

Visual indoor localisation using a 3D building model

by

Debaditya Acharya

(ORCID ID: [0000-0002-6704-5778](https://orcid.org/0000-0002-6704-5778))

Supervised by

Dr Kouros Khoshelham and
Prof Stephan Winter

A thesis submitted in total fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Infrastructure Engineering
THE UNIVERSITY OF MELBOURNE

May 2020

Copyright © 2020 Debaditya Acharya

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the author.

Abstract

WITH the emergence of the global navigation satellite system (GNSS), the performance of outdoor localisation has become excellent over the years. Applications like navigation, location-based services and augmented reality demand seamless localisation capabilities in all environments. However, a single technology that can satisfy the needs of localisation in indoor spaces is absent to date. The major limiting factor in the large scale adoption of indoor localisation systems is the overhead cost of installation and maintenance of dedicated local infrastructure. Consequently, infrastructure-independent indoor localisation has become a focus of research during the past decade.

The ubiquity of smartphones with integrated cameras has resulted in a renewed interest in infrastructure-independent visual localisation approaches for the indoor environments. However, the existing visual approaches face two challenges that restrict the wide applicability of such approaches. Firstly, current visual simultaneous localisation and mapping (SLAM) approaches, where the challenge is the drift caused by the accumulation of errors. Loop closing is a solution to eliminate drift but it poses a limitation in the practical application of visual positioning. For navigation purposes, revisiting the same location might be impractical. The second challenge for the existing visual approaches is the requirement of an initial location. The existing visual approaches that are independent of the initial locations require either the construction of large database of images with known location or a 3D reconstruction of the indoor environment in the form of depth images or 3D point clouds. However, the creation of a database of images with known locations and the acquisition of additional data for large indoor spaces is a challenge due to the cost, time and post-processing involved.

This research presents approaches to address the two above-mentioned challenges of the existing visual approaches by using a 3D building model that is usually available

through the building information modelling process or can be generated with little effort from existing 2D plans. The motivation of using BIM for this research comes from the fact that BIMs of modern buildings are readily available, as they are jointly maintained by the constructors and facility managers.

The research can be broadly classified into two parts. The first part proposes a novel 3D model-based visual tracking approach called BIM-Tracker. BIM-tracker uses the 3D building model to perform a drift-free localisation and addresses the challenge of accumulation of error. Localisation is performed by integrating image sequences captured by a camera, with the 3D building model. A comprehensive evaluation of the approach with photo-realistic synthetic datasets shows the robustness of the localisation approach under challenging conditions. Additionally, the approach is evaluated on real data captured by a smartphone, and achieves an accuracy of ten centimetres. Similar to the requirement of many visual approaches, BIM-Tracker depends on the availability of the initial location.

The second part of the research proposes a deep learning-based method called BIM-PoseNet to estimate the initial location. The requirement of image-based reconstruction of the indoor environment is eliminated by using a 3D building model, thereby addressing another challenge of the existing visual approaches that estimate the initial location. BIM-PoseNet is based on training a CNN with synthetic images obtained from the 3D indoor model to regress the location of a real image taken by a camera. In addition, the uncertainties of camera location estimates are modelled by adopting a Bayesian CNN, as uncertainty provides an indication of confidence and trust in an estimated location in the absence of ground truth. Furthermore, the use of sequences of synthetic images is explored to exploit the spatio-temporal information from the images to improve the performance of BIM-PoseNet by using recurrent neural networks.

The results of the qualitative and quantitative experimentation of the proposed approaches with photo-realistic synthetic and real datasets indicate the proposed research addresses the two major limitations of the existing visual indoor localisation approaches. In addition, the proposed research demonstrates the potential of visual indoor localisation as a single technology for achieving an infrastructure-independent localisation.

Declaration

This is to certify that

1. the thesis comprises only my original work towards the degree of Doctor of Philosophy,
2. due acknowledgement has been made in the text to all other material used,
3. the thesis is less than 80,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed: Debaditya Acharya

Date: 15.05.2020

Acknowledgements

I'd like to express my sincere gratitude to Dr Kourosh Khoshelham for his comprehensive guidance throughout the length of the PhD program, and for teaching the art of performing and documenting research work. I acknowledge his immense support for involving me with several research projects of the university that were out of the curriculum, which in turn helped me to gain knowledge outside my own research area. I'll also thank him for encouraging collaborative research in the group that helped me to create a better network and resulted in several collaborative research publications. Besides, I thank him for allowing me to supervise half a dozen of master students that in turn broaden my knowledge of spatial science altogether. Lastly, I would express my gratitude for his motivation and time that helped me polish my skills and achieve my career goals.

I'll acknowledge Prof Stephan Winter for his invaluable supervision in writing and polishing the scientific works, and for providing moral support throughout the length of my candidature. A special thanks to Dr Martin Tomko for chairing my PhD committee and for providing valuable feedback of the research idea.

I express my gratitude to The University of Melbourne for supporting my work with a Melbourne Research Scholarship. Also, I acknowledge several travel grants from the Melbourne School of Engineering and the Department of Infrastructure Engineering that supported my conference travels. I'm grateful to the Melbourne School of Engineering, The University of Melbourne, for supporting the work with Research Engagement Grants. Moreover, I acknowledge the use of LIEF HPC-GPGPU Facility (LIEF Grant LE170100200) of SPARTAN - the High Performance Computing platform of the University of Melbourne, for my research .

I acknowledge my colleagues and my friends especially Milad Ramezani, Fuqiang Gu and Ha Tran for their valuable inputs related to my work, which resulted in joint publications. I thank my master student Weilin Yan for his excellent work that resulted in a local conference publication and subsequently helped me bagging the Young Author award from Australian Spatial Information Education and Research Association (ASIERA). I also thank Lawrence Molloy for engaging me with the research engagement projects under the Melbourne School of Engineering, MSE2025 initiative program. Lastly, I thank George Fox, Pauline Woolcock and Eileen Doufas Shea from the Department of Infrastructure Engineering for their help and aid regarding procurement and management of equipments.

Finally, I would like to express my hearty gratitude towards my parents, Deb Kumar Acharya and Tamali Acharya, who provided moral support throughout my life. I acknowledge their support that motivated me to move into research and academia, leaving the industry where my father is working for over four decades. A final word of love and gratitude for my wife Sesa Singha Roy, who is responsible for my overall well-being in Melbourne. I thank her for taking good care and keeping me happy during the low points of my life and career. Besides, she helped me with coding, writing and preparing presentations, and this is in addition to the delicious food that she makes. My journey of PhD would have been completely impossible without her aid.

Debaditya Acharya

Change is the end result of all true learning

~Leo Buscaglia

Contents

1	Introduction	3
1.1	Infrastructure-independent visual indoor localisation	4
1.2	Challenges of existing visual localisation approaches	4
1.3	Research objectives	5
1.3.1	Research objective 1	6
1.3.2	Research objective 2	8
1.3.3	Research objective 3	8
1.3.4	Research objective 4	8
1.4	Contributions	9
1.4.1	BIM-Tracker	9
1.4.2	BIM-PoseNet	10
1.4.3	Bayesian BIM-PoseNet	10
1.4.4	Recurrent BIM-PoseNet	11
1.4.5	BIM and photo-realistic synthetic image datasets	11
1.5	Thesis structure	12
2	Background and related work	15
2.1	Overview of indoor localisation approaches	15
2.2	Visual localisation approaches dependent on initial location	17
2.2.1	Visual odometry	17
2.2.2	Visual SLAM	18
2.2.3	3D model-based visual tracking	19
2.3	Visual localisation methods independent of initial location	21
2.3.1	Deep learning and feature engineering	22
2.3.2	Camera pose estimation using convolutional neural networks	22
2.3.3	Modelling uncertainty of camera pose using Bayesian convolutional neural networks	23
2.3.4	Recurrent neural networks and the use of spatio-temporal information	24
2.3.5	Use of synthetic images in deep learning	26
2.4	Building information modelling for visual localisation	27
2.5	Summary	28
3	BIM-Tracker: A model-based visual tracking approach using a 3D building model	31
3.1	Introduction	31
3.2	Methodology	33
3.2.1	Pose initialisation	34
3.2.2	Determination of visible edges of the 3D building model in camera views	35
3.2.3	Line representation	36
3.2.4	Point sampling of the model edges	36
3.2.5	Projecting model edges to the image	39

3.2.6	Creating 3D to 2D point correspondences using edges	40
3.2.7	Pose estimation by non-linear least square minimisation	41
3.2.8	Outlier removal with MSAC	43
3.2.9	Integration of a Kalman filter	47
3.2.10	Reusing inter-frame correspondences	50
3.3	Experiments and results	51
3.3.1	Datasets used	51
3.3.2	Experiments with photo-realistic synthetic data	56
3.3.3	Experiment with real data	65
3.3.4	Computational times	67
3.3.5	Region of convergence	67
3.3.6	Discussions and future directions	69
3.4	Conclusions	70
4	BIM-PoseNet: Camera localisation using a 3D building model and deep learning	75
4.1	Introduction	75
4.2	Methodology	76
4.2.1	Network architecture and fine-tuning	77
4.2.2	Use of 3D building model	79
4.3	Experiments and results	80
4.3.1	Dataset used	81
4.3.2	Baseline performance using real images	84
4.3.3	Fine-tuning with synthetic images	87
4.3.4	Effects of level-of-detail of 3D building models on localisation	98
4.3.5	Discussion and future directions	99
4.4	Conclusions	101
5	Bayesian BIM-PoseNet: Modelling localisation uncertainty of BIM-PoseNet	105
5.1	Introduction	105
5.2	Methodology	106
5.2.1	Rendering synthetic images from a 3D building model	106
5.2.2	Fine-tuning a Bayesian network	108
5.3	Experiments and results	109
5.3.1	Dataset	110
5.3.2	Baseline performance using real images	110
5.3.3	Estimating uncertainty with synthetic images	114
5.3.4	Discussion and future directions	118
5.4	Conclusions	119
6	Recurrent BIM-PoseNet: Learning pose estimation from synthetic image sequences	123
6.1	Introduction	123
6.2	Methodology	125
6.2.1	Deep learning Architecture	125
6.2.2	Fine-tuning and loss function	128
6.2.3	Modelling uncertainty	128
6.3	Experiments and results	129
6.3.1	Implementation details	129
6.3.2	Experimental design	130
6.3.3	Dataset	130
6.3.4	Experiment 1: Baseline performance using real image sequences	131
6.3.5	Experiment 2: Performance of the network fine tuned using synthetic images	134
6.3.6	Experiment 3: Modelling uncertainty using synthetic images	141

6.3.7	Computation times	143
6.4	Conclusions	144
7	Conclusions	149
7.1	Achieved research objectives	150
7.2	Limitations and future directions	151
7.3	Recommendation and applications	153

List of Figures

1.1	The proposed framework for visual indoor localisation	7
2.1	The classification of localisation approaches (after Alarifi et al., 2016)	16
2.2	An example of 3D model-based visual tracking. The left image shows the projection of initial camera pose and the building edges on the image. The right image shows the overlay of the estimated camera pose after a few iterations.	19
2.3	A deep convolutional neural network (VGG), after Wang et al. (2017b).	23
2.4	An example of dropped out connections between neurons. The orange lines show the connections and the blue dotted lines show the dropped out connections. . . .	24
2.5	An example of synthetic images with textured generated from the same viewpoint inside a building, using a BIM. The left image shows the real image, and the right image shows the synthetic image.	26
2.6	A BIM of a building containing the semantic properties (visualised in BIM Vision).	27
3.1	The workflow of BIM-Tracker.	34
3.2	The advantage of segmenting edges as compared to using end points to represent lines. Left: A line is represented by end points and the search for corresponding edges misses out the actual matches. Right: A line is segmented into points that allows additional searches to create correct matches.	37
3.3	Point sampling of the model edges and creating correspondences after back-projection. A'B' is the back-projection of the 3D model edge to the image plane. CD is the corresponding edge to A'B', and EF is a stray edge. Blue dotted lines are the search lines from the sampled points (blue circles). There are two potential correspondences for the points on A'B' on either side of the line. Taking a single hypothesis would have created two wrong correspondences to the edge EF, as the edge EF is closer to A'B' compared to edge CD.	38
3.4	AB is an image edge, whereas CD and C'D' are the back-projected model edges using different poses. It can be seen that the projection of the back-projected sampled point for edge C'D' on AB is further away as compared to the edge CD. This difference creates a bad pose estimation during large translations and rotations. . .	46
3.5	The created BIM from point cloud data. The ceiling is removed for better visualisation, but tube lights are retained. The green line denotes the trajectory of the virtual cameras.	53
3.6	An image frame from the set (a) Set 2 - having 90° FoV (b) Set 4 - having 60° FoV (c) Set 5 - having 120° FoV (d) Set 6 - in portrait mode (e) Set 7 - containing walking humans and potted plants as occlusions (f) Set 8 - having motion blur.	54
3.7	The cumulative distribution function (CDF) of errors for (a) translation and (b) rotation to evaluate the effects of image resolution.	57
3.8	The Cumulative distribution function (CDF) of errors for (a) translation and (b) rotation to evaluate the effects of camera FoV.	58

3.9	The Cumulative distribution function (CDF) of errors for (a) translation and (b) rotation to evaluate the effects of occlusion and motion blur.	59
3.10	The distribution of errors along the trajectory. The colour represents the value of the average normalised error for the sets having a FoV of 90° . Below are the corresponding views of the points on the trajectory.	62
3.11	Localisation error calculated with respect to ground truth for all the frames of the trajectory.	62
3.12	The per frame RMS errors with respect to the ground truth for ORB-SLAM and BIM-Tracker for the synthetic baseline dataset (Set 2).	63
3.13	The cumulative distribution function (CDF) of RMS errors with respect to the reference trajectory for ORB-SLAM and BIM-Tracker.	63
3.14	(a) The low level-of-detail model used for the tracking, containing only walls, floor, ceiling and doors. The geometry of the wall-door intersection was kept, and other details, such as the frame of the door was simplified (b) Projection of a banner for evaluating the estimated pose for augmented reality applications	65
3.15	The reference trajectory and the trajectories estimated by ORB-SLAM and BIM-Tracker. The door is represented in light blue colour.	66
3.16	The region of convergence for pose initialisation. A red dot indicates convergence, whereas the absence of red dot indicates the region of failure of tracking. Euclidean distances are used to calculate the locational differences and Euler angles are used to calculate the angular differences.	68
3.17	(a) The effect of Kalman filter. The Kalman filter was able to predict a better pose (green colour) compared to the pose estimated from the previous frame by Gauss-Newton optimisation (red colour) for a blurry frame. (b) A severely blurry frame during a turn having one model edge visible (c) A frame having one model edge visible (d) A frame having no model edge visible.	69
4.1	The design of BIM-PoseNet.	78
4.2	Different sets of synthetic images rendered from the 3D building model.	83
4.3	Generation of additional images on the trajectory by rotating the camera along Y and Z axes.	83
4.4	A real image (left) and its gradient intensity image (right).	85
4.5	The variation of the locational and angular errors for the baseline experiment to determine the ideal value of β for the scene.	85
4.6	The saliency map predicted by a network fine-tuned with real images, overlaid on the test image. The red colour indicated the pixels in the image that the network considers important for pose regression.	86
4.7	Validation results for the baseline experiment in which the network was trained with real images and tested with real images. The red line represents the ground truth trajectory estimated by the SfM approaches. The estimated locations by the fine-tuned network corresponding to the points in the trajectory are plotted in dark blue to yellow colours, where the colour provides a measure of error of the points.	86
4.8	The validation results for the networks (a) fine-tuned with <i>Syn-car</i> and tested on <i>Syn-car</i> (b) fine-tuned with <i>Syn-pho-real</i> and tested on <i>Syn-pho-real</i> (c) fine-tuned with <i>Syn-pho-real-tex</i> and tested on <i>Syn-pho-real-tex</i> (d) fine-tuned with <i>Gradmag-Syn-car</i> and tested on <i>Gradmag-Syn-car</i> (e) fine-tuned with <i>Syn-edge datasets</i> and tested on <i>Syn-edge datasets</i> . D1 and D2 represent the locations of two doors in the corridor.	89

