The shortcoming of the work by [21] is the coarse localisation, and the

inability to interpolate the absolute camera pose. Lastly, the works of [7,8] suffer from high variance of distances between consecutive frames.

In contrast, we do not perform 3D reconstruction by SfM methods, rather utilise synthetic images rendered from a texture-less 3D building model to adapt to new indoor scenes. Moreover, our approach can interpolate the absolute camera pose effectively in space between the synthetic training frames by regression, instead of predicting the nearest known location of the synthetic images. Lastly, compared to [7,8], we reduce the errors and generate a smoother trajectory.

In VidLoc [20], the authors drop the fully connected layers to improve the inference time of the network, to compensate for using multiple frames. Moreover, the authors performed all the experiments for real images only. In contrast, we retain the fully connected layers and demonstrate the advantage both for the real-real (fine-tuned with real and tested on synthetic) and synthetic-real (fine-tuned with synthetic and tested on real) cases. The presence of fully connected layer improves the camera pose estimation, especially for the synthetic-real case.

3. Methodology

The design of the proposed approach is shown in Figure 1. Recurrent BIM-PoseNet is fine-tuned using several types of synthetic image sequences generated from a 3D indoor model or a BIM. Subsequently, the networks are tested using real image sequences captured by a smartphone camera. The network regresses the camera poses corresponding to each test image in the sequence. The generation of the synthetic image sequences is described in Section 3.1. The architecture of Recurrent BIM-PoseNet is described in Section 3.2, fine-tuning and loss function are described in Section 3.3, and Section 3.4 explains the uncertainty modelling for camera pose estimations. The implementation details are presented in Section 4.1.

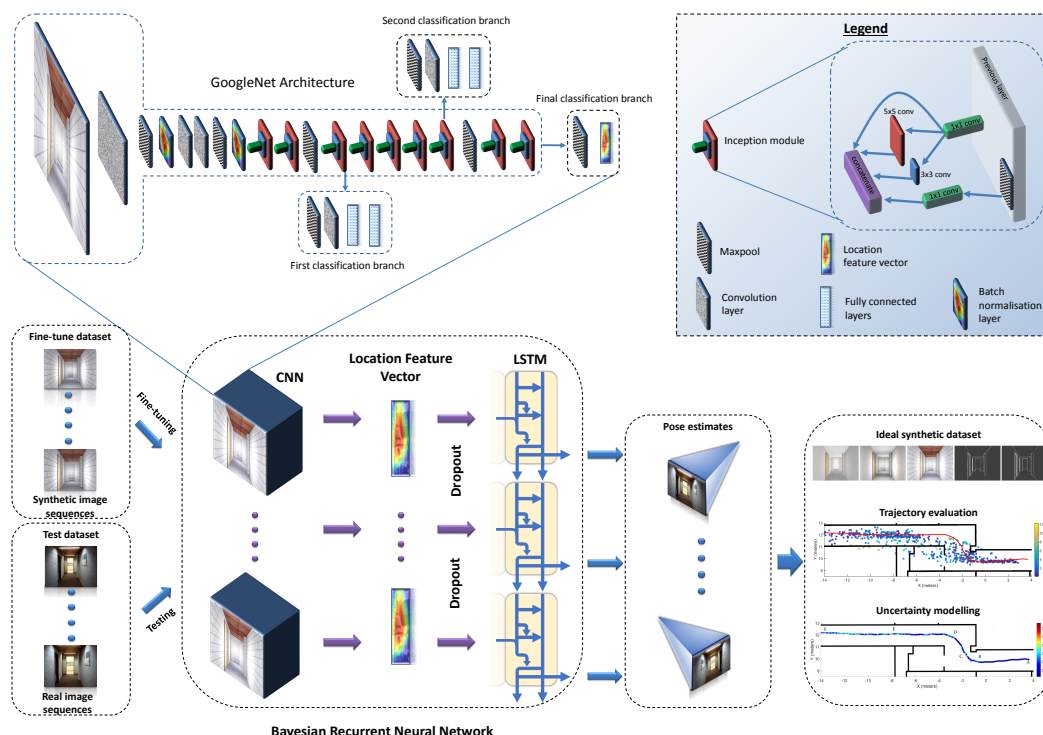


Figure 1. The design of the proposed approach. The network used is GoogleNet [22] containing nine inception modules. The output of the final classification branch yields the location feature vector which serves as input to the LSTM layer.

3.1. Generation of Synthetic Image Sequences

Synthetic images (The synthetic and the real datasets with their corresponding ground-truth poses are available at https://melbourne.figshare.com/articles/UnimelbCorridorSynthetic_zip/10930457) are obtained by rendering from a 3D model using different texture and lighting options. Previous work [7] has shown that a network fine-tuned using edge rendering (i.e., without texture and lighting) can successfully estimate the pose of real images represented as edge gradient magnitude (gradmag). Other types of rendering, such as the Cartoonish, photo-realistic and photo-realistic textured can also be used for the localisation task. Therefore, the first relevant question is which type of synthetic image is most suitable for the task of localisation with Recurrent BIM-PoseNet. Hence, the current research proposes to use several types of synthetic image sequences derived from the 3D indoor model to fine-tune Recurrent BIM-PoseNet and test using real image sequences.

Five sets of synthetic image sequences were generated, namely Cartoonish (Syn-car), photo-realistic (Syn-pho-real), photo-realistic textured (Syn-pho-real-tex), Gradient magnitude of Cartoonish (Gradmag-Syn-car) and Edge render (Syn-edge) with their respective ground-truth poses, as shown in Figure 2. The name in the bracket represents the pseudonyms of the respective datasets for convenience, and the naming convention is according to Blender (Blender is an open-source 3D computer graphics software that is used to perform simulations and animated films. Visit www.blender.org for more information.), which is used to render the synthetic images.

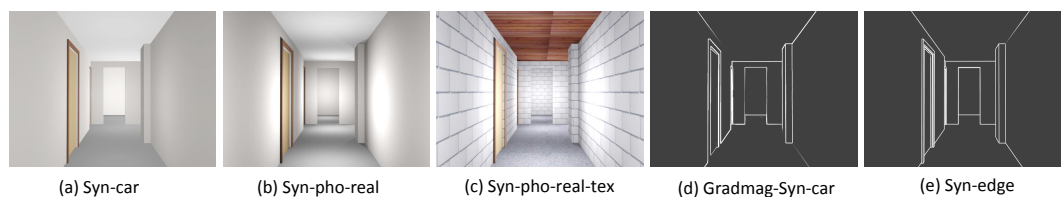


Figure 2. The five types of synthetic images generated from the 3D indoor model, after Acharya et al. [7].

Syn-car images (Figure 2a) were generated using a rendering model that roughly traces the path of the light. Syn-pho-real (Figure 2b) images were generated using an advanced light tracing model that follows the physical rules of light scattering and reflection. Syn-pho-real-tex (Figure 2c) images are also generated using the same light tracing model as Syn-pho-real. However, these images contain the synthetic texture of objects like brick and carpet texture on the walls and floor, respectively. Gradmag-Syn-car (Figure 2d) images are derived from Syn-car images by taking gradient magnitude of the images. Edge render images (Figure 2e) were generated by rendering only the edges of the 3D indoor model in the Field-of-View (FoV) of the virtual camera. More details of the synthetic dataset is present in Section 4.3.

3.2. Deep Learning Architecture

The proposed network consists of a deep Bayesian CNN and an LSTM layer to capture the spatio-temporal dependencies of consecutive frames. The LSTM [23] is a particular recurrent neural networks that is capable of learning long-term patterns in the input data. A standard LSTM consists of forget, input, output and reset gates in addition to a memory cell that enables the flow of data in and out of the memory cells that is regulated by the input and forget gates. We use GoogleNet [24] for image feature extraction by removing the softmax layers, and adding a 2048-dimensional dense layer (location feature vector). GoogleNet pre-trained on the Places image dataset [25] are used as the starting weights due to its suitability for the task of scene classification [1].