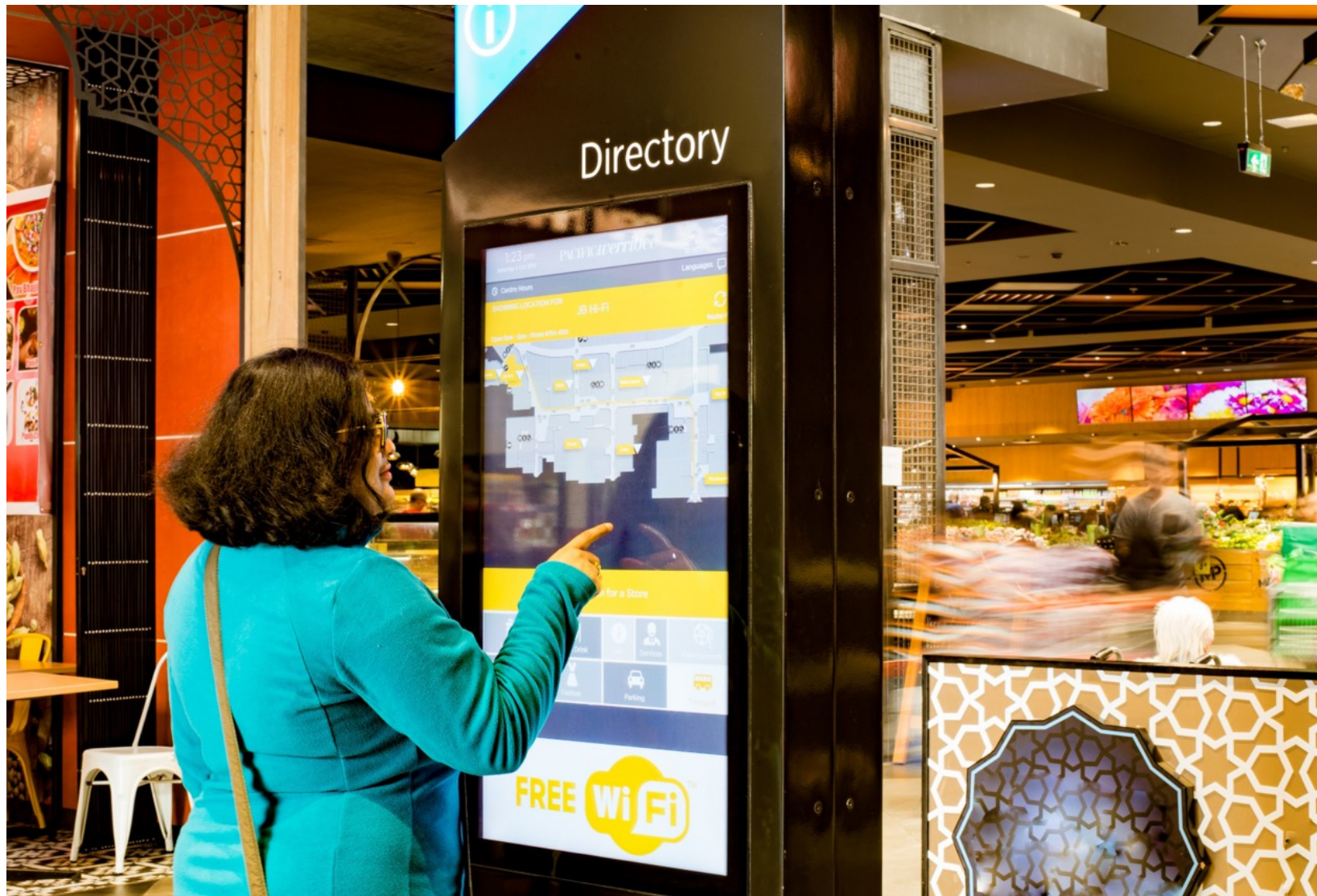
4.9	Estimated locations by the network (a) fine-tuned with <i>Syn-car</i> and tested on real images (b) fine-tuned with <i>Syn-pho-real</i> and tested on real images (c) fine-tuned with <i>Syn-pho-real-text</i> and tested on real images (d) fine-tuned with <i>Gradmag-Syn-car</i> and tested on <i>gradmag</i> of real images (e) fine-tuned with <i>Syn-edge</i> and tested on edge <i>gradmag</i> of real images.	90
4.10	The saliency maps of (a) - (c) a real image and (d) - (e) edge <i>gradmag</i> of real image predicted by networks fine-tuned with synthetic image datasets. The red colour shows the pixels in the image the network considers important for pose regression.	91
4.11	Overlay of the predicted pose on real images by the network fine-tuned on <i>Syn-edge</i> dataset. The values in the bracket represent the errors.	91
4.12	Visualising error for the differnt parts of the trajectory. The colour represents the magnitude of the errors. The maximum error is present (red colour) in the part AB of the trajectory. In the part CD of the trajectory, the errors are consistently larger compared to other parts of the trajectory.	93
4.13	Visualising frames with high errors. The first row of images shows the real images of the frames. The second row shows the respective edge <i>gradmag</i> of the images and the third row shows the images of the <i>Syn-edge</i> dataset with the same pose of the virtual camera. Frame 164, 233 and 274 shows the effects of motion blur, which results in low values of edge <i>gradmag</i> for the blurry areas. Frame 491 and 591 shows the effects of artefacts in the edge <i>gradmag</i> of real images, due to structural differences, such as the presence of posters, notice boards and fire-extinguisher. Errors are shown in brackets.	94
4.14	Generation of a low level-of-detail 3D building model (Right) from the 3D building model derived from BIM (Left) by simplfying geometries. The colours are added to the roof (green), walls (white), doors (orange) and floor (gray) for better visualisation.	95
4.15	Different sets of synthetic images rendered to evaluate the effects of level-of-detail of a 3D building model on the errors of localisation.	95
4.16	The results of the estimated locations of real images and edge <i>gradmag</i> of real images on fine-tuned networks on (a) <i>Syn-edge-lights</i> (b) <i>Syn-edge-reduced</i> and (c) <i>Syn-edge-reduced-lights</i>	96
4.17	The saliency maps of the edge <i>gradmag</i> of real images as predicted by the networks fine-tuned with (a) <i>Syn-edge-lights</i> (b) <i>Syn-edge-reduced</i> (c) <i>Syn-edge-reduced-lights</i> datasets. The red colour represnts the image pixels that the network considers important for the task of pose regression.	96
4.18	Left: Door D1 and Right: Door D2. An example of perceptual aliasing resulting in large errors in two test images containing two different doors whose appearance is identical.	100
5.1	The design of Bayesian BIM-PoseNet.	107
5.2	The variation of locational and rotational errors with β	111
5.3	The variation of locational and rotational uncertainties with β . R denotes correlation coefficient	112
5.4	Baseline performance of a network fine-tuned and validated using real images with $\beta = 500$. (a) The distribution of the estimated camera poses. The red line shows the ground truth trajectory estimated using SfM approaches. The colours represent the magnitude of the errors corresponding to each point. The locational and rotational errors are in metres and degrees respectively. (b) locational uncertainty vs locational errors (c) rotational uncertainty vs rotational errors	112

5.5	The results of the network (a) fine-tuned with Syn-car and tested on real images (b) fine-tuned with Syn-pho-real and tested on real images (c) fine-tuned with Syn-pho-real-tex and tested on real images (d) fine-tuned with Gradmag-Syn-car and tested on edge gradmag of real images (e) fine-tuned with Syn-edge and tested on edge gradmag of real images. Uncertainty plots for (f) Syn-car (g) Syn-pho-real (h) Syn-pho-real-tex (i) Gradmag-Syn-car and (j) Syn-edge datasets.	113
5.6	Predicted camera poses by a network fine-tuned with Gradmag-Syn-car dataset and tested with edge gradmag of real images.	116
5.7	(a) shows the average normalised errors of the trajectory (b) shows the average normalised uncertainties of the trajectory and (c) shows corresponding points on the trajectory with high errors.	117
5.8	(a) Location uncertainty vs error (b) Rotational uncertainty vs errors for Gradmag-Syn-car images.	118
6.1	The design of the proposed approach. The network used is GoogleNet (Szegedy et al., 2015) containing nine inception modules. The output of the final classification branch yields the location feature vector which serves as input to the LSTM layer. .	126
6.2	The predictions of Recurrent 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.	131
6.3	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.	132
6.4	Cumulative distribution function (CDF) of the (a) errors and (b) inter-frame distances of the predictions by BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for the baseline performance using real images.	133
6.5	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.	135
6.6	Estimated camera poses by the proposed Recurrent BIM-PoseNet (a) fine-tuned using Syn-car and tested on real images (b) fine-tuned using Syn-pho-real and tested on real images (c) fine-tuned using Syn-pho-real-tex and tested on real images (d) fine-tuned using Gradmag-Syn-car and tested on gradmag of real images (e) fine-tuned using Syn-edge and tested on edge gradmag of real images.	137
6.7	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 Point A - E. Table 6.3 shows the errors and estimated uncertainties of the frames.	139
6.8	Cumulative distribution function (CDF) of the (a) localisation errors and (b) inter-frame distances of the predictions by BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for Gradmag-Syn-car dataset.	140
6.9	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.	142

7.1	An example of semantic segmentation (right image) of an indoor scene (left image), after Zheng et al. (2019).	152
-----	--	-----

List of Tables

3.1	The details of each set of data	52
3.2	The details of the camera parameters	52
3.3	Error for each set of experiments	52
3.4	The average time for each step of localisation	67
4.1	The details of each set of synthetic data with the rendering times of each frame. . .	83
4.2	The test errors of the fine-tuned networks on real images (training dataset with 475 images) and tested with real images (test dataset with 475 images).	87
4.3	The number of synthetic images used for fine-tuning the five networks to validate with synthetic images.	87
4.4	The fine-tuning errors for the five networks with respect to validation datasets. . .	88
4.5	The test errors of the three networks on real images (third column) and the test errors on the edge <i>gradmag</i> of the real images (fourth column).	88
4.6	The number of synthetic images used to fine-tune the five networks for testing with real images/ <i>gradmag</i> of real images	92
4.7	The test errors the three fine-tuned networks on edge <i>gradmag</i> of real images to determine the effects of level-of-detail of the 3D building model on their pose regression abilities.	93
5.1	Errors and uncertainties corresponding to Figure 5.6. The unit of the rotational uncertainty is scalar dimensionless number.	115
5.2	Errors and uncertainties corresponding to Figure 5.7 (c)	118
6.1	The comparison of the median errors and inter-frame distances of BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for the baseline experiment. . .	133
6.2	The comparison of the errors of BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for the five networks fine-tuned using the different synthetic image datasets and tested using real images.	136
6.3	The comparison of the errors and estimated uncertainties for the frames of Figure 6.7.	139
6.4	The comparison of error and uncertainties for Recurrent BIM-PoseNet with Bayesian BIM-PoseNet.	141
6.5	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.	144



"... infrastructure-independent indoor localisation can change the way we navigate in indoor spaces ..."

≈ South Pacific Werribee Mall, Werribee, Victoria, Australia
(Courtesy: Sesa Singha Roy)

Chapter 1

Introduction

INDOOR localisation is the key enabler of a range of applications including navigation guidance, location-based services, emergency response, and augmented reality. Indoor environments present challenges for localisation due to strong attenuation of global navigation satellite system (GNSS) signals ([Mautz, 2012](#)) compared to the outdoor environments. In the literature, many different approaches for indoor localisation have been proposed. However, the performance of indoor localisation is still lagging behind compared to outdoor localisation ([Lymberopoulos and Liu, 2017](#)), and many localisation applications are waiting for an acceptable solution.

Localisation is the process of finding the location of an object on a map or in a reference coordinate system, and is often used in the literature synonymous to positioning. *Visual localisation*, or finding the location of a camera by using visual methods is an active research area in computer vision. For the scope of this thesis, visual localisation is defined as finding the location (and orientation) of a camera with respect a 3D building model that represent the physical environment of a university corridor. Other terms that are being used to describe localisation include *6-DoF* (degree of freedom) *camera pose* representing location (three coordinates) and orientation (three rotation angles), and *spatio-temporal localisation* where the spatial and temporal information have been used to perform visual localisation. The following subsections introduce the importance of an infrastructure-independent visual indoor localisation technology, in addition to discussing the challenges of the existing visual localisation approaches. Consequently the research objectives are presented which is followed by the contributions of the thesis.

1.1 Infrastructure-independent visual indoor localisation

Present indoor localisation approaches, such as radio frequency identification (RFID) technology or Wifi fingerprinting, are infrastructure-dependent, whose installation and maintenance are costly. A dedicated network of sensors, transmitters or beacons installed in the indoor environment ([Alarifi et al., 2016](#)) is referred to infrastructure in this thesis. The widespread availability of smartphones with cameras has resulted in a renewed interest in visual localisation approaches that are infrastructure-independent. However, the existing visual localisation approaches suffer from drawbacks that limit the applicability of the approaches for mass-market applications, and improvements in indoor localisation systems are likely to generate better prospects for the industry. Consequently, infrastructure-independent visual indoor localisation has become a focus of research and development during the past decade.

1.2 Challenges of existing visual localisation approaches

The existing visual localisation approaches, such as simultaneous localisation and mapping (SLAM) and visual odometry have been applied to many computer vision applications, however, there are two major limitations:

- Firstly, SLAM and visual odometry suffer from the problem of accumulation of localisation errors, resulting in a drift of the estimated trajectory ([Khoshelham and Ramezani, 2017](#); [Scaramuzza and Fraundorfer, 2011](#); [Caron et al., 2014](#)). The drifts are often recalibrated by other techniques, e.g. SLAM is recalibrated by a loop closure ([Piasco et al., 2018](#)), and visual odometry is recalibrated using visual landmarks ([Zhu et al., 2007](#)). In practice, however, it may not always be feasible to recalibrate the error using loop closing or landmark detection, which results in large drifts. For instance, in contrast to a robot, the user cannot be forced to take loops, for navigation purposes. Also, these approaches are known to fail in environments characterised by poor texture or containing repetitive patterns.

- Secondly, visual odometry, SLAM and other visual approaches require the knowledge of initial camera *pose* (location and orientation), prior to localisation. This limitation further elevates in the case of a localisation failure, hence, a localisation approach is desirable that is independent of the initial camera pose. The visual approaches that can provide initial camera pose from single images are image retrieval or camera pose regression approaches, for instance using convolutional neural networks (CNN) (Piasco et al., 2018). These approaches either require a database of images with known camera poses (Arandjelovic and Zisserman, 2012; Radenović et al., 2016) or additional data sources such as 3D point clouds (Irschara et al., 2009) or depth data of the scene (Shotton et al., 2013). However, performing a 3D image-based reconstruction of the environment for large indoor spaces is a challenge due to the cost, time and post-processing involved.

In addition, uncertainty plays an important role in assessing the quality of camera pose estimates in the absence of ground truth (Kendall and Cipolla, 2016). Hence, modelling uncertainties of camera pose estimates is an integral part of any localisation approach, and is used to remove the erroneous estimates.

1.3 Research objectives

Considering the ubiquity of smartphones with cameras, this research proposes an infrastructure-independent visual localisation framework, with the aim of developing a technology that can be universally applicable in indoor spaces. The *key idea* of the localisation framework is to use a 3D building model obtained from *building information modelling* (BIM) to address the two aforementioned challenges of the visual localisation approaches in the following ways:

- Firstly, this thesis aims to develop a drift-free localisation approach using a 3D building model, which does not depend on loop closure or visual landmarks for operation, and is able to operate in environment with poor texture or repetitive repetitive patterns.

- Secondly, this thesis aims to eliminate the requirement of 3D image-based reconstruction for providing the initial camera pose, using a 3D building model to estimate the initial camera pose and model the uncertainties of the camera pose estimates.

The base assumption of the research is that 3D wire-frame models (referred to as *3D building models* henceforth) can be obtained from BIM of the indoor spaces where the localisation is planned. This assumption is practical, as for all modern buildings, BIMs are jointly created and maintained by the constructors and the facility managers.

Figure 1.1 shows the proposed localisation framework, where a 3D building model is used to perform visual localisation. BIM-Tracker is developed to perform accurate camera localisation, with errors less than ten centimetres, and is introduced in details in Chapter 3. BIM-PoseNet provides a coarse initial camera pose, with errors approximately under two metres, and is discussed in details in Chapter 4. In addition, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet provide a measure of uncertainties of the estimated camera poses using a single image or a sequence of images respectively, and are discussed in details in Chapter 5 and 6. The achievable accuracy of BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet are sufficient for applications like pedestrian navigation and can be used independently of BIM-Tracker. The following subsection discusses the four research objectives that help to achieve the key idea of the research.

1.3.1 Research objective 1

For addressing the problem of accumulation of errors, the images taken from a smartphone camera are compared with the corresponding view of the 3D building model for localisation. The edges in the images are matched with the edges derived from the 3D building model to estimate the camera pose with respect to the BIM coordinate system. The following research objective is defined:

- To achieve centimetre level accuracy without any drift by integrating image information with a 3D building model.

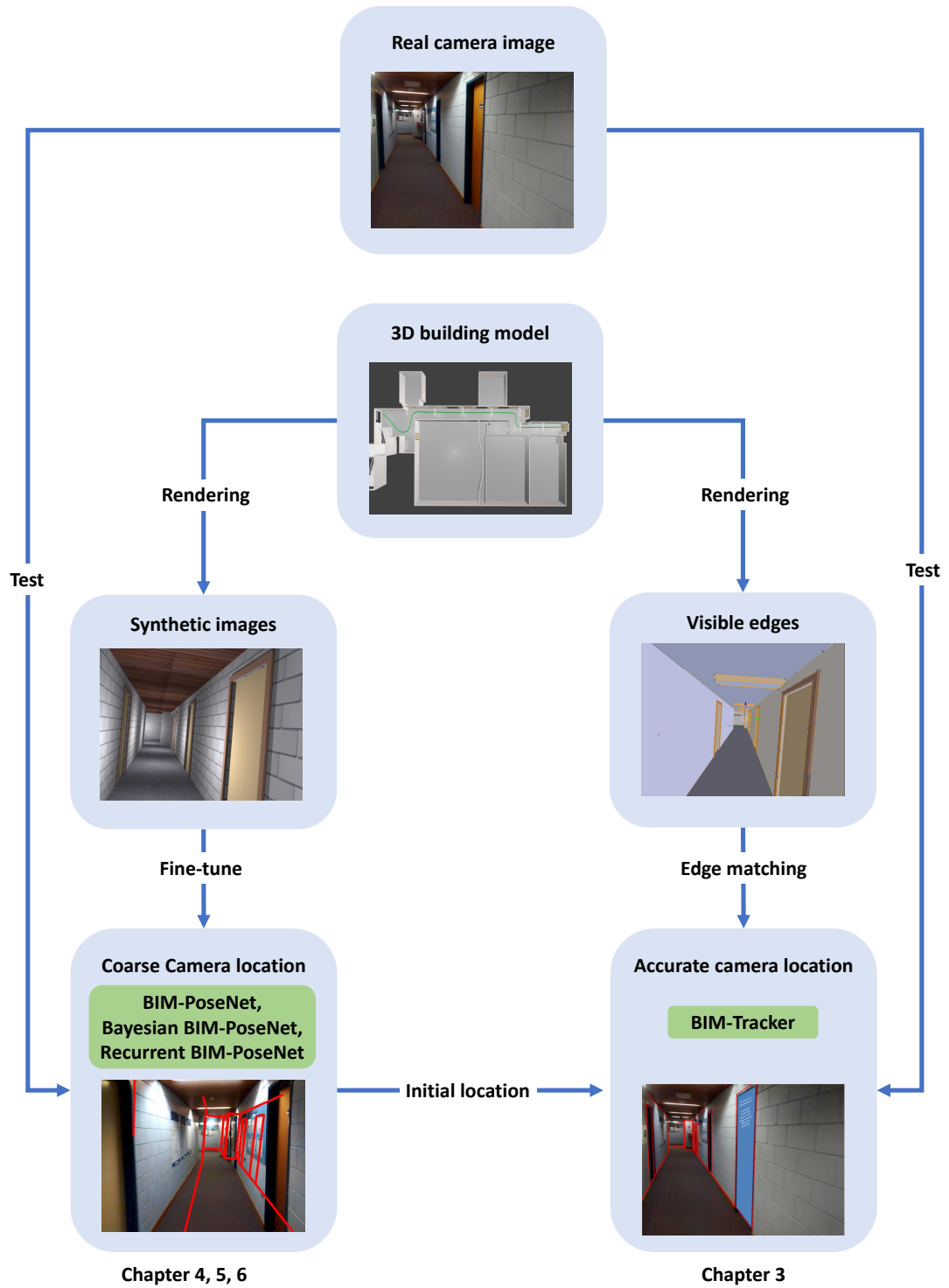


Figure 1.1: The proposed framework for visual indoor localisation

1.3.2 Research objective 2

Existing pose regression networks, such as *PoseNet* (Kendall et al., 2015) learn from real images of known camera poses obtained using structure-from-motion (Furukawa and Ponce, 2010) (SfM) approaches. The SfM approaches involve capturing thousands of overlapping images of the whole environment, and perform 3D image-based reconstruction of the environment, which is challenging for large indoor spaces. Therefore, to address the challenge of performing SfM approaches and hence to eliminate the requirement of 3D image-based reconstruction of the environment, the following research objective is defined:

- To perform camera pose regression of real images without performing a 3D image-based reconstruction of the indoor environment by using synthetic images obtained from the 3D building model.