In the literature, sliding window has been used to generate samples from the data. These samples have been used by the LSTMs to exploit the spatio-temporal information of consecutive samples, and subsequently to model the uncertainty [20,26]. Therefore, to capture the spatio-temporal information, we generate sequences of images by sliding a window of length n , from the total number

of N images. The sequences are the input to the network. The CNNs generate a sequence of location feature vectors $(v_1, v_2, \dots, v_n)$ which is the input to the LSTM layer (refer to Figure 1). The output of the hidden states of the LSTM is utilised to estimate the pose of the sequence of images.

3.3. Fine-Tuning and Loss Function

The output of the LSTM units are connected to a 7-dimensional dense layer to regress n camera pose estimates (p_t^T) at each time step, which is defined as:

$$p_t^T = [x_t^T, q_t^T] \quad (1)$$

where x_t^T is the 3-dimensional location vector and q_t^T is the 4-dimensional rotational vector of the camera in quaternion format, for T^{th} sequence for time step t .

To add the spatio-temporal constraint to the camera pose estimations during the training procedure, a loss function L (Equation (2)) was used that represents the errors of the “whole” sequence of images by unrolling the network and performing back-propagation through time. This constraint enabled the network to learn to estimate consistent camera poses for the test image sequences.

$$L = \sum_{t=1}^{t=n} \|\hat{x}_t^T - x_t^T\|_2 + \beta \|\hat{q}_t^T - q_t^T\|_2 \quad (2)$$

where $\hat{x}_t^T$ and $\hat{q}_t^T$ are the estimated values of location and rotation vector of the t^{th} image of the sequence and β is the scaling factor to weigh the location and the rotational errors, respectively. The value of β is dependent on the indoor scene and is determined experimentally.

3.4. Modelling Uncertainty

During the test phase, the network regresses the unknown pose of the images from a window of real images. The uncertainty is modelled by gathering posterior distribution of the weights of the network [27]. The Monte Carlo samples are obtained by applying dropout during testing phase on the output of the network, and the mean of the samples is considered as the pose estimate. We use the trace of the covariance matrices of the camera pose samples, which provides a good numeric measure of the uncertainty. The multiple pose predictions by the network are utilised in the following manner to model the location and rotation uncertainties:

$$U_{x_t}^T = \text{sqrt}(\text{trace}(C(x_1^T, x_2^T, \dots, x_n^T))), U_{q_t}^T = \text{sqrt}(\text{trace}(C(q_1^T, q_2^T, \dots, q_n^T))) \quad (3)$$

where, $U_{x_t}^T$ and $U_{q_t}^T$ denote the uncertainty of camera location and rotation, respectively for the T^{th} frame, and C denotes the covariance matrix. Subsequently, the correlation factor R [8] is defined to quantify the correlation of the estimated localisation uncertainty and localisation errors in the following equations:

$$R_{LU/LE} = \frac{\text{cov}(U_x^N, E_x^N)}{\sigma_{U_x^N} \sigma_{E_x^N}}, R_{RU/RE} = \frac{\text{cov}(U_q^N, E_q^N)}{\sigma_{U_q^N} \sigma_{E_q^N}}, R_{LU/RU} = \frac{\text{cov}(U_x^N, U_q^N)}{\sigma_{U_x^N} \sigma_{U_q^N}} \quad (4)$$

where $R_{LU/LE}$ denotes the correlation of estimated location uncertainties vs. location errors, $R_{RU/RE}$ denotes the correlation of the estimated rotation uncertainties vs. rotation errors and $R_{LU/RU}$ denotes the correlation of the estimated location uncertainties vs. estimated rotation uncertainties. $U_x^N = [U_{x_t}^1, U_{x_t}^2, \dots, U_{x_t}^N]$ is the vector containing the estimated location uncertainties, $U_q^N = [U_{q_t}^1, U_{q_t}^2, \dots, U_{q_t}^N]$ is the vector containing the estimated rotation uncertainties, $E_x^N = [E_{x_t}^1, E_{x_t}^2, \dots, E_{x_t}^N]$ is the vector containing the location errors and $E_q^N = [E_{q_t}^1, E_{q_t}^2, \dots, E_{q_t}^N]$ is the vector containing the rotation errors. $\sigma_{U_x^N}$, $\sigma_{U_q^N}$, $\sigma_{E_x^N}$ and $\sigma_{E_q^N}$ denote the standard deviations of U_x^N , U_q^N , E_x^N and E_q^N , respectively, and cov represents the covariance between two random variables.

4. Experiments and Results

4.1. Implementation Details

The networks were implemented in Keras [28] using TensorFlow libraries [29] on Linux. Adam gradient descent optimisation algorithm with a fixed learning rate of 10^{-3} was used to fine-tune the networks for 400 epochs. A Tesla P100 graphics processor unit (GPU) was used to accelerate the code with NVIDIA CUDA[®] Deep Neural Network library (cuDNN). The fine-tuning and testing images were resized to a resolution of 320×240 pixels, and a central crop of dimension 224×224 was applied. The mean of the fine-tuning dataset was subtracted from the input images during fine-tuning and test time.

4.2. Experimental Design

To evaluate the performance of Recurrent BIM-PoseNet and determine which rendering of synthetic images is the most suitable for camera pose estimation and modelling uncertainty, we design the following experiments.

- Experiment 1: Using real images for training and testing. A baseline accuracy was established by fine-tuning Recurrent BIM-PoseNet using real images, to compare the results obtained from the proposed approach being fine-tuned using synthetic data. Parameters such as ideal LSTM length, ideal window length and the correlation of localisation errors with the estimated localisation uncertainties are identified. The results are presented in Section 4.4.
- Experiment 2: Using synthetic images for training and real images for testing. In this experiment, Recurrent BIM-PoseNet was fine-tuned using several types of synthetic image sequences. Subsequently, pose regression ability of these fine-tuned networks were evaluated by using real image sequences during test, and compared with the previous approaches. The results are presented in Section 4.5.
- Experiment 3: Modelling uncertainty. This experiment consisted of modelling the uncertainty of the estimated camera poses, and evaluating the correlation of the localisation errors with the estimated localisation uncertainties. The results are compared with the results of Bayesian BIM-PoseNet, and are presented in Section 4.6.

4.3. Dataset

Synthetic image dataset: A 3D indoor model was obtained from a BIM that was created as a part of the ISPRS benchmark on Indoor modelling [30], and was used in our previous works [31–33]. The BIM consists of the third floor of Block B of the Department of Infrastructure Engineering at the University of Melbourne, Australia and covered a part of the corridor with an area of 230 m². The level-of-detail (LoD) of this BIM can be considered as LoD 300 as per the BIM specifications [34,35].

The synthetic images were generated by moving a virtual camera in the BIM at a spacing of 5 centimetres along a trajectory that is approximately 30 m long. Additional images were generated by rotating the camera by 10° around the Y and Z axes, to address the problem of the lower number of images for fine-tuning the networks as pointed out by Jian et al. [19]. For each type of rendering, 2500 synthetic images were generated, having a resolution of 640×480 pixels.

Real image dataset: A total number of 950 images having a resolution of 640×480 were captured by a smartphone camera having a FoV approximately equal to 56°. The focus of the camera was fixed to avoid any out-of-focus images, and the exposure of the images was fixed to limit underexposed images. The camera was calibrated to estimate the intrinsic parameters that were used to design the virtual camera. However, the distortion of the images was not modelled for the virtual images. The images were acquired at a constant rate of 30 frames per second. The edge images of the real images were generated by taking the gradmag of the images and suppressing the weak edges

below a threshold. A sample real image and its corresponding gradmag image are shown in Figure 3a,b, respectively.

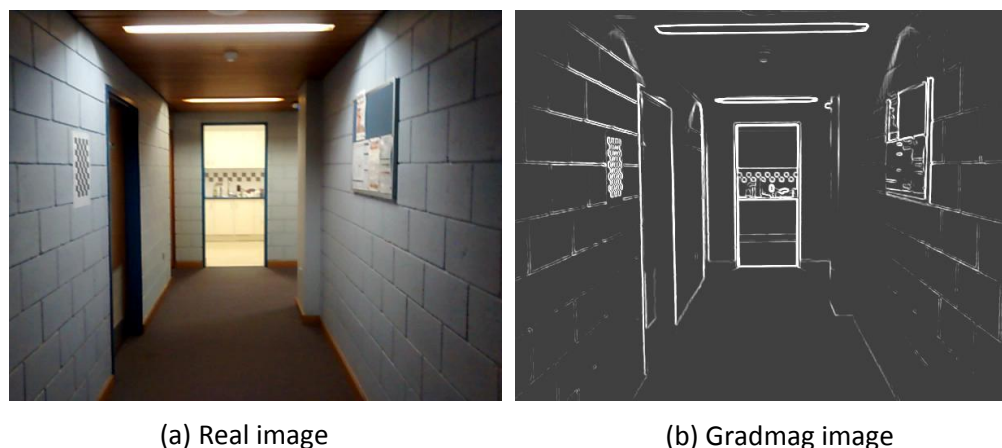


Figure 3. (a) A real sample image and (b) the corresponding gradient magnitude (gradmag) image after thresholding weak edges.

4.4. Experiment 1: Baseline Performance Using Real Image Sequences

A baseline accuracy was established using real image sequences with known ground-truth camera poses (see Supplementary Material). We set the value β in the loss function (Equation (2)) as 600 based on our previous experiments [7]. The whole dataset was used to create real image sequences containing consecutive images, which was partitioned randomly into fine-tuning, validation and testing sets (split 60:20:20).

Achievable accuracy: The length of the LSTM units is important to the performance of the network. A longer length of LSTM units will lead to over-fitting the network, where the network will perform excellently for the validation data but will perform poorly for other unseen test data. On the other hand, short LSTM units will lead to under-fitting where the network will not be able to perform well on validation and test data as well. Additionally, the input to LSTM units are the location feature vector, having a length of 2048. Therefore the short length of LSTM units might not be able to correlate the image features with the camera pose.

To identify the ideal number of LSTM units required for the proposed network, we performed an experiment by fine-tuning networks with different LSTM lengths (64, 128, 256, 512, 1024 and 2048). The window size for this experiment was maintained at a constant of three images. It was experimentally identified that there is no improvement in the accuracy of the estimations using more than 512 LSTM units. Therefore, for the rest of the experiments using real data, we use 512 LSTM units.

To identify the ideal window length, the networks were fine-tuned and tested using different window lengths of images. We fine-tuned four networks using a window of 3, 5, 10 and 15 images and tested them with a window of 1, 3, 5, 10 and 15 images. It was identified that the network fine-tuned using a window of ten image sequences and tested with a window of ten images performs best amongst the other networks, and the median of test error is 0.17 m and 0.83°. Figure 4f shows the distribution of the predicted points by the network.

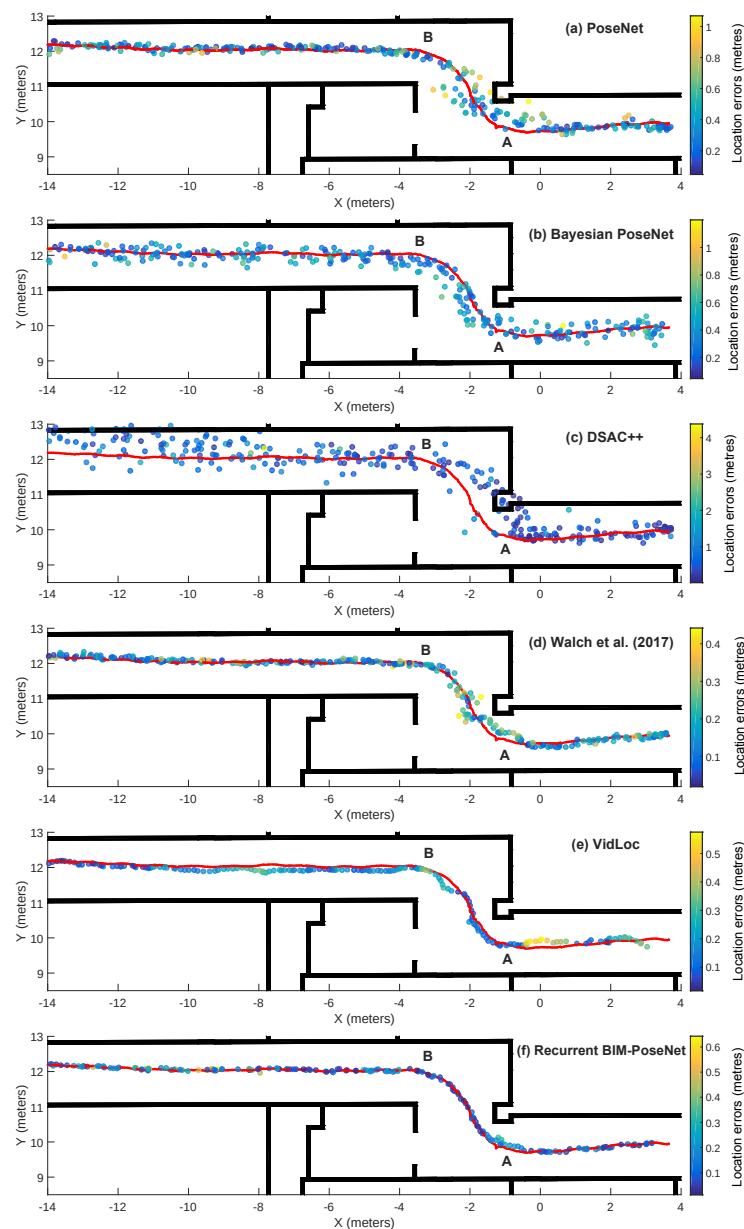


Figure 4. The predictions of (a) PoseNet [1], (b) Bayesian PoseNet [2], (c) differentiable RANSAC (DSAC)++ [16], (d) Walch et al. [4], (e) VidLoc [20], (f) Recurrent building information model (BIM)-PoseNet fine-tuned and tested using a sequence of ten real images. The colour represents the magnitude of error of each point, and the red line denotes the ground-truth trajectory.

Comparison with previous approaches: We compare the proposed approach with PoseNet [1], Bayesian PoseNet [2], DSAC++ [16], our implementation of VidLoc [20] and Walch et al. [4]. Figure 4 shows the estimated trajectories for the different approaches. It is observed that PoseNet, Bayesian PoseNet and DSAC++ lack the temporal smoothness for the whole trajectory, and contains large errors near the turn AB of the corridor which is prone to motion blur. Although the approaches proposed by [4] and VidLoc produce considerably smoother trajectories, they fail to generate consistent results near the turn AB of the trajectory. It is observed that the trajectory predicted by Recurrent BIM-PoseNet is consistent and smoother for the whole trajectory length as compared to the other approaches.

Table 1 summarises the errors and the inter-frame distances for the approaches. It is observed that the errors and the inter-frame distances of PoseNet and Bayesian PoseNet are similar. DSAC++ is known to perform poor for datasets containing a lesser number of training images, and can

plausibly explain the poor performance for the current dataset with approximately 600 training images. The approach proposed by [4] performs better than the aforementioned approaches. It is observed that by using a window of ten images, the location and rotation errors for Recurrent BIM-PoseNet approach are reduced by a factor of 3 as compared with PoseNet and Bayesian PoseNet. Although VidLoc and Recurrent BIM-PoseNet show comparable results in terms of inter-frame distances, Recurrent BIM-PoseNet performs better in terms of absolute errors, and demonstrates the advantage of retaining the fully connected layers.