1.3.3 Research objective 3

In the literature, Bayesian CNNs (Kendall and Cipolla, 2016) have been used to model the uncertainty of the camera pose estimates for real images. Similar to the CNN pose regression networks, the drawback of using such networks is the requirement of 3D image-based reconstruction of the environment. Hence, to address the drawback the following research objective is defined:

- To model the uncertainty of camera pose estimates by pose regression using CNN and a 3D building model.

1.3.4 Research objective 4

The spatio-temporal information of a sequence of real images has been used for improving the performance of pose regression networks by using recurrent CNNs (Walch et al., 2017; Clark et al., 2017). However, the performance of recurrent networks to estimate the pose of real images and to model the uncertainty of estimations when learning from photo-realistic synthetic images is an unexplored research area. Therefore, this part of

the research evaluates the possibility of recurrent CNNs to regress the camera pose of a sequence of real images by learning from a sequences of photo-realistic synthetic images. Hence, the requirement of 3D image-based reconstruction of the environment is eliminated, and the camera poses of the sequence of real images is estimated. An experimental set-up is designed to address the following research objective:

- To reduce the errors of pose regression with real images for the networks that learn from synthetic images, by exploiting the spatio-temporal information of image sequences using recurrent CNNs.

1.4 Contributions

To achieve the above-mentioned objectives, the research proceeds with the following five contributions.

1.4.1 BIM-Tracker

To achieve objective 1, a 3D building model is obtained from a BIM, which is then used to create a photo-realistic synthetic image dataset. This dataset is used to perform a comprehensive evaluation of BIM-Tracker under challenging conditions. The MATLAB code for implementing BIM-Tracker is shared at <https://github.com/debaditya-unimelb/BIM-Tracker> and a video demonstration is presented at the URL <https://youtu.be/cq7mk4mfdRA>. Additionally, BIM-Tracker is tested on real data captured by a smartphone's camera to demonstrate the approach is drift-free and addresses a challenge in the literature. The results show that an accuracy of localisation under ten centimetres can be achieved. The following refereed journal publication summarises the main finding of BIM-Tracker, and Chapter 3 has been adapted from the publication:

- **Acharya, D.**, Ramezani, M., Khoshelham, K. and Winter, S., 2019. BIM-Tracker: A model-based visual tracking approach for indoor localisation using a 3D building model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150, 157-171.

1.4.2 BIM-PoseNet

To achieve research objective 2, the 3D building model is used to generate several types of photo-realistic synthetic image datasets. The synthetic and the real datasets with their corresponding ground truth poses are available at https://melbourne.figshare.com/articles/UnimelbCorridorSynthetic_zip/10930457. Subsequently, each type of synthetic image is evaluated for its learning ability with BIM-PoseNet to predict the camera poses of real images. Besides, a low level-of-detail 3D building model is generated from the 3D building model to assess the localisation performance of BIM-PoseNet for different levels of detail of the 3D building model. The results indicate that a localisation accuracy under two metres can be achieved by BIM-PoseNet. The following referred journal publication reports the results of the experiments, and Chapter 4 has been adapted from the publication:

- **Acharya, D.**, Khoshelham, K. and Winter, S., 2019. BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images. *ISPRS journal of photogrammetry and remote sensing*, 150, 245-258.

1.4.3 Bayesian BIM-PoseNet

To achieve research objective 3, the synthetic images generated for evaluating BIM-PoseNet are used to assess the learning ability of a Bayesian CNN to regress the camera pose of real images. The uncertainties associated with the estimated camera poses are modelled, thus providing a measure with which the estimations can be trusted. The results of the experiments indicate that a correlation exists between the modelled uncertainties and the errors, and accuracy of under two metres can be achieved. The following refereed publication summarises the results of the experiments, and Chapter 5 has been adapted from the publication:

- **Acharya, D.**, Singha Roy, S., Khoshelham, K. and Winter, S., 2019. Modelling uncertainty of single image indoor localisation using a 3D model and deep learning. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5, 247-254.

1.4.4 Recurrent BIM-PoseNet

Lastly, to achieve research objective 4, the learning ability of Recurrent BIM-PoseNet is evaluated with sequences of synthetic images to perform camera pose regression with sequences of real images. A Keras implementation of Recurrent BIM-PoseNet can be found at <https://github.com/debaditya-unimelb/RecurrentBIM-PoseNet>. The results of the experiments suggest that an accuracy of approximately 1.6 metres is achieved, and the estimated camera poses result in a smoother trajectory as compared to BIM-PoseNet and Bayesian BIM-PoseNet. Also, the evaluation of the trajectory shows that the uncertainties of the estimated camera poses are correlated with the errors. A manuscript has been submitted for consideration in a journal for publication, and Chapter 6 has been adapted from the text of the submitted manuscript.

1.4.5 BIM and photo-realistic synthetic image datasets

The research involved the creation of a BIM that is used to generate photo-realistic synthetic images, and the BIM serves as a benchmark dataset for indoor modelling. The dataset can be found at the URL <http://www2.isprs.org/commissions/comm4/wg5/benchmark-on-indoor-modelling/datasets.html>. The photo-realistic synthetic images, and the benchmark dataset contributed to the following publications:

- Gu, F., Hu, X., and Ramezani, M., and **Acharya, D.**, and Khoshelham, K., and Valaee, S. and Shang, J., 2019. Indoor Localization Improved by Spatial Context-A Survey. *ACM Computing Surveys*, 52(3), 64:1-35.
- Khoshelham, K., Tran, H., Daz-Vilario, L., Peter, M., Kang, Z. and **Acharya, D.**, 2018. An evaluation framework for benchmarking indoor modelling methods. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, XLII-4, 297-302.
- Khoshelham, K., Vilario, L.D., Peter, M., Kang, Z. and **Acharya, D.**, 2017. The ISPRS benchmark on indoor modelling. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, XLII-2/W7, 367-372.

- Ramezani, M., **Acharya, D.**, Gu, F., and Khoshelham, K., 2017. Indoor positioning by visual-inertial odometry. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4, 371-376.

1.5 Thesis structure

Chapter 2 provides a review on the existing visual indoor localisation approaches and their limitations in details. Chapter 3, 4, 5 and 6 summarises the main findings of BIM-Tracker, BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet, respectively, and have been adapted from the mentioned publications. Chapter 7 concludes the main findings and points towards the future directions of the research.

Chapter 2

Background and related work

This chapter reviews the existing approaches to indoor localisation and the motivations of using visual indoor localisation. The contents can be broadly divided into two parts. One part discusses the related approaches that are dependent on an initial location for their operation and the other part discusses the approaches that are independent of the initial location. Subsequently, the limitations of both the visual approaches are discussed and the possible ways to mitigate the limitations are introduced.

LOCALISATION is the process of estimating the location of a person or an object on a map or in a reference coordinate system, and finds a wide range of applications that includes navigation, location-based services, and robotics ([Yassin et al., 2017](#)). In the literature, several indoor localisation approaches have been proposed considering the vastness of the indoor environments and the absence of a practical indoor localisation approach. The following paragraphs provides an overview of such approaches. Some of the presented approaches in this chapter are specific to indoor localisation, however, there are studies that focus both on indoor and outdoor localisation. The proposed approaches in the thesis are specific to indoor localisation, and the adaptation to outdoor scenes is beyond the scope of the research.

2.1 Overview of indoor localisation approaches

The indoor localisation approaches can be classified either as infrastructure-dependent or infrastructure-independent ([Alarifi et al., 2016](#)). In this thesis, infrastructure is defined as a dedicated network of sensors, transmitters or beacons installed in the indoor environment, such as a network of Wifi access points, radio frequency identification (RFID)

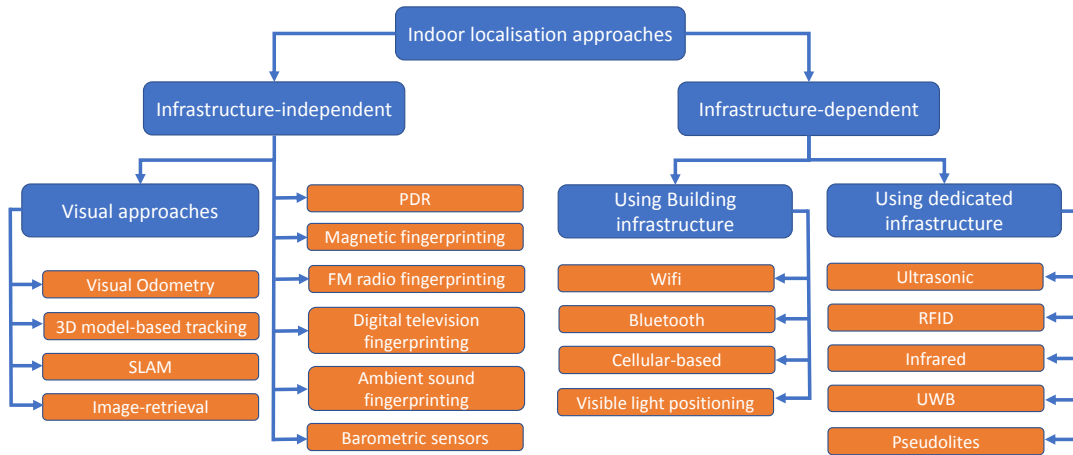


Figure 2.1: The classification of localisation approaches (after [Alarifi et al., 2016](#))

readers or artificial visual markers (such as bar codes). Figure 2.1 shows the broad classification of the indoor localisation approaches.

Infrastructure-dependent localisation approaches

Infrastructure-dependent approaches require dedicated or existing infrastructure for localisation. These approaches include marker-based visual approaches, RFID, Ultra-Wide-band (UWB), Infrared, Ultrasonic, high sensitive global navigation satellite system (GNSS), radar, Pseudolites. These approaches require dedicated infrastructure for operation, where significant cost is involved for installation and maintenance. Approaches such as Wifi, Cellular-based and Blue-tooth use existing infrastructure of the building for operation, however, such approaches do not provide adequate accuracies for many applications ([Mautz, 2012](#)).

Infrastructure-independent localisation approaches

With the widespread availability of smartphones with inertial sensors and cameras, infrastructure-independent approaches such as the pedestrian dead reckoning (PDR) ([Harle, 2013](#)) and visual approaches, such as simultaneous localisation and mapping (SLAM) ([Davison, 2003](#)), visual odometry ([Nister et al., 2004](#)) and model-based visual tracking

(Lepetit and Fua, 2005; Acharya et al., 2019b), have turned out to be more suitable for many applications. Other infrastructure-independent localisation approaches include digital television, FM radio, magnetic fingerprinting, ambient sound fingerprinting and barometric sensors. However, the accuracies of these approaches are not sufficient for many indoor applications (Xie et al., 2014; Muralidharan et al., 2014; Ye et al., 2014; Tarzia et al., 2011; Serant et al., 2011).

Before proceeding into the details of the visual approaches, a brief description is presented regarding the motivation of using building information modelling (BIM) for the current research and the works related to using BIM for localisation in the next section.

2.2 Visual localisation approaches dependent on initial location

Visual approaches that require an initial location at the start of the localisation process are visual odometry, SLAM and 3D model-based tracking. For the scope of the research, the review is restricted to the approaches that use image features for improving localisation.

2.2.1 Visual odometry

Visual odometry (Nister et al., 2004; Scaramuzza and Fraundorfer, 2011) is a local motion estimation approach, where the camera motion is estimated by recursive tracking of visual features in a sequence of images. Consequently, errors are accumulated and the estimated locations drift from the true locations. Visual landmarks (Zhu et al., 2007) have been used to reduce the drift by recalibrating the location at each known landmarks. The robustness of visual odometry improves with the fusion of IMU data; however, the drift is not eliminated completely (Khoshelham and Ramezani, 2017; Ramezani et al., 2018; Ramezani and Khoshelham, 2018).

Image edges have been utilised in visual odometry for increased robustness. Witt and Weltin (2013) propose a stereo visual odometry by matching straight lines between two stereo image pairs by re-projection and considering all lines while performing minimisation, and report a drift of 0.65% for a 25-metre trajectory. Koletschka et al. (2014) combine point features with lines to improve the localisation and report a drift of 4 metres on a

trajectory of several hundred metres. [Jose Tarrío and Pedre \(2015\)](#) propose an edge-based visual odometry approach by using structural information in the form of edge extraction and report similar drifts (0.8-2.8% for a 40 metres trajectory) compared to state-of-the-art feature-based and direct approaches. [Gomez-Ojeda and Gonzalez-Jimenez \(2016\)](#) proposes a stereo approach to visual odometry to reduce the drift by a probabilistic combination of points and line segments by minimising the errors of both points and line segment features and report a drift of 3.26% for a trajectory of several hundred metres.

[Briales and Gonzalez-Jimenez \(2016\)](#) estimate the angular rotation between two frames by matching the orthogonal lines in a random sample consensus framework employing the orthogonal assumptions for the real world. However, the approach cannot be directly applied to localisation as the translations have to be separately estimated by another approach. [Gomez-Ojeda and Gonzalez-Jimenez \(2018\)](#) propose a line segment tracking for high dynamic range stereo image sequences using image edges and applying the geometric consistency along the consecutive frames and report decimetre level accuracy. However, they report worse accuracies as compared to feature-based approaches and they face the problem of a smaller number of edge correspondences. In all these visual odometry approaches localisation errors accumulate and the drift persists.

2.2.2 Visual SLAM

Visual SLAM is an approach in which the localisation is performed with respect to a map of the environment which is created simultaneously by the use of a single camera ([Mur-Artal et al., 2015](#); [Davison, 2003](#)) or multiple cameras ([Scaramuzza and Fraundorfer, 2011](#); [Gomez-Ojeda et al., 2017](#)). A visual SLAM is potentially more precise than visual odometry because it enforces many more constraints on the path ([Silveira et al., 2008](#)), such as a loop closure where a previously visited location has to be revisited to correct the accumulated drift ([Mur-Artal and Tardós, 2014](#)).

Image edges have also been used to create robust SLAM approaches. [Gee and Mayol-Cuevas \(2006\)](#) propose a monocular model-based SLAM algorithm by edge detection, reconstruction of the 3D edges in an Unscented Kalman filter framework and use RANSAC to eliminate wrong matches. But their approach is susceptible to failure under large trans-

lations and rotations, and their experiments were limited to a part of a small room. [Hofer et al. \(2013\)](#) perform a line-based 3D reconstruction of wiry objects by using epipolar constraints of the from the already known camera poses. However, the main limitation of their approach is the prior requirement of camera poses estimated by structure-from-motion approaches.

[Zhang and Koch \(2014\)](#) propose a SLAM algorithm to reconstruct structural lines and detect the motion of the camera from line correspondences that are assumed to be known and perform sparse bundle adjustment. However, the performance of their algorithm was poor for real data that contained more mismatches compared to the synthetic data they use for experiments. The primary limitation of all SLAM algorithms is the loop closure, without which all SLAM algorithm reduces to visual odometry and the drift increases without a loop closure ([Cadena et al., 2016](#)). In navigation applications, loop closure is not always practical, as the user cannot be forced to make loops.

2.2.3 3D model-based visual tracking

3D Model-based visual tracking approaches ([Lepetit and Fua, 2005](#)) refer to the techniques that use a priori knowledge of the environment in the form of a 3D model for matching the features in the camera view with the features in the 3D model. Localisation can subsequently be performed directly by using the depth information from the 3D model, without using the information of the camera motion, and Figure 2.2 shows an example of model-based tracking using a 3D building model.

Image edges have been used for model-based visual tracking by minimising the re-projection errors using an approximate camera pose with a robust cost function. A strong edge may not correspond to an actual correspondence, for example, due to shadows, and wrong correspondences (or outliers) can lead to a wrong estimation of the camera pose. Therefore, the major drawback of matching edges is that they are not distinctive enough to provide information for matching two separate images ([Choi and Christensen, 2012](#)). This drawback leads to failure in the presence of texture or cluttered scenes. Hence, robust estimation is an integral part of visual tracking approaches to counter the influence of spurious data ([Lepetit and Fua, 2005](#)). In the literature, RANSAC ([Zisserman,](#)



Figure 2.2: An example of 3D model-based visual tracking. The left image shows the projection of initial camera pose and the building edges on the image. The right image shows the overlay of the estimated camera pose after a few iterations.

1995; Choi and Christensen, 2012; Lepetit and Fua, 2005), M-estimators (Drummond and Cipolla, 2002; Lepetit and Fua, 2005) and Tukey estimators (Wuest et al., 2005) have also been used for eliminating the wrong correspondences.

David et al. (2003) extends SoftPOSIT algorithm (David et al., 2002) to match 3D CAD model edges with image edges for performing model-to-image registration. However, the authors do not report the accuracies for the experiments with real images, neither they perform a detailed trajectory analysis. Many studies utilise multiple hypotheses (Vacchetti et al., 2004; Wuest et al., 2005) in particle filter frameworks such as condensation (Vacchetti et al., 2004; Lepetit and Fua, 2005), to increase the robustness of tracking. The increase in robustness is compensated by the additional computation requirements (Vacchetti et al., 2004; Lepetit and Fua, 2005) and real-time applications of the same remains a challenge, especially on smartphones.

Kitanov et al. (2007) use lines derived from the camera image and compare them with lines in rendered synthetic images from a 3D vector model to estimate the orientation

and location of the camera and dm-level accuracy in localisation is reported. [Teulire et al. \(2015\)](#) used multiple-hypothesis tracker within a particle filtering framework using edges to perform model-based tracking with an unmanned aerial vehicle (UAV). An accuracy of 16 centimetre was reported for translation. Similarly, [Qiu et al. \(2017\)](#) match images obtained from a camera mounted on a drone and synthetic images derived from a 3D model by minimising the geometric re-projection error using distance transform of the images. Additionally, the authors integrate the IMU measurements of the drone in an extended Kalman filter framework. They reported an accuracy of 7 centimetre. However, all these model-based tracking approaches have been designed for tracking small objects and in small spaces such as a room and are therefore unsuitable for continuous localisation in large indoor environments.

2.3 Visual localisation methods independent of initial location

The visual localisation approaches that are independent of initial location can be classified as either as appearance-based or pose estimation-based approaches ([Piasco et al., 2018](#)). The appearance-based approaches consider localisation as an image retrieval problem, where the current view of the camera is compared with the previously stored database of geo-tagged images or image features ([Arandjelovic and Zisserman, 2012](#); [Radenović et al., 2016](#)). However, these approaches require a large database of images with known pose and are susceptible to viewpoint changes, thereby not suitable for many applications ([Piasco et al., 2018](#)).

Pose estimation-based approaches directly estimate the 6-DOF pose of a camera, and can be further classified into three categories: 1) matching point features with 3D point clouds (e.g. [Irschara et al., 2009](#); [Svarm et al., 2017](#)) 2) pose regression using RGB-D images (e.g. [Shotton et al., 2013](#)) and 3) pose regression using images only (e.g. [Kendall et al., 2015](#)). In the approaches that match point features with 3D point clouds, the camera poses are estimated directly by using classical photogrammetry approaches such as P3P ([Gao et al., 2003](#)), thanks to the geometrically structured database and the availability of 3D information of each point feature. The main challenge is the requirement of a 3D point

cloud, which is usually derived from the structure-from-motion (SfM) approaches.

Pose regression can be done by using RGB-D images that contain the depth information of the scene, by encoding the world location of each pixel of the image to a known environment. The common approach is pair-wise registration that uses corresponding 3D key points in consecutive frames to estimate a relative pose (Santos et al., 2016). Regression forests (Shotton et al., 2013), neural networks (Massiceti et al., 2017) and regression ferns (Glocker et al., 2015) have been for performing pose regression with RGB-D images. These approaches are fast and precise. However, the major limitation is the requirement for RGB-D (or depth) cameras which are not as widespread as ordinary cameras. With the advancement of deep learning in the last few years, a substantial amount of work has been done in the field of pose regression using images only.

2.3.1 Deep learning and feature engineering

The conventional paradigm of image feature engineering includes the extraction predefined features (Yilmaz et al., 2006), such as histogram of oriented gradients (HOG). Such image features are susceptible to the variations of object appearance that are complex and highly non-linear. CNNs overcome this limitation by learning features that optimally describe the image content, hence automating the feature generation. Figure 2.3 shows a deep CNN, where the first few (shallow) layers learn the low-level features such as edge and texture of the image, and the *deep* layers learn relevant high-level features of the image. These high-level features are used for different tasks that range from image classification and activity recognition. Consequently, the CNN-based approaches have achieved state-of-the-art accuracies in image classification (Krizhevsky et al., 2012; Acharya et al., 2018) and object detection (Acharya et al., 2017). Recently, CNNs are finding application in the field of localisation, e.g. in SLAM (Parisotto et al., 2018), relative pose estimation (Melekhov et al., 2017b), ego-motion estimation (Costante et al., 2016), visual odometry (Wang et al., 2017a) and camera pose regression (Kendall et al., 2015).

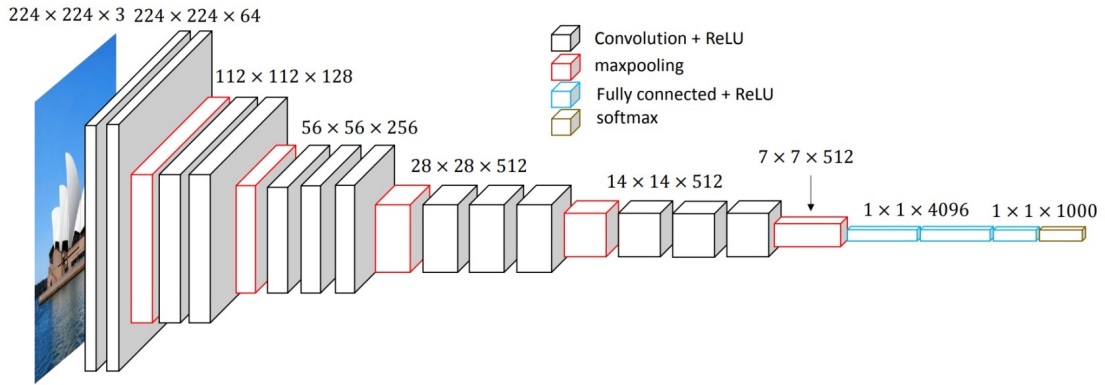


Figure 2.3: A deep convolutional neural network (VGG), after Wang et al. (2017b).

2.3.2 Camera pose estimation using convolutional neural networks

Motivated by the success of the CNNs, Kendall et al. (2015) propose the use of a CNN regressor (*PoseNet*) for pose regression using images only. They fine-tuned¹ a pre-trained network on image samples with ground-truth poses derived from the SfM approaches. They state that CNNs trained for the task of classification preserve pose information until the final layer by leveraging transfer learning, despite being trained for a different task with a different dataset. The authors show that the generated localisation feature vector can be mapped to pose, and the network can generalise unseen scenes with a few fine-tuning examples. Such an approach does not require depth maps of the scene nor a database to store and retrieve images. However, the obtained pose is not as accurate as the metric approaches such as SLAM or visual odometry, but is robust to changes in scale and appearance.

Subsequently, many approaches have been proposed that improve the accuracy of *PoseNet*. For example, Kendall and Cipolla (2016) integrate uncertainty in pose estimation, Kendall et al. (2017) explore new loss functions for fine-tuning the network. Walch et al. (2017) and Clark et al. (2017) use long short-term memory (LSTM) units to exploit the correlation between successive images. Additionally, Wu et al. (2017) use data augmentation for increasing fine-tuning samples, Melekhov et al. (2017a) and Müller et al.

¹Fine tuning is updating the weights of (or a part of) a pre-trained network using a smaller set of training images.

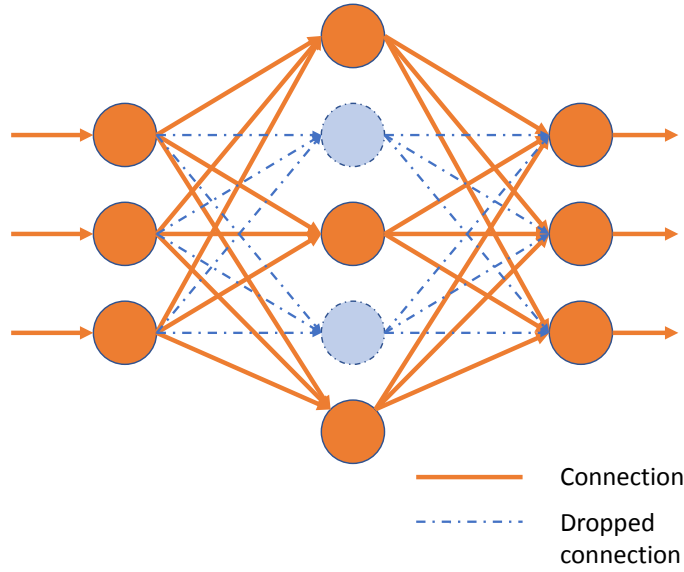


Figure 2.4: An example of dropped out connections between neurons. The orange lines show the connections and the blue dotted lines show the dropped out connections.

(2017) use different network architectures. Moreover, [Brahmbhatt et al. \(2018\)](#) use constraints derived from visual odometry, global positioning system (GPS) and inertial measurements, [Valada et al. \(2018a\)](#) embed geometric information derived from visual odometry and [Valada et al. \(2018b\)](#) combine geometric and semantic knowledge of the world. However, the limitation of all these approaches is the requirement of SfM approaches to estimate the ground truth camera poses that is required during fine-tuning the networks ([Gu et al., 2019](#)).

2.3.3 Modelling uncertainty of camera pose using Bayesian convolutional neural networks

The uncertainty of a system plays a vital role in understanding the confidence of the estimations. In the literature, Bayesian systems have been used to model the uncertainty of a system, such as Kalman filtering ([Groves, 2013](#)). On the other hand, neural networks which consider uncertainty are known as Bayesian neural networks ([MacKay, 1992](#)). These networks interpret the probabilities from the network weights, by inferring distributions approximately, such as variation inference ([Graves, 2011](#)).

Dropout is a regularisation technique in CNNs to prevent over-fitting to training or fine-tuning data and helps the network learn features that perform well on unseen data (Srivastava et al., 2014). Figure 2.4 shows an example of the dropout, where the blue dotted lines show the removed connections, and the solid orange lines show the connections between two neurons. Dropout involves randomly removing connections within a network during the training phase. Dropout has been used to approximate Bayesian inference over the network's weights (Gal and Ghahramani, 2016a) and has been used at test time to impose a Bernoulli distribution over CNN weights (Gal and Ghahramani, 2015). This approximation is achieved by sampling the network with dropout at the test time and the outputs can be considered as Monte Carlo samples from the posterior distribution of models.

Kendall and Cipolla (2016) model the uncertainties of camera pose estimates by using a Bayesian neural network. The authors state that the estimated uncertainties correlate with the actual errors obtained from ground truth camera poses for real images. They also showed that the location and the rotation uncertainties are correlated, and thus can be used to form a single uncertainty value which represents the overall model uncertainty.

2.3.4 Recurrent neural networks and the use of spatio-temporal information

Recurrent neural networks (Goller and Kuchler, 1996) are a class of machine learning algorithms that use previous outputs of the network as recurrent inputs. The advantage of using such a model is to exploit the historical information of a data sequence which finds applications in domains such as natural language processing and image recognition. However, such a model suffers from the problem of vanishing gradient and accessing information of a longer sequence becomes a challenge. Gates are often used in recurrent neural networks to tackle the problem of vanishing gradient.

LSTM (Hochreiter and Schmidhuber, 1997) is a specific type recurrent network that uses gates to accumulate or forget relevant contextual information in its hidden state. Deep architectures combining CNN and LSTM have been successfully applied in the field of visual recognition from videos, by encoding the contextual information, and by applying LSTMs on temporal sequences.

Recently, LSTMs have been applied in the field of camera pose regression, to improve the pose estimations, hence reducing the errors. [Walch et al. \(2017\)](#) propose the use of a CNN-LSTM architecture to perform a structural dimensionality reduction on the image feature vector derived from the CNN. Such a feature reduction enabled the model to better correlate the image features and the camera pose estimates and improved the pose estimates as compared to using dropout on the feature vector. The authors reported an improvement in the localisation accuracy of approximately 30% with respect to the previous deep learning models.

The CNN-based pose regression approaches discussed so far in the chapter, estimate the camera pose of images independently, and do not exploit the valuable constraint of temporal smoothness of the captured images. This practise leads to cases where the inter-frame distances are larger than the motion of the camera. Moreover, such approaches are susceptible to the problem of perceptual aliasing, where two different places appear similar. To tackle the challenge of perceptual aliasing, [Clark et al. \(2017\)](#) propose a recurrent network architecture consisting of LSTM units and window of image sequences. The authors show that the estimated camera poses are smoothed and localisation errors reduce compared to existing approaches, even by considering short sequences (twenty frames). Furthermore, the authors modelled the uncertainties from the variance of camera pose estimates. However, all of these pose regression approaches are dependent on a 3D reconstruction of the indoor environment using SfM approaches for operation. In the next section, the approach of using synthetic images to eliminate the requirement of SfM approaches is presented.

2.3.5 Use of synthetic images in deep learning

With the advancement of computer graphics and graphics processing units, the realism of the synthetically generated images has greatly improved. [Figure 2.5](#) shows an example photo-realistic synthetic image generated using synthetic textures. In the literature, synthetic images generated from 3D object models have been used to eliminate the challenge of creating manually labelled images (ground truth images), which is required for training or fine-tuning CNNs. [Su et al. \(2015\)](#) generate millions of synthetic image from

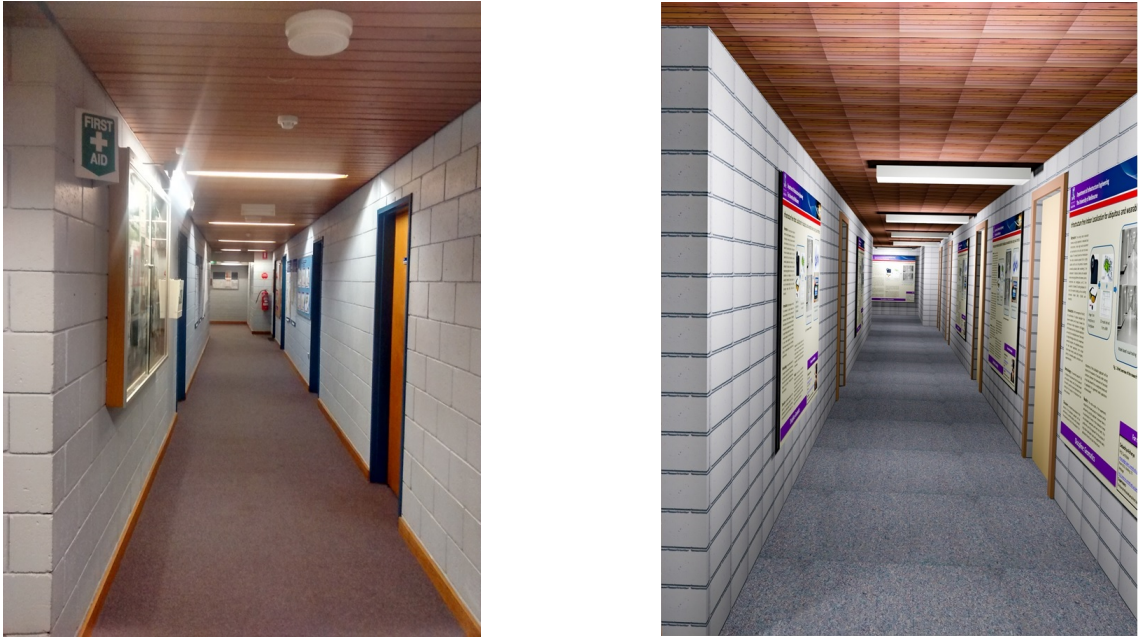


Figure 2.5: An example of synthetic images with textured generated from the same view-point inside a building, using a BIM. The left image shows the real image, and the right image shows the synthetic image.

3D object models to classify object orientation in the image (also known as viewpoint estimation) using CNN features. Similarly, [Varol et al. \(2017\)](#) generate millions of synthetic images from a 3D human model for estimating depth and body part segmentation in real images by training CNNs with synthetic images. [Peng et al. \(2015\)](#) train linear classifiers using a few (2000) synthetic images generated from 3D object models for object detection in real images by using features from CNNs. Motivated by the use of synthetic images generated from 3D object models, [Ha et al. \(2018\)](#) present an image-based retrieval approach to perform a coarse indoor localisation using a BIM. Synthetic images derived from the BIM are compared with real images on the basis of CNN features.

2.4 Building information modelling for visual localisation

A BIM is a digital representation of all the physical and functional characteristics of a building through its entire life cycle. Recently, BIMs are used to acquire valuable sources of information for enabling indoor navigation as they deliver additional information compared to 2D maps. For example, a BIM can provide semantic information, or the

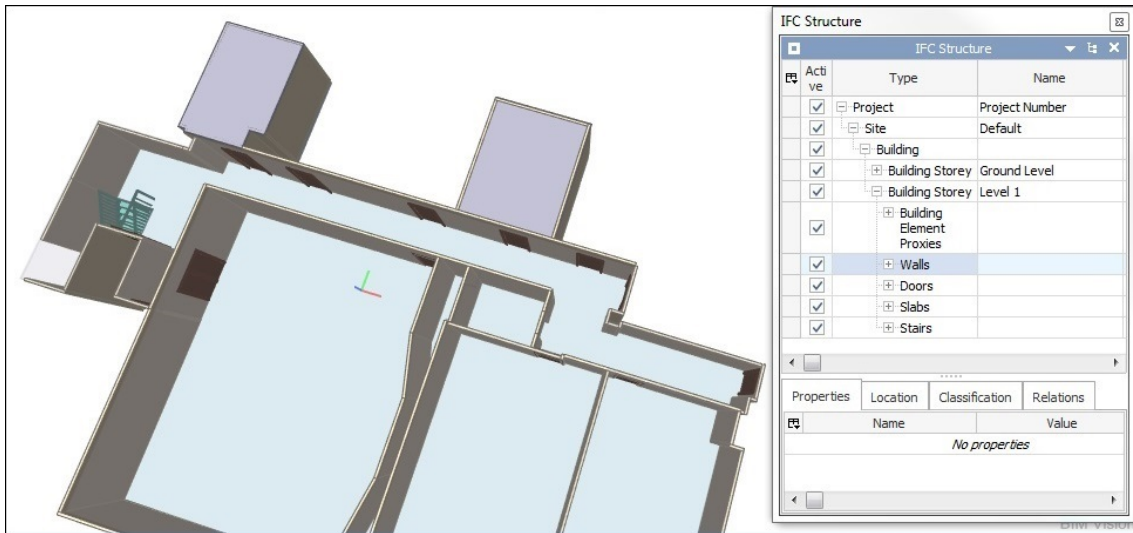


Figure 2.6: A BIM of a building containing the semantic properties (visualised in BIM Vision).

properties and functional states of the building elements, spatial relationships between them and geometric information (Isikdag et al., 2013). Moreover, BIMs have the potential for large-scale coverage of an indoor environment without a significant increase in costs. Figure 2.6 shows a BIM that contains the semantic information of building elements such as walls, doors, slabs and stairs.

Despite the easy availability of BIMs for most of the modern buildings, the visual localisation approaches that use BIMs are in their infancy and very few works have been reported in the literature. It is possible to identify the building structural elements that are visible from a viewpoint inside a building, by placing a virtual camera inside the corresponding BIM. Besides, synthetic images for any viewpoint of the building can be generated as well. Chen et al. (2011) used a BIM to create synthetic images derived from virtual cameras and compared with real images on the basis of image features. Subsequently, the virtual image is mapped to the real image and a surveillance camera is registered in the BIM coordinate system. Kropp et al. (2012) propose a conceptual framework using an inertial measuring unit (IMU), camera and BIM for predicting delay of construction works. Initial registration in the BIM has to be done by the user. Subsequently, IMU and camera will be utilised for localisation by tracking of visual features with a sequence of video frame taken beforehand. Following that, missing items will

be detected and displayed on a hand-held device. Recently, [Ha et al. \(2018\)](#) propose to perform coarse localisation using synthetic images obtained from a BIM and comparing them with real images on the basis of convolutional neural network (CNN) features. This method is explained in further details in Section [2.3.1](#).