Table 1. The comparison of the median errors and inter-frame distances of different approaches.

Approach	Errors (Metres, Degrees)	Inter-Frame Distances (Metres)
PoseNet [1]	0.33 m, 1.85°	0.30 m
Bayesian PoseNet [2]	0.29 m, 1.53°	0.38 m
DSAC++ [16]	0.53 m, 0.61°	0.64 m
Walch et al. [4]	0.15 m, 1.62°	0.14 m
VidLoc [20]	0.16 m, 0.87°	0.10 m
Recurrent BIM-PoseNet (ours)	0.10 m, 0.83°	0.10 m

Figure 5 shows the cumulative distribution function (CDF) of the location errors and the inter-frame distances for the approaches. The location errors are calculated as the Euclidean distances from the ground-truth, and the inter-frame distances are the Euclidean distances between two consecutive frames. Figure 5 shows that Recurrent BIM-PoseNet is the most precise in terms of localisation errors, and its performance is competitive compared to VidLoc in terms of inter-frame distances. The precision of the approach proposed by [4] is also challenging to Recurrent BIM-PoseNet and VidLoc. The performance of PoseNet and Bayesian PoseNet are very close, DSAC++ being the worse performing.

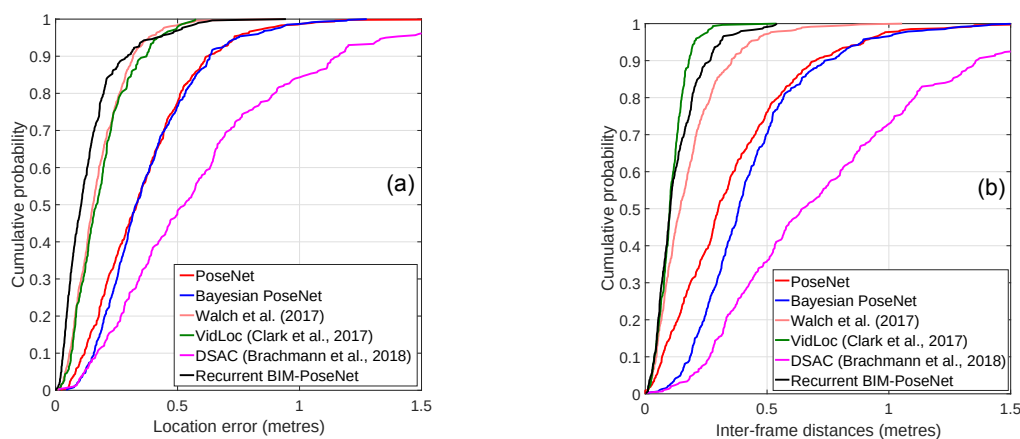


Figure 5. Cumulative distribution function (CDF) plots of the (a) errors and (b) inter-frame distances of the estimated camera poses for different approaches for Experiment 1.

Estimated uncertainty: Figure 6 shows the correlation between the estimated uncertainties and the actual errors for the predictions of a network that was fine-tuned and tested with a window of ten images. Compared to the uncertainty modelled by Bayesian BIM-PoseNet on the same dataset [8], Recurrent BIM-PoseNet shows a better correlation between the estimated location uncertainty vs. location errors ($R = 0.31$), as well as for the estimated rotation uncertainty vs. rotation errors ($R = 0.67$). However, the location uncertainty vs. rotation uncertainty estimated by Recurrent BIM-PoseNet shows low correlation ($R = 0.22$) with the rotation errors as compared to Bayesian BIM-PoseNet.

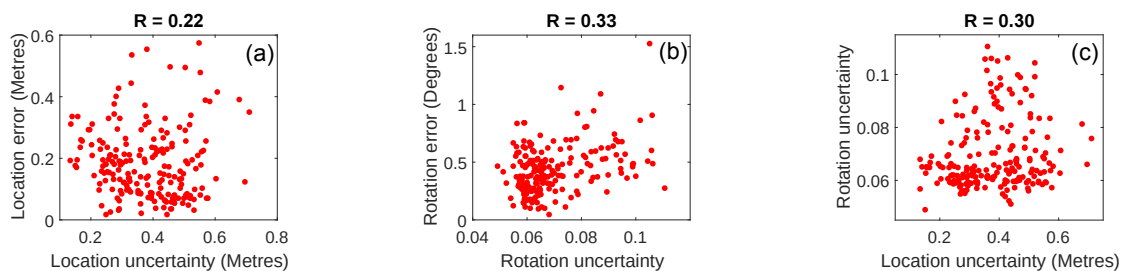


Figure 6. The modelled uncertainty by Recurrent BIM-PoseNet. (a) Estimated location uncertainty vs. location errors. (b) Estimated rotation uncertainty vs. rotation errors. (c) Estimated location uncertainty vs. estimated rotation uncertainty. R denotes the correlation factor.

Figure 7 shows the training and validation losses for network fine-tuned and validated with real image sequences, with a window of 10 images. The results indicate that the validation loss did not improve after 250 epochs, and the best network was selected as the one having the least validation loss.

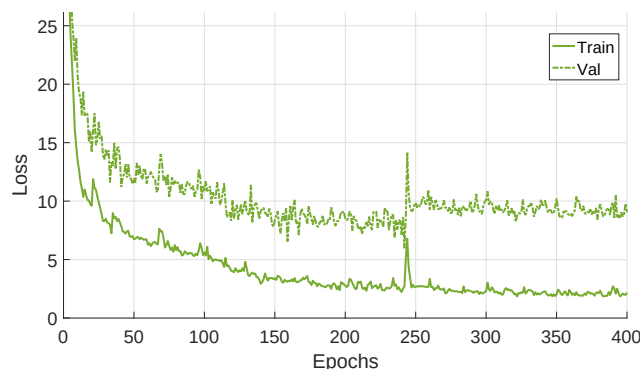


Figure 7. The training and validation losses for the network fine-tuned and validated with real image sequences, having a window size of 10.

4.5. Experiment 2: Performance of the Network Fine-Tuned Using Synthetic Images and Tested with Real Images

Recurrent BIM-PoseNet was fine-tuned with the five synthetic image datasets as shown in Figure 2, using $\beta = 600$, and subsequently, these fine-tuned networks were tested with real images as shown in Figure 3. The networks fine-tuned with images Syn-Car, Syn-photo-real and Syn-Pho-real-tex were tested with the real image (Figure 3a) and the networks fine-tuned with Gradmag-Syn-Car and Syn-edge images were tested with gradmag of the real image (Figure 3b). We used naive (0–0.5) and variational dropouts (0–0.5) for the LSTM units [36] which improved the accuracy of camera pose regression being tested with real images.

Similar to the experiments with real data, to identify the ideal length of the LSTM units for synthetic image sequences, we fine-tune networks using different LSTM lengths (64, 128, 256, 512, 1024 and 2048 units). Gradmag-Syn-car dataset, which has shown the best performance among different renderings [8] was used to fine-tune the networks. For fine-tuning and testing, a window of length three was used. It was identified that there is no improvement in the location and rotation accuracy beyond using 256 LSTM units. Compared to using real data (512 units), this is lower and can be explained by the less information content of synthetic images as compared to real images.

Figure 8 shows the location and rotation errors for the network that was fine-tuned using different window lengths of Gradmag-Syn-Car dataset and tested with varying window lengths of gradmag of real images. Similar to the benchmark experiment using real images, Figure 8 indicates that the localisation accuracy of the network fine-tuned using a window of ten synthetic images and tested with a sequence of ten real images is the best amongst the other combinations, where the achievable accuracy is 1.62 m and 9.29°.