2.5 Summary

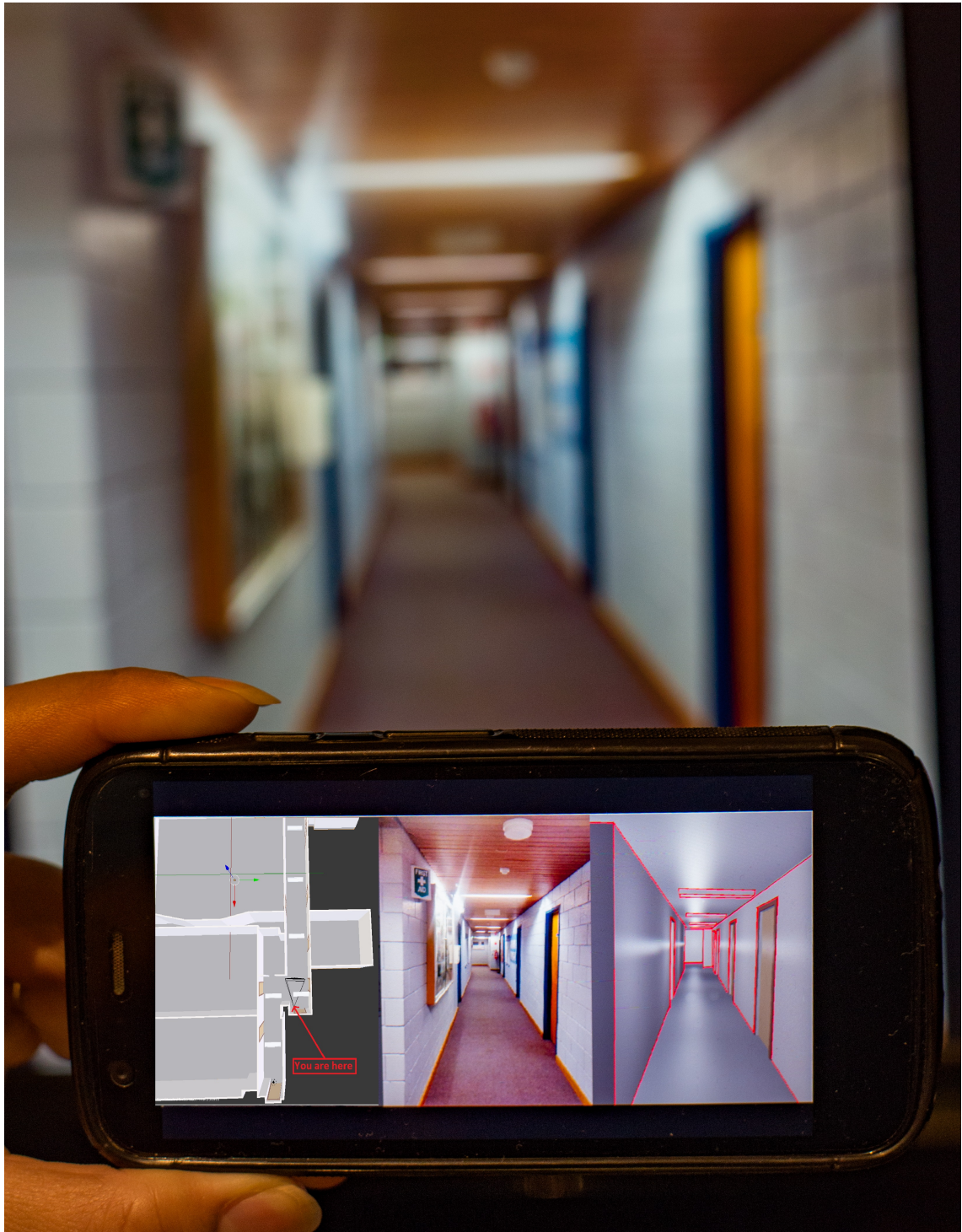
In summary, the review of the literature indicates the absence of a single technology that can be used for indoor localisation, due to the cost involved. Therefore, approaches that do not require any additional infrastructure, and can be used on ubiquitous devices have become popular. The presence of cameras on ubiquitous devices have resulted in a renewed interest in visual localisation approaches, especially with the recent advancements in deep learning approaches.

The limitations of the visual approaches, such as SLAM, visual odometry and model-based visual tracking restricts their wide applicability. SLAM suffers from the requirement of a loop closure that is not always practical. Visual odometry suffers from the problem of error accumulation, and often use other sources for calibrating the error. The use of model-based visual tracking approaches for continuous localisation in large indoor environments and the achievable accuracy is an unexplored research area. Moreover, such approaches are dependent on an initial location.

Visual approaches that are independent of an initial location are image-based retrieval approaches. These approaches involve performing a 3D image-based reconstruction of the indoor environment, therefore limiting their applicability on large indoor spaces. Additionally, some of these approaches are dependent on additional sensors, such as depth sensors, the presence of which is rare for most of the ubiquitous devices. Therefore, the limitations of the existing approaches need to be addressed in order to achieve an infrastructure-independent visual localisation approach.

The wide availability of BIMs for most of the modern buildings provides an opportunity to address the challenges of the existing visual approaches, and to perform infrastructure-independent localisation. There exist a very few works in the literature that pro-

pose the use of BIMs either for reducing the drift of localisation or for eliminating the 3D image-based reconstruction of the environment. The thesis highlights the contributions of the current research by using a BIM, and points towards an infrastructure-independent visual localisation.



"... Visual localisation is estimating the camera's location ..."

≈ Block B, Department of Infrastructure Engineering, The University of Melbourne, Australia
(Courtesy: Moto G)

Chapter 3

BIM-Tracker: A model-based visual tracking approach using a 3D building model

This chapter presents an accurate and robust visual indoor localisation approach that is infrastructure-independent and avoids the accumulation of error. The motivation of using the approach is the ubiquity of mobile devices with cameras and the availability of 3D building models for most modern buildings. Localisation is performed by matching image sequences captured by a camera, with a 3D building model in a model-based visual tracking framework. A comprehensive evaluation of the approach with a photo-realistic synthetic dataset shows the robustness of the localisation approach under challenging conditions. Additionally, the approach is tested and evaluated on real data captured by a smartphone. The results of the experiments indicate that a localisation accuracy better than 10 centimetres can be achieved by using this approach. Since localisation errors do not accumulate the proposed approach is suitable for indoor localisation tasks for long periods of time and augmented reality applications, without requiring any local infrastructure.

3.1 Introduction

The discussion of the relevant literature in Section 2.2 suggests that infrastructure-independent visual localisation approaches such as visual odometry and simultaneous localisation and mapping (SLAM), suffer from the accumulation of localisation errors resulting in the drift of the estimated trajectory (Khoshelham and Ramezani, 2017; Scaramuzza and Fraundorfer, 2011; Caron et al., 2014). Moreover, visual odometry and SLAM approaches are susceptible to poorly textured indoor environments such as a corridor, due to the lack

of image features. Additionally, SLAM involves performing a loop closure, which is not practical for navigation applications, e.g. in a long tunnel, as the user cannot be forced to make loops. A drift of 1% percent in the trajectory is acceptable for short distances but is not suitable for long-distance indoor localisation applications.

Therefore, in this chapter BIM-Tracker is introduced to eliminate the accumulation of error, and to work in poorly textured environments. BIM-Tracker is a 3D model-based visual tracking approach to indoor localisation which is based on matching images captured by a mobile device with the corresponding view of a 3D building model. The edges in the images are matched with the edges derived from the BIM to estimate the location of the camera in the BIM coordinate system. The advantage of performing localisation in a BIM coordinate system is that the estimated locations are not prone to drifts compared to the incremental tracking approaches that perform localisation by local motion estimation ([Khoshelham and Ramezani, 2017](#)).

The existing model-based visual tracking approaches, such as the work of [Drummond and Cipolla \(2002\)](#) eliminate the requirement of textured indoor environments, and are computationally inexpensive ([Lepetit and Fua, 2005](#)). However, these approaches have been designed for tracking small objects and in small spaces such as a room and are therefore unsuitable for continuous localisation in large indoor environments. Moreover, the existing works that use model-based visual tracking lack a comprehensive evaluation of the achievable accuracy, robustness and the estimated trajectory.

Therefore, in BIM-Tracker a detailed analysis of the trajectory with synthetic images and their respective ground truth poses are performed. The distribution of the errors along the trajectory are analysed and evaluate the factors that result in significant errors. Besides, the effects of the image resolution, camera field-of-view (FoV), clutter occlusion and motion blur on the localisation performance are assessed. To the best of our knowledge, such a detailed analysis of the trajectory and the factors affecting the errors have not been reported in the literature, for model-based visual tracking.

Because most of the smartphones and smartglasses are equipped with a camera, and the fact that a low level-of-detail 3D model of the environment is usually available or can be easily generated, the present research proposes BIM-Tracker for accurate and drift-free

localisation in an infrastructure-independent indoor environment. The following are the main contributions of the work:

1. An MSAC (M-estimator sample consensus) framework is formulated to use two hypotheses on either side of a back-projected model line to search the corresponding image edges. This strategy is a balance between the higher computational costs of using several multiple hypotheses for tracking and the robustness that it provides, and is discussed in Section 3.2.
2. Experimental insight are provided into the optimal camera configurations and factors that contribute to errors for a model-based visual tracking approach in an indoor environment. A detailed analysis of the estimated trajectory is performed using a photo-realistic synthetic dataset with several configurations such as different image resolutions, camera field-of-views (FoV), motion blur, clutter and occlusions. The results are presented in Section 3.3.
3. The ability of BIM-Tracker for drift-free localisation using real images is demonstrated, and compare with a state-of-the-art SLAM approach. The results are presented in Section 3.3.3 and demonstrates the ability of BIM-Tracker for navigation and augmented reality applications.

3.2 Methodology

The main hypothesis of BIM-Tracker is that centimetre level accuracy in localisation can be achieved without any drift by integrating image information with a 3D building model. To test the hypothesis, a series of experiments with photo-realistic synthetic datasets by changing the conditions for each set of data were conducted. Further experimentation is performed with real data from a smartphone to evaluate the quality of the estimated camera pose for augmented reality applications.

Figure 3.1 shows the workflow of BIM-Tracker, where localisation starts from a rough initial camera pose (initialisation step), to render the visible edges on-line from the 3D building model by ray tracing. For the scope of this research, it is assumed that the

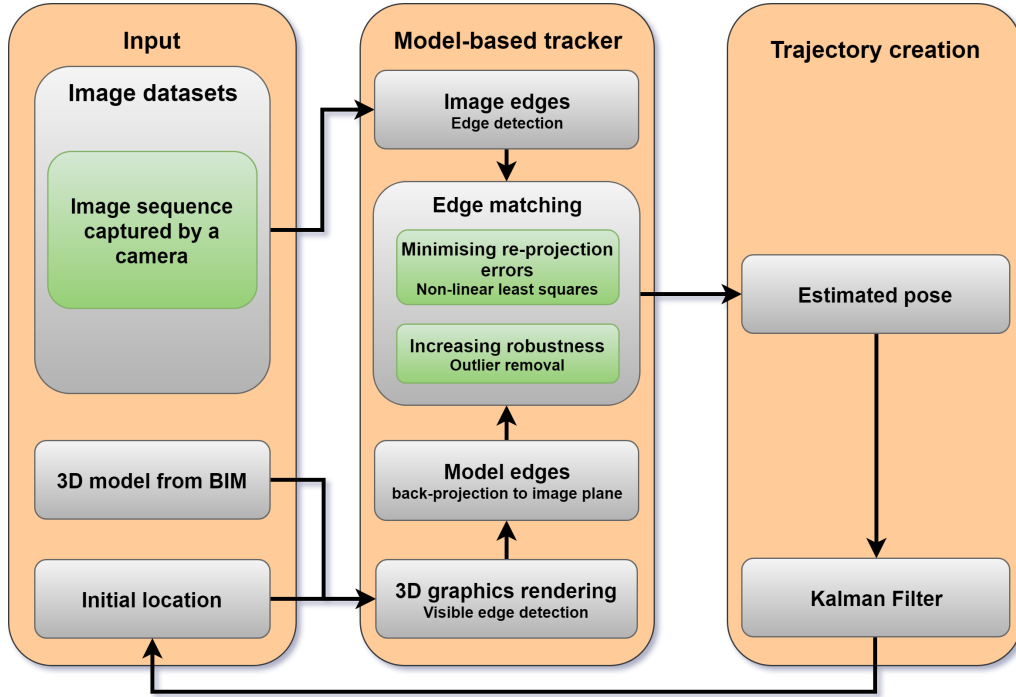


Figure 3.1: The workflow of BIM-Tracker.

initial camera pose at the start localisation is known beforehand. This step is followed by sampling the visible edges according to their length, to split edges into 3D points. Subsequently, the sampled points are back-projected on the image plane. Consequently, correspondences are created by searching in the proximity of back-projected sampled points with the structural edges in the image. To remove the wrong 3D to 2D correspondences, that can likely be a result of shadows or repetitive texture, MSAC estimator (Torr and Zisserman, 2000) is used. The camera pose is estimated by minimising the distance between the image points and back-projected 3D points iteratively using Gauss-Newton non-linear minimisation techniques (Triggs et al., 2000). The estimated camera pose is fed to a Kalman filter, which is used to predict the camera pose for the next frame.

3.2.1 Pose initialisation

For the current research, the pose is initialised manually at the start of the sequence, by roughly selecting three points on the image and the corresponding 3D points in the model. The pose is subsequently estimated by using Gauss-Newton minimisation (Sec-

tion 3.2.7). The initialisation need not be accurate as BIM-Tracker can handle bad initial poses in the region of convergence (Section 3.3.5). In practical applications, the initial pose can be estimated by using direct pose regression approaches (Kendall et al., 2015; Piasco et al., 2018), which can instantaneously recover the pose of a single image in a known coordinate system either by utilising textured 3D models, RGBD (RGB + depth) images or sequence of images. The reported errors are under 0.6 metres and 8° , which is sufficient for initialising BIM-Tracker.

3.2.2 Determination of visible edges of the 3D building model in camera views

Assuming an initial estimate of the camera pose is available from the previous step, a virtual view is rendered and visible edges as well as their corresponding 3D coordinates in the BIM coordinate system, are determined by using a ray tracing algorithm of Blender¹. The advantage of such an approach is that it can handle many complex situations as compared to precomputed tabular approaches (Drummond and Cipolla, 2002). Bounding volume hierarchies (BVH) trees are used (Wald et al., 2007) to reduce the computational cost of ray tracing.

An edge of an object in a 3D building model is described as a set of two 3D vertices and hence for an edge which is visible in a scene requires both the vertices to be detected. But, under certain camera views, both the vertices of the edge are unlikely to be visible, though most of the edge is plausible to be visible in the camera view. To solve the problem, the edges are subdivided during the creation of the 3D building model to facilitate the proper detection of model edges in the camera view and therefore increase the robustness. The model edges are subdivided according to the length off-line, and during many experiments it was found that dividing model edges at an interval of 50 centimetres is sufficient to address the problem. However, this increases the overall complexity of the presented approach to some extent.

The problem is partially solved by eliminating the visible model edges that are located in close proximity of each other during back-projection onto the image plane. This

¹Blender is an open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Available from www.blender.org

action solves another problem, that is when independent model edges are very close to one another under certain camera views when back-projected on the image plane (Reitmayr and Drummond, 2006; Vacchetti et al., 2004). This scenario usually happens to the edges that are distant from the camera view and the back-projected model edges on the image can be combined to produce a single edge. As this can result in incorrect 3D to 2D correspondences (discussed in Section 2.2.3), the back-projected model edges that are very close (less than a pixel) in the camera view are discarded. In addition, for the same reason, the back-projected model edges that were less than a few pixels (d_{min}) are removed.

3.2.3 Line representation

In the literature, a 2D line is usually represented by its two end points (Gomez-Ojeda and Gonzalez-Jimenez, 2018, 2016; Witt and Weltin, 2013; Zhang and Koch, 2014). However, such a representation might lead to incorrect matches or no matches while searching for a corresponding image edge. For instance, consider an example in Figure 3.2 where a line represented by its two endpoints ends up creating wrong edge correspondences. This problem can be particularly challenging when the number of visible model edges is small, and that can result in complete tracking failure.

A solution to overcome this challenge is to represent a line with multiple segmented points (Vacchetti et al., 2004; Wuest et al., 2005; Drummond and Cipolla, 2002; Choi and Christensen, 2010). As shown in Figure 3.2, this representation increases the number of search lines and provides more correct matches. However, such a representation increases the computational cost but provides better solutions for the frames having fewer number of model edges. Therefore, to add robustness against the small number of lines, this representation of the line is selected.