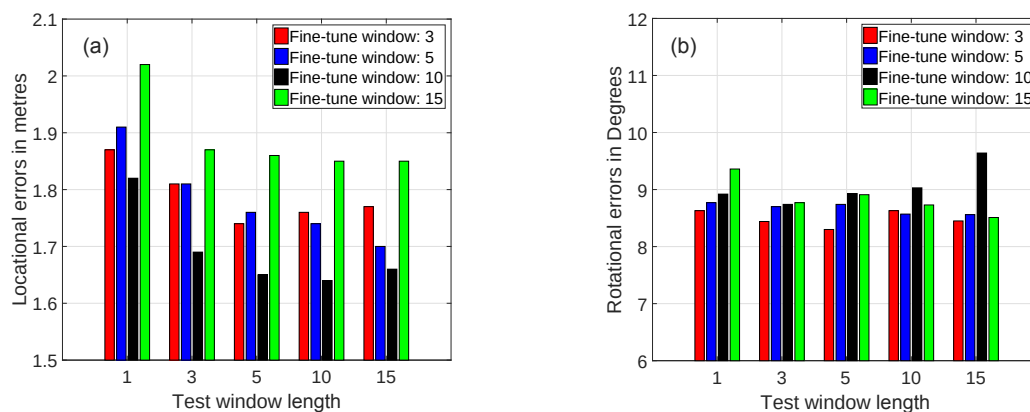


Figure 8. The effects of window length on the (a) location errors and (b) rotation errors for networks fine-tuned using Gradmag-Syn-car dataset and tested on gradmag of real images.

The fine-tuned networks can generalise better on a longer window of test image sequences than when it was fine-tuned. For instance, consider the network that was fine-tuned on a window of three synthetic images, which makes good predictions of the location and rotation of the real image sequences having a window length of 5, 10 and 15 images. The performance of the networks to predict the location of image sequences having a shorter window length is worse, compared to testing with the same window length. For example, consider the network fine-tuned using a window of ten images, where its performance for predicting the location of the sequence of images having a length of 1 and three images is worse, compared to testing it using a window of ten image sequences. In contrast, this trend is not valid for rotational errors, where the networks can make sound predictions of rotation for a longer and shorter window of image sequences than when it was fine-tuned. Figure 8b shows that there is a slight decrease in the rotation errors with the increase in the length of the window of the image sequence.

Accuracy evaluation and comparison with previous works: To evaluate and compare the performance using different types of synthetic images, five networks were fine-tuned using the five different synthetic datasets and tested using real images. As the datasets, Syn-car, Syn-pho-real and Syn-pho-real-tex are similar in appearance to the real image, the networks fine-tuned using these datasets were tested directly using real images. However, the networks that were fine-tuned using the gradmag datasets, namely Gradmag-Syn-car and Syn-edge were tested using gradmag of real images. For fine-tuning all the networks, a window of ten synthetic images was used, and for testing a window of ten real images was used.

Figure 9 shows the estimated trajectories by Recurrent BIM-PoseNet fine-tuned with different synthetic image datasets. It is observed that the distribution of the estimated points for the network fine-tuned with Gradmag-Syn-car images (Figure 9d) is more consistent with the ground-truth, and a similar trend is noticed for the Syn-edge images (Figure 9e). Additionally, the bias or shift in the location of the estimated points reported in [7,8] is greatly reduced, especially for the Gradmag-Syn-car dataset. The predictions for the network fine-tuned with Syn-Car, Syn-pho-real and Syn-pho-real-tex are skewed and deviate from the ground-truth.

Figure 10 shows the CDF plots of the localisation errors and the inter-frame distances of the different approaches. The errors and the inter-frame distances are calculated as explained in Experiment 1. It is evident that Recurrent BIM-PoseNet outperforms all the other approaches in terms of precision. The inter-frame distances of all the approaches are comparable; however, only a slight improvement is identified for Recurrent BIM-PoseNet.

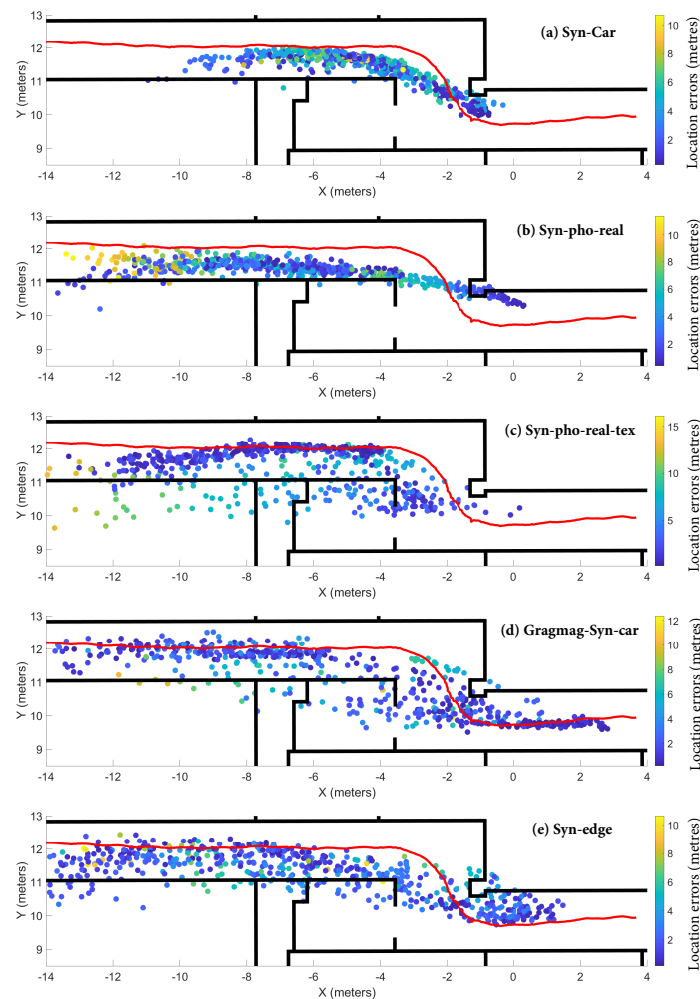


Figure 9. Estimated camera poses by the Recurrent BIM-PoseNet fine-tuned using (a) Syn-car, (b) Syn-pho-real, (c) Syn-pho-real-tex, (d) Gragmag-Syn-car, (e) Syn-edge. Networks fine-tuned with Syn-Car, Syn-pho-real and Syn-pho-real-tex images were tested with real images, whereas networks fine-tuned with Gragmag-Syn-car and Syn-edge were tested with gradmag of real images.

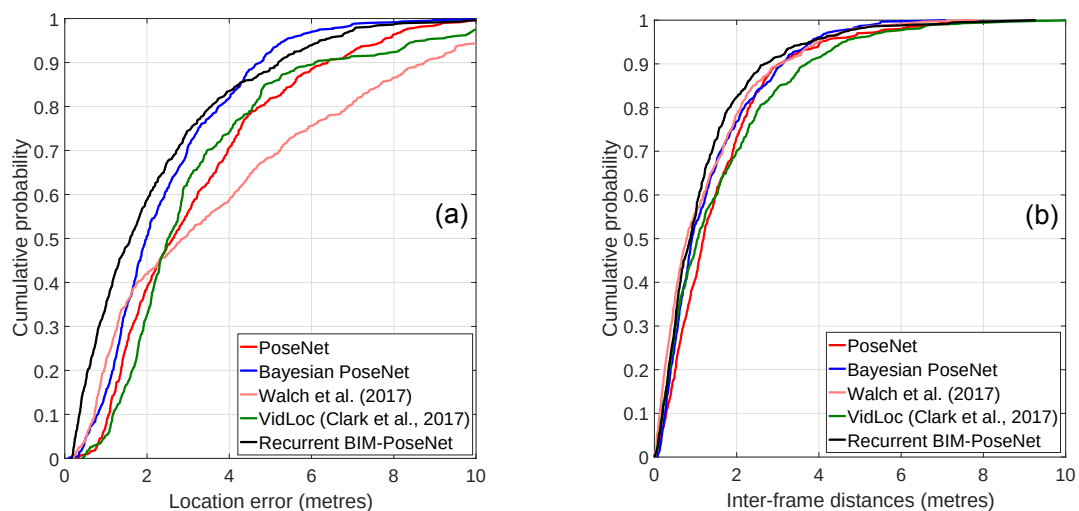


Figure 10. CFD plots of the (a) localisation errors and (b) inter-frame distances of the estimated camera poses for different approaches for Gragmag-Syn-car dataset.

Table 2 shows the test errors of Recurrent BIM-PoseNet compared to the other approaches. From Table 2, we observe that Recurrent BIM-PoseNet fine-tuned with Gradmag-Syn-car images achieves the best localisation accuracy (1.62 m, 9.29°) amongst all approaches. Compared to [7,8], we see improvements in the accuracies for all the image renderings, with the exception of Syn-car images. Compared to [4] we notice improvements for all renderings except for Syn-pho-real images. In addition, Recurrent BIM-PoseNet achieves better results in contrast to VidLoc for the Syn-pho-real-tex, Gradmag-Syn-car and Syn-edge images. This improved accuracy of Recurrent BIM-PoseNet demonstrates the advantage of retaining the fully connected layers, as compared to VidLoc. This could be plausibly because the fully connected layers aid in the identification of the high-level structural information in the gradmag images, which the shallow layers of the network are incapable of. The improvement on rotation accuracy of Recurrent BIM-PoseNet is however apparent only for some of the renderings. DSAC++ [16] did not converge for any of the synthetic image renderings, in spite of having 2500 training images. Interestingly, for all approaches the performance of Gradmag-Syn-car and Syn-edge is quite similar and noticeably better than the other types of synthetic images.

Table 2. The comparison of the errors of different approaches for the five synthetic image datasets being tested using real images. DSAC++ [16] did not converge for all the synthetic datasets in spite of containing 2500 training images.

Approach	Syn-Car	Syn-Pho-Real	Syn-Pho-Real-Tex	Gradmag-Syn-Car	Syn-Edge
BIM-PoseNet	6.25 m, 37.16°	5.99 m, 11.33°	3.06 m, 12.25°	2.63 m, 6.99°	1.88 m, 7.73°
Bayesian BIM-PoseNet	3.87 m, 8.38°	4.08 m, 25.03°	3.73 m, 13.53°	1.98 m, 7.33°	2.41 m, 12.53°
Walch et al. [4]	4.09 m, 22.28°	2.88 m, 15.31°	2.50 m, 11.99°	2.89 m, 19.22°	1.90 m, 12.42°
VidLoc [20]	3.04 m, 11.81°	2.78 m, 11.45°	2.73 m, 11.12°	2.60 m, 11.42°	2.30 m, 7.26°
Recurrent BIM-PoseNet	3.97 m, 15.20°	3.01 m, 8.50°	2.23 m, 8.31°	1.62 m, 9.29°	1.87 m, 11.15°

Figure 11a shows the distribution of the errors and some of the frames that resulted in large errors along the trajectory (fine-tuned on Gradmag-Syn-car dataset), to identify error-prone areas. Table 3 shows the errors and estimated uncertainties for these frames. The large errors are likely the result of poor geometry of the scene, where the doors are far away from the camera, for example, near Point A. Other error sources include the motion blur that results in larger errors for the Points B, C and D of the trajectory. Additionally, the objects that are present in the images, but not in the 3D indoor model, such as the notice boards, poster and light flares (seen in Point E and F), are the other sources of errors. The errors near Point D in Figure 11a are reduced for Recurrent BIM-PoseNet (up to 6 m) as compared to [7] (up to 8 m) and [8] (up to 10 m).

Table 3. The comparison of the errors and estimated uncertainties for the frames of Figure 11.

Frame	Location Error (Metre)	Location Uncertainties (Metre)	Rotation Error	Rotational Uncertainty
11	6.82	0.36	4.52°	0.09
149	7.60	0.41	2.85°	0.08
188	9.11	0.73	30.61°	0.16
250	5.92	0.68	9.67°	0.14
400	6.45	0.53	4.51°	0.09
568	6.91	0.53	15.16°	0.12

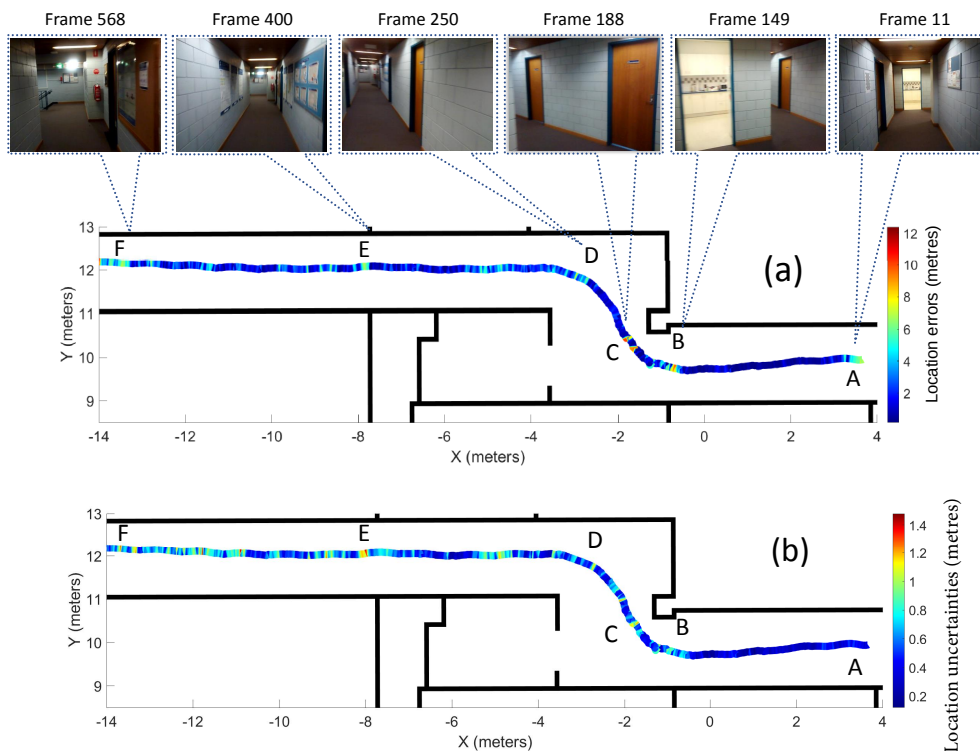


Figure 11. The distribution of the (a) location errors and (b) estimated location uncertainties along the trajectory for the network fine-tuned using Gradmag-Syn-car dataset. The first row shows some of the frames with large location errors near Points A–E. Table 3 shows the errors and estimated uncertainties of the frames.

4.6. Experiment 3: Modelling Uncertainty Using Synthetic Images

Figure 12 shows the correlation between the estimated uncertainties and the actual errors. Figure 12a–i show the (1) localisation errors vs. estimated location uncertainties, (2) rotation error vs. estimated rotation uncertainties and (3) estimated location uncertainties vs. estimated rotation uncertainties for the networks fine-tuned using the Syn-car, Syn-pho-real and Syn-pho-real-tex datasets. The network fine-tuned using Syn-pho-real-tex dataset shows the highest correlation for location errors vs. estimated location uncertainties, estimated location uncertainties vs. estimated rotation uncertainties and similar rotational errors vs. estimated rotation uncertainties, compared to the network fine-tuned using Syn-pho-real dataset. With the exception of the network fine-tuned using Syn-car dataset, there is an improvement in the correlation of the location and rotational uncertainties for Recurrent BIM-PoseNet compared to Bayesian BIM-PoseNet for Syn-pho-real ($R = 0.50$ vs. $R = 0.04$) and Syn-pho-real-tex ($R = 0.66$ vs. $R = 0.33$) datasets. However, the network fine-tuned using Syn-car dataset shows a negative correlation with location errors vs. estimated location uncertainties as well as with rotation errors vs. estimated rotation uncertainties. In addition, the correlation of the estimated location uncertainties vs. estimated rotation uncertainties are least as compared to the other datasets. Table 4 compares the correlation factor of Bayesian BIM-PoseNet with Recurrent BIM-PoseNet.