3.2.4 Point sampling of the model edges

After the determination of the visible edges of the 3D building model in the camera view, the endpoints of the model edges are projected on the image plane, and the length of

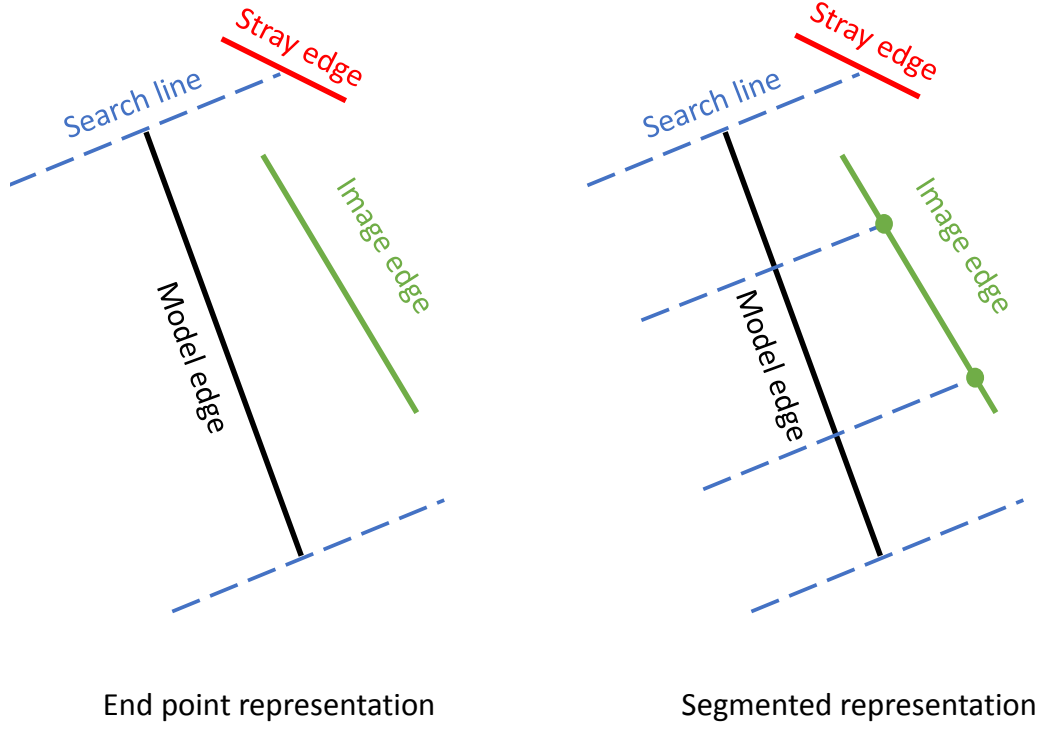


Figure 3.2: The advantage of segmenting edges as compared to using end points to represent lines. Left: A line is represented by end points and the search for corresponding edges misses out the actual matches. Right: A line is segmented into points that allows additional searches to create correct matches.

each model edge is calculated during an initial back projection. The length is necessary for calculating the number of points, the model edges will be sampled. The number of times each model edge is sampled is proportional to the length of the back-projected model edge, and thereby, longer model edges are sampled more than the shorter ones. Ideally, the point sampling should be done on the image plane and not in the 3D space to uniformly distribute the sampled points along the model edges. But performing point sampling on the image plane and then finding its corresponding 3D points is computationally expensive considering the number of points in each iteration. Consequently, sampling the 3D model edges are performed in 3D space according to Eq. 3.1 and the sampled 3D points are back-projected into the image plane (Figure 3.3).

$$n_l = \frac{l_p}{l_s} \quad (3.1)$$

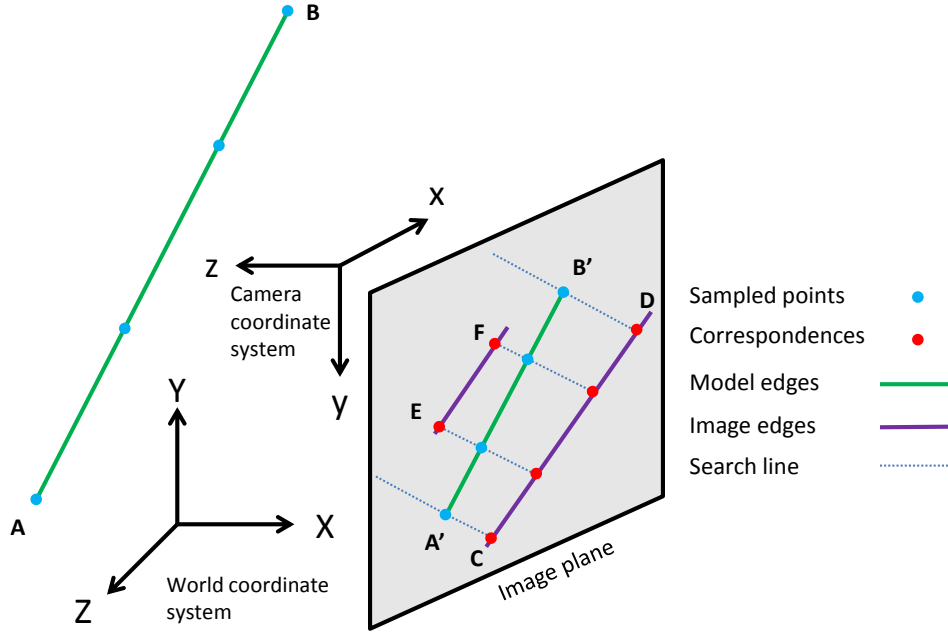


Figure 3.3: Point sampling of the model edges and creating correspondences after back-projection. $A'B'$ is the back-projection of the 3D model edge to the image plane. CD is the corresponding edge to $A'B'$, and EF is a stray edge. Blue dotted lines are the search lines from the sampled points (blue circles). There are two potential correspondences for the points on $A'B'$ on either side of the line. Taking a single hypothesis would have created two wrong correspondences to the edge EF , as the edge EF is closer to $A'B'$ compared to edge CD .

where n_l is the number of times l^{th} model edge was sampled, l_p is the length of the back-projected edge on image plane and l_s is a parameter for setting the sampling distance within the projected edge, and this is similar to the metric used in (Wuest et al., 2005) for sampling 3D model edges.

3.2.5 Projecting model edges to the image

A perspective camera model is considered for the projection of the 3D world points onto the image plane of the camera, which is based on the collinearity equations. Mathematically this can be expressed as:

$$\hat{P}_{oi} = \lambda R \hat{P}_{ow} \quad (3.2)$$

where $\hat{P}_{ow}$ is the vector representing the distance from the optical centre of the camera to a world point and is expressed in world coordinate system, $\hat{P}_{oi}$ is the corresponding vector from the optical centre to the projection of the world point on the image plane, λ is a scale factor, and R is the rotation matrix which defines the rotation of the camera with respect to the world coordinate system. Terms $\hat{P}_{oi}$, $\hat{P}_{ow}$ and R can be expanded and the Eq. 3.2 can be rewritten as:

$$\begin{bmatrix} x - x_p + \Delta x \\ y - y_p + \Delta y \\ -f \end{bmatrix} = \lambda R_z(\kappa) R_y(\phi) R_x(\omega) \begin{bmatrix} X_w - X_0 \\ Y_w - Y_0 \\ Z_w - Z_0 \end{bmatrix} \quad (3.3)$$

where x and y are the projection of a point world point (X_w, Y_w, Z_w) in image plane, x_p and y_p are the image coordinates of the principal point of the camera (X_0, Y_0, Z_0) in the image plane, f denotes the focal length of the camera, ω, ϕ , and κ are the rotation about X, Y and Z axes respectively. Δx and Δy are the terms that model the effects of symmetrical and asymmetrical lens distortion and mathematically they can be explained by:

$$\begin{aligned} \Delta x &= \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2\bar{x}^2) + 2p_2\bar{x}\bar{y} \\ \Delta y &= \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1\bar{x}\bar{y} + p_2(r^2 + 2\bar{x}^2) \\ &\quad + a_1\bar{x} + a_2\bar{y} \end{aligned} \quad (3.4)$$

where $\bar{x} = x - x_p$, $\bar{y} = y - y_p$, $r^2 = \bar{x}^2 + \bar{y}^2$, and k_1, k_2 and k_3 represent lens radial distortion, p_1 and p_2 represent lens decentering distortion parameters and a_1 and a_2 represents affinity distortion parameters. Hence, using the collinearity equations, a real world point (X_w, Y_w, Z_w) can be projected in the image plane as (x, y) .

It is assumed that the camera used in the current research is calibrated, that is the intrinsic parameters of the camera are known before the task of the localisation. This assumption is practical because a camera can be calibrated off-line easily and the internal parameters of the camera do not change with time. Additionally, this increases the performance of the presented approach as the intrinsic camera parameters are not estimated during the pose estimation process.

3.2.6 Creating 3D to 2D point correspondences using edges

A Canny (Canny, 1986) edge detector is used to detect the edges present in the image. Subsequently, the 3D points are sampled and back-projection onto the image plane. Correspondences are generated by performing a search for edges on the image in the perpendicular direction of the back-projected model edges from the sampled points (Figure 3.3) similar to the works of Drummond and Cipolla (2002); Wuest et al. (2005); Comport et al. (2003); Marchand et al. (2001). A search distance of 25-40 pixels is empirically found sufficient for tracking purposes considering the frame rate is close to 30 frames per second (FPS) for an image resolution of 640x480 pixels. For lower frame rates and higher resolution of images, this distance should be higher.

As explained in Section 2.2.3, multiple hypotheses are kept to increase the robustness of the tracking, and this comes at the cost of additional memory and computation requirement. In the current research, only two hypotheses on either side of the model edges are kept rather than all hypotheses in the search distance. There are three main reasons for doing this 1) if a stray image edge exists in the proximity of the back-projected model edge, wrong correspondences might be created (Figure 3.3) 2) memory and computation load decreases rather than keeping all hypothesis and 3) considering multiple hypotheses increases the outlier ratio, which is the number of outliers (wrong correspondences) divided by the total number of correspondences, and this could be a challenge for doing the robust estimation by MSAC (explained in Section 3.2.8). Thus, a set of $2n_l$ correspondences are generated for each model edge and considering the number of model edges to be l , the total number of correspondences created is $n = 2n_l l$.

3.2.7 Pose estimation by non-linear least square minimisation

Given adequate initial values of the unknown parameters, camera pose is estimated by the Gauss-Newton algorithm of least-squares minimisation, which iteratively minimise the re-projection errors (Section 3.2.7) within an MSAC framework (Section 3.2.8). The non-iterative solution to the P3P problem (Gao et al., 2003) is also explored, where the solution of the estimated poses comes from solving a bi-quadratic equation and validating

the correct solution with an additional 3D point. Although the direct solution is faster, the iterative method provides more accurate estimates. Therefore the iterative Gauss-Newton method is used.

Gauss-Newton method of least-squares minimisation

The principal assumption of Gauss-Newton method is that the minimising function is locally linear and the function can be linearised by using Taylor's theorem. The vector of re-projection errors is define as e , which is the length between the image edge and the back-projected sample point in Eq. 3.5.

$$e = \begin{bmatrix} u_1 - x_1 \\ v_1 - y_1 \\ .. \\ u_m - x_m \\ v_m - y_m \end{bmatrix} \quad (3.5)$$

where (u, v) denotes the location of the corresponding edge in the image plane, and m represents the number of measurements. The next step involves minimising the 2-norm of the residual (Eq. 3.6), by linearising the non-linear model explained in Eq. 3.3 using first order Taylor's expansion around the previous camera pose.

$$||J\Delta - e||^2 \quad (3.6)$$

where the vector $\Delta = [dX_0 \ dY_0 \ dZ_0 \ d\omega \ d\phi \ d\kappa]^T$ is solved at each iteration in such a way that the error measurements e between the model edges and the image edges is minimised, and J is the Jacobian matrix containing $2mx6$ partial derivatives of the measurement with respect to the unknown parameters.

$$J = \begin{bmatrix} \frac{\partial x_1}{\partial X_0} & \frac{\partial x_1}{\partial Y_0} & \frac{\partial x_1}{\partial Z_0} & \frac{\partial x_1}{\partial \omega} & \frac{\partial x_1}{\partial \phi} & \frac{\partial x_1}{\partial \kappa} \\ \frac{\partial y_1}{\partial X_0} & \frac{\partial y_1}{\partial Y_0} & \frac{\partial y_1}{\partial Z_0} & \frac{\partial y_1}{\partial \omega} & \frac{\partial y_1}{\partial \phi} & \frac{\partial y_1}{\partial \kappa} \\ .. & & & & & \\ \frac{\partial x_m}{\partial X_0} & \frac{\partial x_m}{\partial Y_0} & \frac{\partial x_m}{\partial Z_0} & \frac{\partial x_m}{\partial \omega} & \frac{\partial x_m}{\partial \phi} & \frac{\partial x_m}{\partial \kappa} \\ \frac{\partial y_m}{\partial X_0} & \frac{\partial y_m}{\partial Y_0} & \frac{\partial y_m}{\partial Z_0} & \frac{\partial y_m}{\partial \omega} & \frac{\partial y_m}{\partial \phi} & \frac{\partial y_m}{\partial \kappa} \end{bmatrix} \quad (3.7)$$

As x and y contain the elements of the rotation matrix $R = R_z(\kappa)R_y(\phi)R_x(\omega)$, chain rule is used for partial differentiation to calculate the elements of the Jacobian matrix. Subsequently, Δ is solved using the following Eq. :

$$J\Delta = e \quad (3.8)$$

The above equation can be further simplified as $J^T J\Delta = J^T e$ and hence can be rewritten as:

$$\Delta = (J^T J)^{-1} J^T e \quad (3.9)$$

Consequently, the vector Δ is used to estimate the vector of unknown parameter $p = [X_o \ Y_o \ Z_o \ \omega \ \phi \ \kappa]^T$ to be subtracted from the current estimate on each iteration. The above process can be mathematically expressed as:

$$p^{(s+1)} = p^{(s)} - \Delta^{(s)} \quad (3.10)$$

where $p^{(s)}$ is the vector of parameters for iteration s and $p^{(s+1)}$ is vector of estimated parameters at iteration $s + 1$. Hence, at each iteration, considering local linearity, the residuals are minimised, and the back-projected sampled points and the image edge points get closer after subtracting the corrections. A threshold for Δ is used, that facilitated fast convergence, through impacting the accuracy of the estimates of the unknown parameters p and the value for Δ is determined by the units of the measurements of the parameters. Consequently, a threshold of 10^{-5} is used for Δ where the unit of measurement is in millimetres and radians for distance (X_o, Y_o, Z_o) and angles (ω, ϕ, κ) respectively.

3.2.8 Outlier removal with MSAC

In the current research, an M-estimator (MSAC ([Torr and Zisserman, 2000](#))) is used, which is the modified version of the RANSAC algorithm ([Fischler and Bolles, 1981](#)). RANSAC algorithm is a hypothesis-and-test-procedure that has been developed in the context of camera pose estimation in the presence of spurious data.

The vector p in Eq. 3.10 consists of six unknowns and the matrix $J^T J$ in Eq. 3.9 is invertible only if $j \geq i$, which means the number of measurements is either greater

than or equal to the number of unknown parameters. Hence, to satisfy this condition, the minimum number of 3D to 2D correspondences required is three, as each 3D to 2D correspondence provides two measurements (Lepetit and Fua, 2005). From the set of all 3D to 2D correspondences (n), a set of three correspondences is extracted randomly and subsequently the unknown parameters of Eq. 3.10 are estimated.

Subsequently, using the estimated parameters by the set, all the 3D points are back-projected onto the image plane. A 3D to 2D correspondence is treated as an inlier if the residual is below an error tolerance threshold (T). The error tolerance threshold is set a few times the standard deviations of the re-projection error of all the correspondences, and hence this eliminates any fixed manual threshold that is dependent on the resolution of the image. Torr and Zisserman (2000) state that an error tolerance threshold of 1.96 times the standard deviation only rejects the Gaussian inliers 5% of the time. A robust cost function C (Eq. 3.11) is used to minimise the sum of the residuals of all the n correspondences.

$$C = \sum_{i=1}^n \rho(\delta_i^2) \quad (3.11)$$

where, δ_i is the re-projection error of i^{th} correspondence, and the error term ρ is used to compute the distance metric used for the process is defined by

$$\rho(\delta^2) = \begin{cases} \delta^2, & \delta^2 < T^2 \\ T^2, & \delta^2 \geq T^2 \end{cases} \quad (3.12)$$

The number of trials N required to ensure with a probability P that at least one of the sample is free from gross error (Fischler and Bolles, 1981) is defined by the following:

$$N \geq \frac{\log(1 - P)}{\log(1 - w^k)} \quad (3.13)$$

where w is the inlier ratio that is the number of correct correspondences (m) divided by the total number correspondences (n), and k is the minimum number of points to perform the estimation. Subsequently, m correct correspondences (inliers) are determined from the set having the minimum value of C and the final camera pose is refined by Gauss-Newton minimisation.

Because two hypotheses for each back-projected sample points are used, one of the hypotheses will always be wrong, and this brings the inlier ratio at the best case to 0.5, however, for the worst case, there could be more outliers. Hence, the inlier ratio is set to be half of the best case, which brings down the inlier ratio to 0.25. From Eq. 3.13 it can be derived that the number of trials required by the MSAC algorithm to ensure that at least one of the sample free from outliers with a probability of 0.95 is 191, considering an inlier ratio to be 0.25.

Jump-out rules

To improve the convergence rate and to reduce the computational load of the MSAC process, the following set of rules are applied:

1. Rule 1: Gauss-Newton method of least-squares minimisation (see Section 3.2.7) requires that the 3D to 2D correspondences are non-collinear. Otherwise, the matrix $J^T J$ will be singular which will result in numerical errors in estimation.. Therefore, this requirement is satisfied by selecting three non-collinear points for each random set.
2. Rule 2: An incorrect pose may be estimated if any of the 3D to 2D correspondences of the random set are incorrect. Thereby, it is desirable to terminate the subsequent steps (back-projecting the sampled points and calculating the errors) to save some computational load. The termination criterion used for this case is based on the assumption that the change in camera pose cannot be beyond a certain limit between two consecutive image frames. The limit is determined by the rate of image acquisition and the motion of the camera.
3. Rule 3: To reduce the sudden changes in the estimated camera poses, the camera poses are filtered out using Mahalanobis distance utilising the state covariance matrix S_t^p (Section 3.2.9). This strategy ensures the estimation of camera poses that are in agreement with the previous state covariance matrix and hence generates a smooth trajectory reducing jitter.

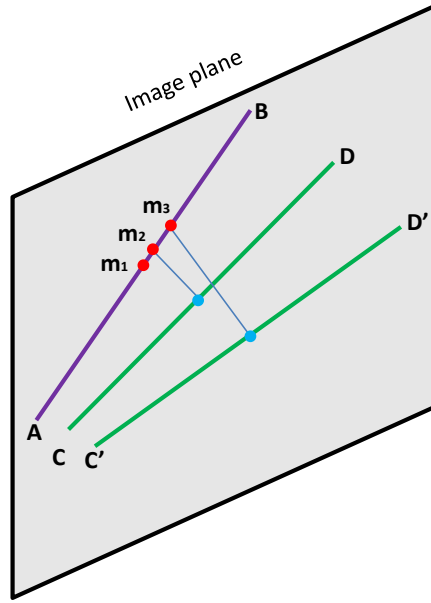


Figure 3.4: AB is an image edge, whereas CD and C'D' are the back-projected model edges using different poses. It can be seen that the projection of the back-projected sampled point for edge C'D' on AB is further away as compared to the edge CD. This difference creates a bad pose estimation during large translations and rotations.

Updating the correspondences

During a combined large rotation and translation of the camera, the back-projected model edges are shifted further away from the image edges. This shift results in the creation of erroneous 3D to 2D correspondences, in spite of eliminating the outliers by using the above mentioned MSAC algorithm, as shown in Figure 3.4. Thus, the estimated camera pose is close to the actual but not accurate. Hence, after the pose estimation by MSAC procedure, the model edges are back-projected on to the image using the estimated pose. Consequently, the 3D to 2D correspondences are updated and a refined pose is estimated iteratively by repeating the MSAC process until convergence.

Convergence criteria

At each iteration of MSAC pose estimation, the model edges come closer to the image edges, and therefore the residuals are minimised. In theory, after a sufficient number of MSAC iterations, the model edges should exactly overlap on the image edges, but due

to the presence of noise in the measurements, this never happens, and the residuals are never zero. Subsequently, a threshold (d_{thr}) for the residual is used as the convergence criteria. Hence, the MSAC iterations terminate if the residuals of all the correct correspondences (inliers) are below d_{thr} . The parameter d_{thr} (Section 3.3.2) is dependent on the resolution of the image. Moreover, the number of times the MSAC process will iterate is limited to a few iterations, and this limits the computational expanse in case of a non-convergence.

It is possible that the pose estimation by the proposed approach is not accurate leading to a wrong initialisation for the next image frame, which can result in loss of tracking. This is likely due to the following reasons. Firstly, when the camera moves very close to a wall or a door, which results in the lack of visible model edges or the presence of only parallel model edges. Secondly, motion blur can shift the edges by a short distance, and that results in the incorrect creation of 3D to 2D correspondences. Thirdly, sudden and sharp changes in the camera pose and this result in no 3D to 2D correspondence, as the search distance exceeds the actual shift between the model edges and the image edges. Lastly, presence of background clutter and occlusion that restricts the creation of correct 3D to 2D correspondences. However, the jump out rule in Section 3.2.8 (Rule 2), restricts abrupt changes in two consecutive frames. Moreover, the integrated Kalman filter restricts sudden changes in the trajectory.

3.2.9 Integration of a Kalman filter

A Kalman filter is used to predict the camera pose for the initialisation of the Gauss-Newton minimisation of the next frame. The Kalman filter predicts the parameters of a system, called state vector, through a model and updates the parameters utilising measurements which are associated with noise (Groves, 2013). In our Kalman filter, the state vector at time t is described by a twelve column vector $S_t = [X \ V_x \ \theta \ V_\theta]^T$ that contains the system parameters to be estimated, where $X = [X_o \ Y_o \ Z_o]$ is the system location, $V_x = [v_x \ v_y \ v_z]$ is the system linear velocity, $\theta = [\omega \ \phi \ \kappa]$ is the attitude and $V_\theta = [v_\omega \ v_\phi \ v_\kappa]$ is the angular velocity.

It is assumed that velocity remains constant throughout the trajectory. The reason for this assumption is that the camera is carried by the user (or mounted on the frame of a pair of smartglasses). Therefore the motion of the camera has the same characteristics of motion of the user, and such a motion can be approximated as a constant-velocity motion. Additionally, using a constant-velocity model for the Kalman filter facilitates the state prediction even when the camera is stationary. This condition is not true while using a constant-acceleration model, which will wrongly predict the noise of the pose estimates as accelerations of the camera motion.

Assuming that the time derivative of each state ($\dot{S}_t$) is a linear function of the other states and of white noise sources, the system model of a Kalman filter is described as:

$$\dot{S}_t = A_t S_t + G_t w_t \quad (3.14)$$

where A_t is the system matrix, G_t is the system noise distribution matrix, and w_t is the system noise which is assumed to be zero-mean Gaussian white noise. In the constant-velocity system model, since the acceleration is not estimated, this is considered as the system noise (Groves, 2013). The state dynamics can be described in a matrix form as:

$$\begin{bmatrix} \dot{X} \\ \dot{V}_x \\ \dot{\theta} \\ \dot{V}_\theta \end{bmatrix} = \begin{bmatrix} 0_3 & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix} \begin{bmatrix} X \\ V_x \\ \theta \\ V_\theta \end{bmatrix} + \begin{bmatrix} 0_3 & 0_3 \\ I_3 & 0_3 \\ 0_3 & 0_3 \\ 0_3 & I_3 \end{bmatrix} \begin{bmatrix} a_x \\ a_\theta \end{bmatrix} \quad (3.15)$$

where I_3 represents an identity matrix of dimension three, and a_x and a_θ are the respective linear and angular accelerations. Having formulated the system matrix and noise distribution matrix, the process noise covariance matrix Q_t is described (Eq. 3.16), assuming the process noise has zero mean multivariate normal distribution.

$$Q_t = GE \left(\int_{t-\delta t}^t \int_{t-\delta t}^t w(t') w^T(t'') dt' dt'' \right) G^T \quad (3.16)$$

where $E(\cdot)$ denotes expectation and δt is the sampling time. In our case, assuming that the accelerations are white noise, the matrix Q_t can be given by:

$$Q_t = \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & \eta_{ax}^2 \delta t I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & \eta_{a\theta}^2 \delta t I_3 \end{bmatrix} \quad (3.17)$$

where η_{ax}^2 and $\eta_{a\theta}^2$ are the Power Spectral Density of the accelerations. Note that the process noise in Eq. 3.16 - 3.17 is the result of discrete approximation of the constant velocity, and would involve cubic diagonals and quadratic off-diagonal terms in sampling time for a random walk model. To establish a relationship between the measurements and the state vector, a measurement model is defined (Eq. 3.18) to update the state vector estimates from a group of measurements.

$$M_t = H_t S_t + v_t \quad (3.18)$$

where $M_t = [X_o \ Y_o \ Z_o \ \omega \ \phi \ \kappa]$ is the vector containing the measurements estimated from the Gauss-Newton minimisation, H_t is the transformation matrix that maps the state vector parameters into the measurement domain and v_t contains the measurement noise of each measurement. Similar to process noise the measurement noise is assumed to have zero mean white noise with covariance matrix R_t .

Since the Gauss-Newton minimisation estimates the measurements of the Kalman filter, it is possible to directly calculate the covariance matrix R_t from the optimisation. The error propagation principle is applied on Eq. 3.9, to estimate the measurement covariance matrix R_t by Eq. 3.19.

$$R_t = (J^T J)^{-1} J^T Q_e ((J^T J)^{-1} J^T)^T \quad (3.19)$$

where matrix Q_e is the $2m \times 2m$ covariance matrix of the residual vector e , which is calculated from

$$Q_e = J_e Q_{xyXYZ} J_e^T \quad (3.20)$$

where J_e is the Jacobian matrix of the partial derivative of the residual vector for x, y on the image plane and X, Y, Z in the 3D space (Eq. 3.21). The matrix Q_{xyXYZ} is the covariance of the measurements on the image and 3D points in the BIM, and was calculated

from the accuracy of the BIM and the precision of the measurements on the image plane.

$$J_e = \begin{bmatrix} \frac{\partial e_x^1}{\partial x} \frac{\partial e_x^1}{\partial y} \frac{\partial e_x^1}{\partial X} \frac{\partial e_x^1}{\partial Y} \frac{\partial e_x^1}{\partial Z} \\ \frac{\partial e_y^1}{\partial x} \frac{\partial e_y^1}{\partial y} \frac{\partial e_y^1}{\partial X} \frac{\partial e_y^1}{\partial Y} \frac{\partial e_y^1}{\partial Z} \\ \dots \\ \frac{\partial e_x^m}{\partial x} \frac{\partial e_x^m}{\partial y} \frac{\partial e_x^m}{\partial X} \frac{\partial e_x^m}{\partial Y} \frac{\partial e_x^m}{\partial Z} \\ \frac{\partial e_y^m}{\partial x} \frac{\partial e_y^m}{\partial y} \frac{\partial e_y^m}{\partial X} \frac{\partial e_y^m}{\partial Y} \frac{\partial e_y^m}{\partial Z} \end{bmatrix} \quad (3.21)$$

where e_x and e_y are the vectors containing re-projection errors along the x and y directions on the image plane, and the superscripts $(1 - m)$ indicate the number of observations. The standard Kalman filter is based on Gaussian probability density functions. The variances and the covariances of the probability density functions of the state estimations and the measurements at time t are described by a covariance matrix P_t . The standard Kalman filter consists of two main steps, state prediction and measurement updates. The state prediction step comprises of predicting the state of the camera as well as the covariance matrix and is described by Eq. 3.22-3.23.

$$S_t^p = A_t S_{t-1}^p \quad (3.22)$$

$$P_t^p = A_t P_{t-1}^p A_t^T + Q_t \quad (3.23)$$

where S_t^p and P_t^p denotes the predicted values of the state and the covariance matrix at time t . The measurement update step consists of updating the predicted value of S_t^p and P_t^p based on the combined knowledge of the Kalman filter model by multiplying the probability density functions of the state estimations and the measurements at time t . The update steps are described by Eq. 3.24-3.25.

$$S_t^e = S_t^p + K_t (M_t - H_t S_t^p) \quad (3.24)$$

$$P_t^e = P_t^p - K_t H_t P_t^p \quad (3.25)$$

where S_t^e and P_t^e denotes the estimated values of the state and the covariance matrix at time t and K is the Kalman gain which is described by Eq. 3.26.

$$K_t = P_t^p H_t^T (H_t P_t^p H_t^T + R_t)^{-1} \quad (3.26)$$

For each frame of the camera, the values of the state S_t^p as well as the covariance matrix P_t^p are predicted using Eq. 3.22 and Eq. 3.23. Consequently, using the measurement vector M_t , which contains the estimated values of the previous pose parameters of the camera (from Eq. 3.10 after MSAC estimation), the values of the S_t^e and P_t^e are updated using Eq. 3.24 and Eq. 3.25. Subsequently, K_t is updated using Eq. 3.26. In the case that MSAC process provides a robust solution, the Kalman filter will have a large gain, therefore a high impact on the update of the state vector. On the other hand, if it does not provide a robust solution, the update will not have a high impact on the estimation, and the estimated values of the state vector S_t^e will be very close to the predicted value of the state vector S_t^p . During the experiments, it is shown that the predicted values of the state vector by the Kalman filter in the event of a tracking failure, provides a good initialisation for the next frame for tracking (explained in Section 3.3.3).

3.2.10 Reusing inter-frame correspondences

When the relative motion between the frames is very small, it is often unnecessary to perform a robust estimation. Instead, the correct correspondences from the previous frame can be reused to improve the running time. To reuse the correct correspondences of Frame t , the model edges are back-projected using the predicted camera pose from the Kalman filter using Eq. 3.22 for Frame $t + 1$. Subsequently, the corresponding image edges are identified in Frame $t + 1$ in a reduced neighbourhood, which is limited to a few times the standard deviation of the re-projection errors e of the Frame t . Next, these newly created correspondences are used to estimate the camera pose and the measurement covariance matrix by Eq. 3.19. The correspondences are only reused when the inlier ratio m/n exceeds 0.25 which corresponds to the worst case, considering two hypotheses for each sampled point. Also, the visible model edges are reused in the camera FoV of the previous frame without performing new graphical rendering that further improves the running speed.

3.3 Experiments and results

The correctness (and high accuracy) of ground truth plays an important role in the evaluation of localisation accuracy. Ground truth for the evaluation of trajectories in indoor spaces are usually collected by camera-based (Fod et al., 2002), motion capture (Huang et al., 2017), laser tracking (Teulire et al., 2015), surveying (Brki et al., 2010) or target tracking (Boochs et al., 2010) methods. All the methods, require specific hardware or platforms for the collection of the ground truth and usually have little coverage of indoor space. Additionally, there is a limit of the correctness of the ground truth generated by such methods. On the other hand, use of synthetic photo-realistic data provides absolute camera pose for each point of the trajectory without any error, and have been used in the literature to evaluate the accuracies of generated trajectories (Ramezani et al., 2017; Handa et al., 2014; Caron et al., 2014). Hence, in this research, the performance of BIM-Tracker is evaluated by experimenting with a dataset of synthetic photo-realistic images (Section 3.3.1). Additionally, a dataset of real images captured by a smartphone camera of the same corridor (Section 3.3.1) is used to qualitatively assess the performance of the approach with real images.

3.3.1 Datasets used

A Zebedee (Zeb1) spring mounted hand-held scanner is used for acquiring a point cloud of the corridor of the third floor of Block B of the Department of Infrastructure Engineering at the University of Melbourne, Australia. The reported absolute accuracies of the generated point clouds by the sensor are in the range of 3 - 40 centimetres (Khoshelham et al., 2017). A BIM of this part of the building, which is shown in Figure 3.5 was created manually from the acquired point cloud in an industry foundation class (IFC) format, which is the widely accepted data format of the BIMs. Subsequently, a 3D building model was derived from the BIM which was used in this research for performing the model-based visual tracking. In the current research, two sets of information were extracted from the BIM: 1) A low level-of-detail BIM containing geometries of the main structural elements and 2) the semantic information of the building elements such as walls, roof and floor.

Table 3.1: The details of each set of data

Set	FoV	Resolution	f (millimetres)	l_s (pix)	d_{search} (pix)	d_{thr} (pix)	d_{min} (pix)	pixel size (μ)	Remarks
Set 1	90°	320x240	16	5	50	1.5	2.5	100	Baseline exp.
Set 2	90°	640x480	16	10	100	3	5	50	
Set 3	90°	1280x960	16	20	200	6	10	25	
Set 4	60°	640x480	27.71	10	100	3	5	50	Portrait mode Occlusions Motion Blur
Set 5	120°	640x480	9.23	10	100	3	5	50	
Set 6	90°	640x480	16	10	100	3	5	50	
Set 7	90°	640x480	16	10	100	3	5	50	
Set 8	90°	640x480	16	10	100	3	5	50	

Table 3.2: The details of the camera parameters

Resolution	Frame rate	FoV	Sensor size (millimetres)
640x480	30	56.32°	3.63x2.72

Table 3.3: Error for each set of experiments

Translational and rotational error	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8
Average translational error (millimetres)	28.22	17.13	12.74	27.39	22.91	20.39	20.45	16.70
Average rotational error (Degrees)	0.21	0.12	0.08	0.22	0.14	0.13	0.11	0.11

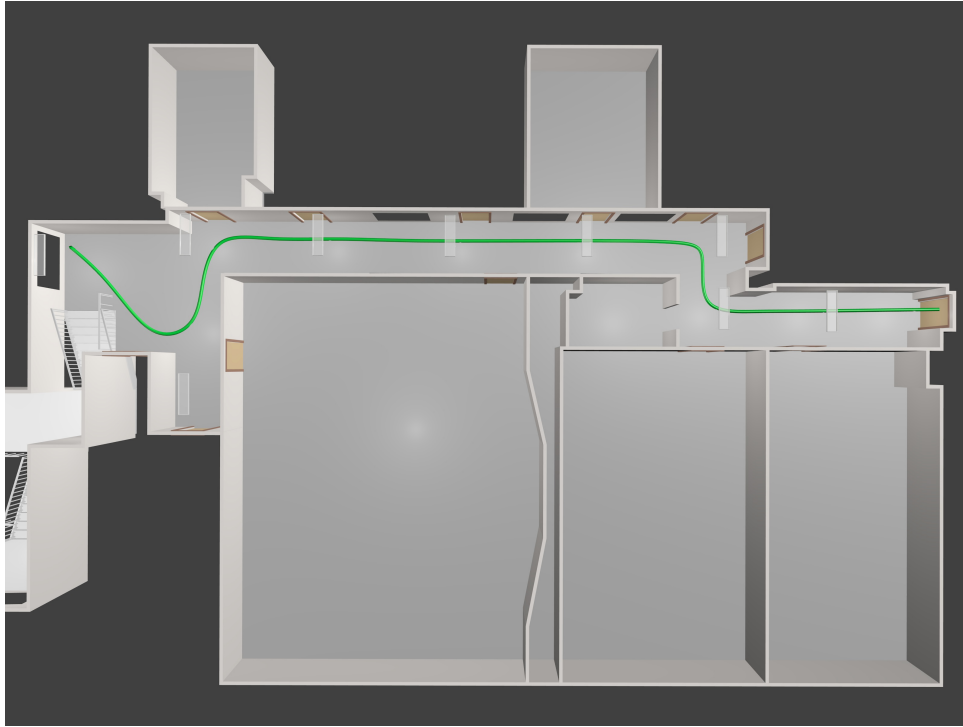


Figure 3.5: The created BIM from point cloud data. The ceiling is removed for better visualisation, but tube lights are retained. The green line denotes the trajectory of the virtual cameras.

A BIM can provide additional information about the building that is not possible to obtain from 2D representations such as building plans. The additional information includes semantic information, properties of each building element, functional states of the building elements and movable objects, spatial relationships between elements and geometric information. The use of such additional information to improve the localisation accuracy was beyond the scope of the current research and remains a part of future work.

Photo-realistic synthetic dataset

The building elements namely the walls, floor, ceiling, doors, stairs, railings and tube lights were imported from the created BIM in Blender, and texture was added to the walls, floor and roof. The scene of the corridor was illuminated by placing artificial light sources near the actual locations of the tube lights. Figure 3.5 shows the trajectory in the green colour that was defined inside the corridor for generating the ground truth for



Figure 3.6: An image frame from the set (a) Set 2 - having 90° FoV (b) Set 4 - having 60° FoV (c) Set 5 - having 120° FoV (d) Set 6 - in portrait mode (e) Set 7 - containing walking humans and potted plants as occlusions (f) Set 8 - having motion blur.

the poses of the camera. The height of the trajectory was kept in the range of 1.5 - 1.8 metres from the floor, considering the average range of heights for holding a smartphone in hand or the average height of eye level of the user in case of wearing a smartglass. Consequently, virtual cameras with different intrinsic parameters were traversed on the trajectory to render eight sets of photo-realistic images. The length of the trajectory was approximately 30 metres, and the average speed for traversal was 1.6 metres per seconds. Table 3.1 provides the intrinsic parameters of the cameras for each set and Figure 3.6 shows some of the outputs of the respective camera configurations. The sensor size for all the virtual cameras was set to 32 millimetres x 24 millimetres.

Note that for all the sets the FoV of the camera refers to the horizontal FoV of the camera, except for Set 6, where the referred FoV was in the vertical direction. For Set 7 of the dataset, 3D models of humans and potted plants were placed throughout the scene of the corridor to simulate the effects of occlusion and clutter. Additionally, a motion model was used for the 3D human models to simulate the impact of walking. The 3D model used to detect the visible vertices for Set 7, did not contain the 3D models of the human or the potted plants. For Set 8 of the dataset, motion blur was introduced in the photo-realistic rendered images to simulate the effects of camera motion. The ground truth trajectory and all the sets of the photo-realistic images were generated at a constant rate of 30 FPS.

Real dataset

A video sequence of the corridor was recorded by the camera of a smartphone (Motorola G 1st generation), which was calibrated beforehand, and Table 3.2 provides the details of the camera parameters. The smartphone was held in the hands near the eye level in landscape configuration during traversal at the average speed of walking of the user. The direction of the look of the camera was not constrained in the direction of traversal. The camera was calibrated using a 11x9 checkerboard pattern using MATLAB's camera calibrator toolbox, where the reprojection error was 0.14 pixels for 21 images. The focal length and the principle points were converted from pixel to millimetres using the known sensor size.

3.3.2 Experiments with photo-realistic synthetic data

Three experiments were conducted to evaluate the performance of BIM-Tracker under varying conditions. The varying conditions include the effects of 1) camera FoV, 2) image resolution and 3) clutter, occlusion and motion of the camera. Subsequently, a detailed analysis of the trajectory was done regarding error patterns and accuracy.