Figure 12j–o show the errors and the uncertainties for the networks fine-tuned using Gradmag-Syn-car and Syn-edge datasets. It is identified that the network fine-tuned using Gradmag-Syn-car dataset performs better as compared to the network fine-tuned using Syn-edge dataset for rotation error vs. estimated rotation uncertainty, and estimated rotation uncertainty vs. estimated location uncertainty. However, there is a slight decrease in the correlation for the location error vs. estimated location uncertainty. Compared to Bayesian BIM-PoseNet, the correlation of estimated location uncertainty vs. estimated rotational uncertainties improves for both the datasets and is summarised in Table 4. Moreover, the location error vs. estimated location uncertainty, and rotation

error vs. estimated rotation uncertainty for the network fine-tuned using Gradmag-Syn-car dataset are better as compared to Bayesian BIM-PoseNet (Table 4).

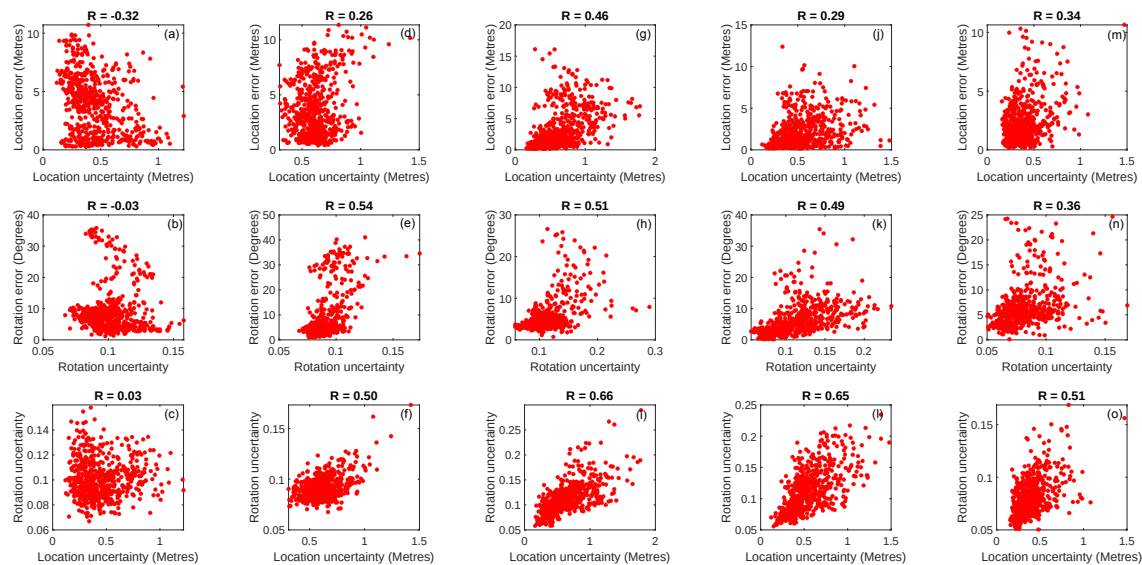


Figure 12. The uncertainty modelled by the network (a–c) fine-tuned using Syn-car and tested on real images, (d–f) fine-tuned using Syn-pho-real and tested on real images, (g–i) fine-tuned using Syn-pho-real-tex and tested on real images, (j–l) fine-tuned using Gradmag-Syn-car and tested on gradmag of real images, (m–o) fine-tuned using Syn-edge and tested on gradmag of real images. The first column shows the estimated location uncertainty vs. location error. The second column shows the estimated rotation uncertainty vs. rotation error, and the third column shows the estimated location uncertainty vs. estimated rotation uncertainty.

Figure 11b shows the trend of the uncertainties modelled by the network fine-tuned on Gradmag-Syn-car dataset for the visualisation of the uncertain areas of the trajectory. It is observed that there is a good correlation between the errors (Figure 11a) and the uncertainties throughout the trajectory, except for Point A, where the errors are large, but the modelled uncertainties are low. Recurrent BIM-PoseNet is more confident in handling perceptual aliasing; for instance, near Point D in Figure 11. However, the uncertainty of camera pose estimation increases in the presence of artefacts such as notice boards, posters and light flares, and that explains the high uncertainties for Points E and F.

Table 4. The comparison of error and uncertainties for Recurrent BIM-PoseNet with Bayesian BIM-PoseNet.

Fine-Tuned on	Bayesian BIM-PoseNet			Recurrent BIM-PoseNet		
	$R_{LU/LE}$	$R_{RU/RE}$	$R_{LU/RU}$	$R_{LU/LE}$	$R_{RU/RE}$	$R_{LU/RU}$
Syn-car	0.12	0.31	0.34	−0.32	−0.03	0.03
Syn-pho-real	0.36	−0.01	0.04	0.26	0.54	0.50
Syn-pho-real-tex	0.33	0.53	0.33	0.46	0.51	0.66
Gradmag-Syn-car	0.42	0.50	0.59	0.29	0.49	0.65
Syn-edge	0.46	0.40	0.41	0.34	0.36	0.51

Figure 13 shows the training and validation losses for the networks fine-tuned with a window of 10 synthetic images and validated with a window of 10 real images. It is observed that the validation losses of the networks are significantly higher as compared to the networks fine-tuned on real images (Figure 7). The higher validation losses might be a result of the differences between the synthetic and the real images (artefacts), and points towards a future research direction (explained in the conclusions).

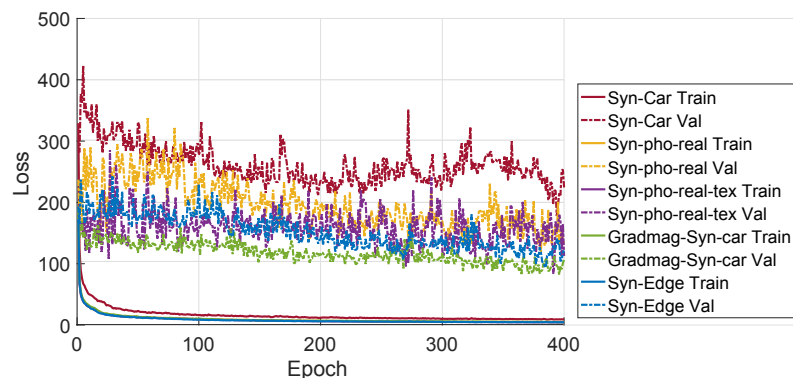


Figure 13. The training and validation losses for the networks fine-tuned with synthetic image sequences and validated with real images sequences, having a window size of 10.

4.7. Computation Times

Table 5 shows the fine-tuning times and test times of the networks for different window lengths. A Tesla P100 GPU (@1.32 GHz) having a memory of 12 GB and a Xeon (@2.20 GHz) CPU was used for fine-tuning the networks. A memory of 64 GB was allocated for the fine-tuning process and for testing a memory of 16 GB was allocated for all the networks. A batch size of 25 was used to fine-tune the networks, except for the network trained on a window of 15 images, where a batch size of 15 was used due to memory constraints. For reference, the test times of BIM-PoseNet and Bayesian BIM-PoseNet are 5 milliseconds and 67 milliseconds on a similar GPU, respectively.

Table 5. The fine-tuning time and test times for the networks using different window lengths. The network with window length 1 is the implementation of BIM-PoseNet for comparing times.

Window Length	Fine-Tune Time (Hrs)	Test Time GPU (ms)	Test Time CPU (s)
1	1:44	12	0.13
3	4:34	16	0.40
5	7:07	35	0.67
10	12:43	52	0.94
15	23:48	108	2.12

As most of the smartphones are equipped with a GPU [37], Recurrent BIM-PoseNet can run in real-time on such devices, deeming it suitable for practical applications. However, there will be a lag in the camera pose estimation depending on the window length. For instance, while using a window length of ten images, there will be a lag of ten frames, which is feasible for real-time operations.

5. Conclusions

We propose a deep Bayesian recurrent CNN that, when fine-tuned using sequences of synthetic indoor images, can estimate the camera pose for a sequence of real images. The proposed approach eliminates any requirement of 3D reconstruction of the indoor space by SfM approaches. The results of the experiments suggest that an accuracy of 1.6 m can be achieved by fine-tuning the proposed network using synthetic edge image sequences, that do not contain any colour or texture information, but only structural edges.

It is identified that a window of ten synthetic images is ideal for fine-tuning the proposed network for camera pose regression. The estimated camera poses results in a slightly smoother trajectory as compared to the existing approaches by exploiting the spatio-temporal information of the image sequences. Moreover, we show that the modelled uncertainty of the estimated camera poses is correlated with the errors.

The errors due to perceptual aliasing are reduced using image sequences as compared to the previous approaches, and as a result, the proposed network estimates the camera poses more accurately.

The large errors are caused either due to the presence of motion blur or artefacts in the image, such as light flare or notice boards. Poor geometry of the scene is another error source where the image landmarks such as doors cover a small portion of the image being far away from the camera.

There is still some room for improvement in the current study that points towards interesting future directions. The semantic information of the BIM can be used to generate segmented images that can be used to fine-tune the proposed network, to reduce the effect of the artefacts. At the test time, semantically segmented real images [38] can be used.

Supplementary Materials: The following are available online at https://melbourne.figshare.com/articles/UnimelbCorridorSynthetic_zip/10930457.

Author Contributions: Conceptualization, D.A., K.K. and S.W.; methodology, D.A. and K.K.; software, D.A. and S.S.R.; validation, D.A., and K.K.; formal analysis, D.A., K.K. and S.W.; investigation, D.A. and K.K.; resources, K.K. and S.W.; data curation, D.A.; writing—original draft preparation, D.A.; writing—review and editing, D.A., K.K. and S.W.; visualization, D.A. and S.S.R.; supervision, K.K. and S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This research is supported by a Research Engagement Grant from the Melbourne School of Engineering and a Melbourne Research Scholarship. This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. 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 (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
2. Kendall, A.; Cipolla, R. Modelling uncertainty in deep learning for camera relocalization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4762–4769.
3. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; Volume 3, p. 8.
4. Walch, F.; Hazirbas, C.; Leal-Taixe, L.; Sattler, T.; Hilsenbeck, S.; Cremers, D. Image-based localization using lstms for structured feature correlation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 627–637.
5. Furukawa, Y.; Ponce, J. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1362–1376. [[CrossRef](#)] [[PubMed](#)]
6. Gu, F.; Hu, X.; Ramezani, M.; Acharya, D.; Khoshelham, K.; Valaee, S.; Shang, J. Indoor Localization Improved by Spatial Context—A Survey. *ACM Comput. Surv.* **2019**, *52*, 1–35. [[CrossRef](#)]
7. Acharya, D.; Khoshelham, K.; Winter, S. BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images. *ISPRS J. Photogramm. Remote. Sens.* **2019**, *150*, 245–258. [[CrossRef](#)]
8. Acharya, D.; Singha Roy, S.; Khoshelham, K.; Winter, S. Modelling uncertainty of single image indoor localisation using a 3D model and deep learning. *Isprs Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, *IV-2/W5*, 247–254. [[CrossRef](#)]
9. Davison, A.J. Real-time simultaneous localisation and mapping with a single camera. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Nice, France, 14–17 October 2003; Volume 2, pp. 1403–1410.
10. Nister, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 27 June–2 July 2004; Volume 1, pp. 652–659.
11. Acharya, D.; Ramezani, M.; Khoshelham, K.; Winter, S. BIM-Tracker: A model-based visual tracking approach for indoor localisation using a 3D building model. *ISPRS J. Photogramm. Remote. Sens.* **2019**, *150*, 157–171. [[CrossRef](#)]

12. Piasco, N.; Sidibé, D.; Demonceaux, C.; Gouet-Brunet, V. A survey on Visual-Based Localization: On the benefit of heterogeneous data. *Pattern Recognit.* **2018**, *74*, 90–109.
13. Irschara, A.; Zach, C.; Frahm, J.M.; Bischof, H. From structure-from-motion point clouds to fast location recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 May 2009; pp. 2599–2606.
14. Kneip, L.; Scaramuzza, D.; Siegwart, R. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 2969–2976.
15. 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 (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2930–2937.
16. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4654–4662.
17. Cavallari, T.; Golodetz, S.; Lord, N.A.; Valentin, J.; Di Stefano, L.; Torr, P.H. On-the-fly adaptation of regression forests for online camera relocalisation. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4457–4466.
18. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 6684–6692.
19. Wu, J.; Ma, L.; Hu, X. Delving deeper into convolutional neural networks for camera relocalization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5644–5651.
20. Clark, R.; Wang, S.; Markham, A.; Trigoni, N.; Wen, H. VidLoc: A deep spatio-temporal model for 6-dof video-clip relocalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; Volume 3.
21. Ha, I.; Kim, H.; Park, S.; Kim, H. Image retrieval using BIM and features from pretrained VGG network for indoor localization. *Build. Environ.* **2018**, *140*, 23–31. [[CrossRef](#)]
22. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
23. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
24. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
25. Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; Oliva, A. Learning deep features for scene recognition using places database. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 487–495.
26. Zhu, L.; Laptev, N. Deep and confident prediction for time series at uber. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 28–21 November 2017; pp. 103–110.
27. Gal, Y.; Ghahramani, Z. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv* **2015**, arXiv:1506.02158
28. Chollet, F. Keras. 2015. Available online: <https://keras.io> (accessed on 15 August 2020).
29. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 15 August 2020)
30. Khoshelham, K.; Vilariño, L.D.; Peter, M.; Kang, Z.; Acharya, D. The isprs benchmark on indoor modelling. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2017**, *42*, 367–372. [[CrossRef](#)]
31. Ramezani, M.; Acharya, D.; Gu, F.; Khoshelham, K. Indoor positioning by visual-inertial odometry. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2017**, *4*, 371–376.

32. Khoshelham, K.; Tran, H.; Vilariño, L.D.; Peter, M.; Kang, Z.; Acharya, D. An evaluation framework for benchmarking indoor modelling methods. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, *XLII-4*, 297–302. [[CrossRef](#)]
33. Tran, H.; Khoshelham, K.; Kealy, A. Geometric comparison and quality evaluation of 3D models of indoor environments. *ISPRS J. Photogramm. Remote. Sens.* **2019**, *149*, 29–39. [[CrossRef](#)]
34. Volk, R.; Stengel, J.; Schultmann, F. Building Information Modeling (BIM) for existing buildings—Literature review and future needs. *Autom. Constr.* **2014**, *38*, 109–127. [[CrossRef](#)]
35. Porwal, A.; Hewage, K.N. Building Information Modeling (BIM) partnering framework for public construction projects. *Autom. Constr.* **2013**, *31*, 204–214. [[CrossRef](#)]
36. Gal, Y.; Ghahramani, Z. A theoretically grounded application of dropout in recurrent neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 1019–1027.
37. Sehgal, A.; Kehtarnavaz, N. Guidelines and Benchmarks for Deployment of Deep Learning Models on Smartphones as Real-Time Apps. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 450–465. [[CrossRef](#)]
38. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).