Baseline performance

The performance of set 2 of the dataset having a resolution of 640x480 pixels, a FoV of 90° , a frame rate of 30 FPS and in landscape orientation was used as the baseline. The average translational error was observed as 17.13 millimetres, and the average rotational error was 0.12° . The other design parameters of the proposed approach, which include l_s , d_{search} , d_{thr} and d_{min} , are set for each experiment to compensate the effect of the changed conditions from the baseline experiment. The following subsections explain in detail the changed conditions and their respective design parameters.

For the calculation of rotational errors, the difference of the ground truth angles and the estimated angles were taken. For convenience, the three angles were converted to one Euler angle and $\theta_{Euler} = \cos^{-1}([r_{11} + r_{22} + r_{33} - 1]/2)$, where r_{11} , r_{22} and r_{33} are the diagonal elements of rotational matrix R (Khoshelham, 2016). Error in translation was calculated as the average Euclidean distance for the whole trajectory with respect to the ground truth:

$$Average\ error = \frac{1}{s} \sum_{1}^s ||P_{gt}^s - P_e^s|| \quad (3.27)$$

where P_{gt}^s and P_e^s denotes the ground truth and the estimated point of the trajectory respectively, s is the number of points of the trajectory and $||.||$ denotes the l_2 -norm.

Effect of image resolution

To analyse the effect of image resolution on localisation performance experiments were conducted using Sets 1, 2 and 3. As the sensor size of the virtual cameras is the same for all the resolutions, the respective pixel sizes decrease with increasing resolution. Hence,

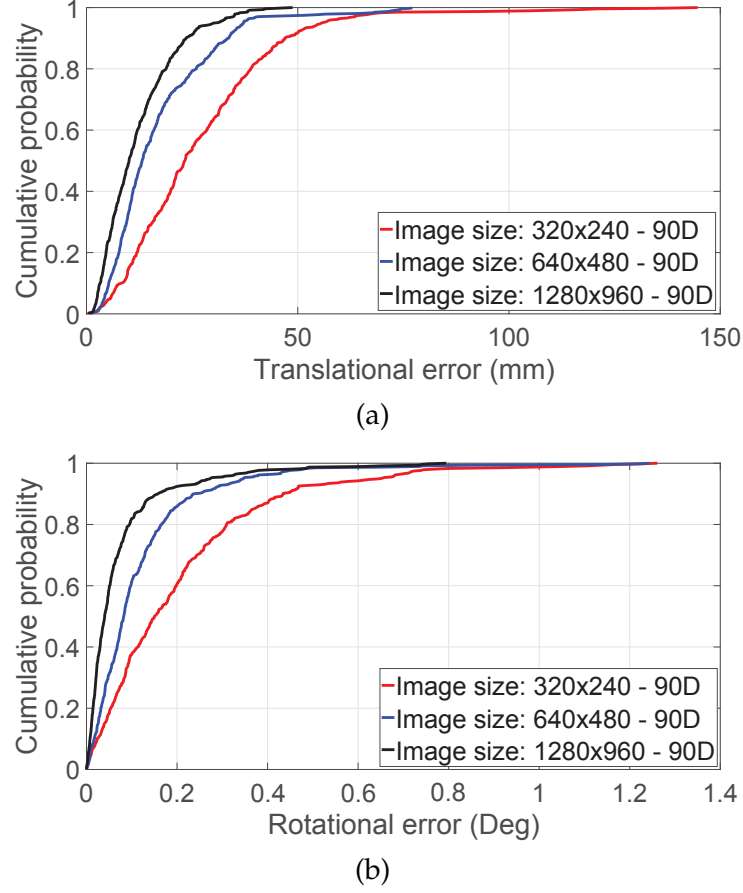


Figure 3.7: The cumulative distribution function (CDF) of errors for (a) translation and (b) rotation to evaluate the effects of image resolution.

the respective parameters l_s , d_{search} , d_{thr} and d_{min} for the resolutions were set in such a manner that they are identical on the image plane.

Figure 3.7 shows the cumulative distribution for the translation and rotational errors of different image resolutions. Table 3.3 provides the average translational and rotational errors for each set of the dataset. From Figure 3.7 it is observed that Set 3 having a higher image resolution was the most accurate and Set 1 was the least accurate regarding translational and rotational errors. This fact is also reflected in Table 3.3. The reason is likely because of the higher resolution of the images, individual edges are well resolved, in contrast to the congested edges which results in clutter for the image having lower resolutions. Thus for the higher resolution images, correct 3D-2D point correspondences are created, which results in the improved accuracy.

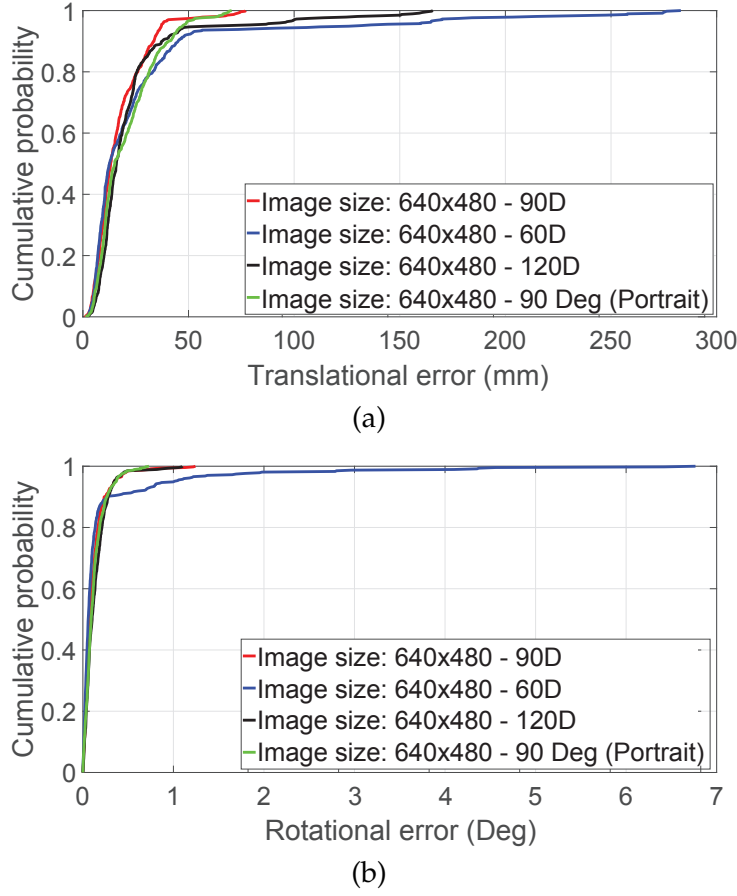


Figure 3.8: The Cumulative distribution function (CDF) of errors for (a) translation and (b) rotation to evaluate the effects of camera FoV.

Effect of camera FoV

A wide FoV (shorter focal length) enables more edge features to be observed in an image, which is a favourable condition for tracking in a constrained environment with fewer edge features, such as a long corridor or a tunnel. However, the quality of the edge features derived from a wide angle camera is inferior as compared to a narrow FoV camera having same image resolution, as a greater part of the environment is sensed by a smaller portion of the sensor.

As the horizontal and vertical FoV of most of the cameras are not equal, another factor to consider is the orientation of the image for tracking, which could be either in landscape or portrait configuration. For the case of a corridor in a landscape configuration, more

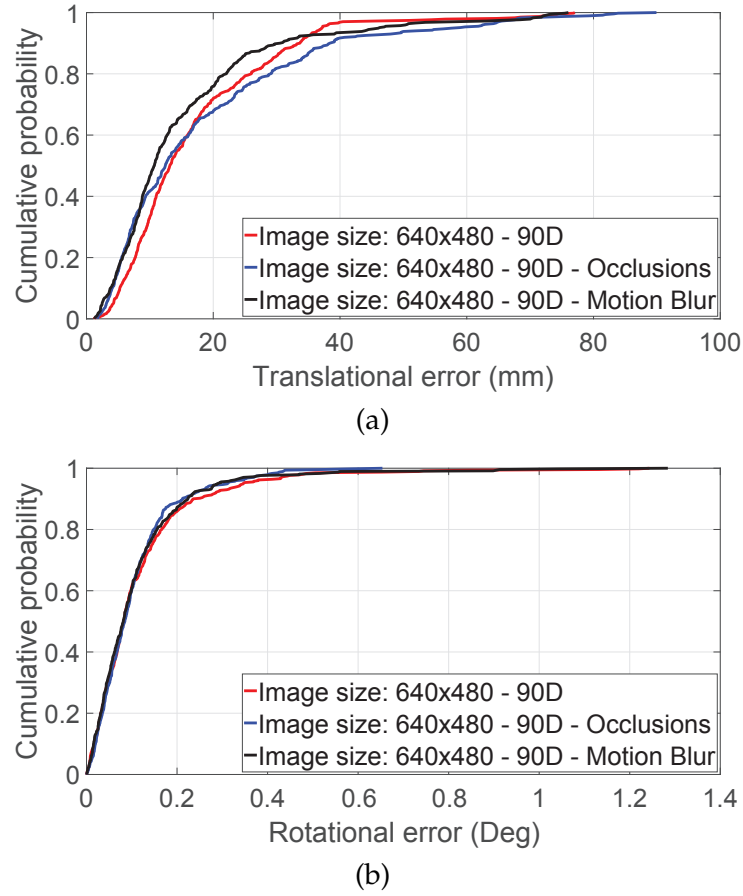


Figure 3.9: The Cumulative distribution function (CDF) of errors for (a) translation and (b) rotation to evaluate the effects of occlusion and motion blur.

features are captured along the walls as the horizontal FoV is greater than the vertical FoV. Whereas, for the case of the portrait configuration, more features are captured along the roof and the floor. Therefore, it is crucial to determine the correct configuration of the camera FoV for optimal tracking, and four sub-experiments were performed with Sets 2, 4, 5 and 6.

Figure 3.8 shows the cumulative distribution for the translation and rotational errors of different FoV of the virtual camera. From Figure 3.8 it is observed that Set 2 having a FoV of 90° is the most accurate regarding translational errors. Regarding rotational accuracies, Set 2, Set 5 and Set 6 are similar and are better than Set 4 having a FoV of 60° . From Table 3.3 it was observed that the Set 4, having the least FoV was the worst in terms of translational and rotational accuracies. The reason is plausible because of the lower

FoV of the camera, for some frames, there were not enough number of edges to facilitate the tracking, and this either led to the estimation of bad poses or tracking failures.

The accuracy of Set 6, having a portrait configuration was marginally lower than the landscape configuration, and this can be explained by the absence of the edges near the camera view, in this configuration. Additionally, for a long corridor, the wall-roof and wall-floor edges cover only a fraction of the image, and the edges far away from the camera cover a small part of the image. This configuration forms a bad geometry for pose estimation, as it is desired that the edge features are distributed evenly along the whole image. In other words, it is desirable to have long edges near the camera, and this helps in better tracking as compared to the edges that are far from the camera. Hence, landscape configuration is better suitable for tracking tasks, where a wider view increases the visibility of edges near the camera.

Note that the accuracy of Set 5, having a higher FoV was not the best. As the resolutions of all the images were same, the low accuracy can be explained by the image edge clutter that results due to the presence of a higher number of image edges in a small area. Additionally, as described above this forms a bad geometry for long corridors for pose estimation.

Performance in presence of occlusion and motion blur

Due to the presence of external objects such as people, which are not present in the 3D model, but are present in the image derived from the camera, occlusions become a problem for the tracking. This problem is because of the decreased inlier ratio with the increased occlusions, and thus results in wrong correspondences if the inlier ratio is lower than the least assumed value. Additionally, due to the motion of the camera, blur is introduced in the images, which is another challenge. Due to the motion blur, the sharpness of the edges are lost, and this results in fewer image edges for matching. Therefore to evaluate the effects mentioned above on the localisation accuracy, three sub-experiments were conducted using Sets 2, 7 and 8.

Figure 3.9 shows the cumulative distribution of the translation and rotational errors for different environmental conditions, such as the presence of occlusions and motion

blur. From Figure 3.9 it is found that the translational and rotational errors of the three sets were similar. From Table 3.3 it is observed that there is a drop in the accuracy of the sets containing occlusions (Set 7) as compared to the baseline performance (Set 2). Set 7 was less accurate as compared to Set 8 and this is because of the decreased inlier ratio for Set 7 due to the presence of occlusions. It is also observed that the presence of motion blur does not degrade the translational or rotational accuracies.

Trajectory analysis

Figure 3.10 shows the average normalised error distribution along the trajectory for the sets having a FoV of 90° , i.e. sets 1-3, 6-8. The images in the bottom of the Figure 3.10 shows the corresponding views of the camera for Set 2 at the marked location along the trajectory. It can be seen that at point A of the trajectory, the combined error is highest. From the corresponding camera view (A) it can be observed that most of the visible model edges are distributed in the centre of the image such as the edges derived from the doors and walls. A very few visible model edges are present along the other parts of the image such as the edges derived from the intersection of the wall-roof and wall-floor. This configuration forms a poor geometry for the estimation of the pose and explains the error at Point A and Point D. Additionally, this configuration accounts for the poor accuracy of Set 5 that has a wide FoV.

Points B, C and E of the trajectory are a part of the turns of the trajectory. Due to the combined effect of motion blur and large translation and rotation, the back-projected model edges are distant from the corresponding edges in the image. This effect increases the number of wrong correspondences and explains the estimation of bad poses. Additionally, for Points B and C the number of visible model edges in the whole image is low. For Point F of the trajectory, the presence of railings of the staircase creates a dense clutter. The number of model edges to represent the railings are higher as compared to the total number of lines representing the other building elements, and this creates wrong 3D-2D correspondences and explains the poor accuracy for that part of the trajectory.

Figure 3.11 shows the plot of the errors for each dataset for all the frames. It is evident that there is no accumulation of errors in any of the datasets. Additionally, there is a clear

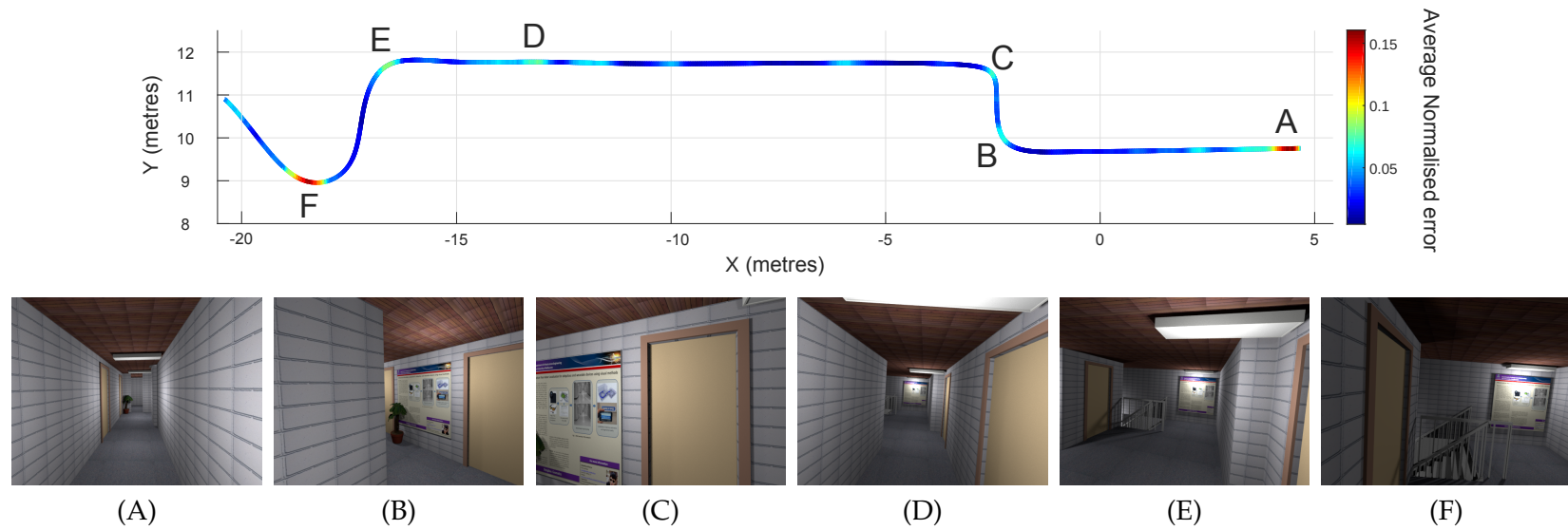


Figure 3.10: The distribution of errors along the trajectory. The colour represents the value of the average normalised error for the sets having a FoV of 90° . Below are the corresponding views of the points on the trajectory.

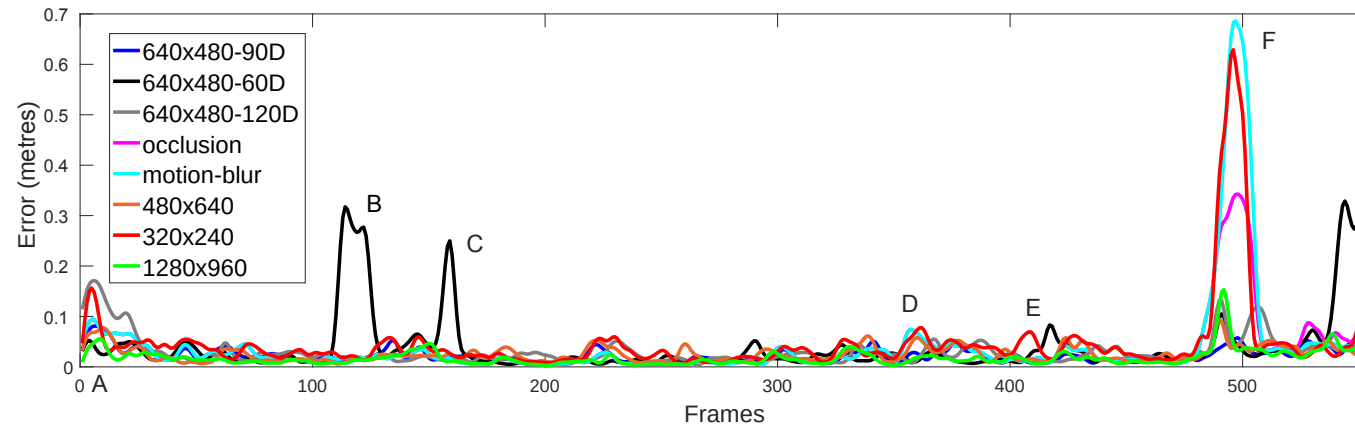


Figure 3.11: Localisation error calculated with respect to ground truth for all the frames of the trajectory.

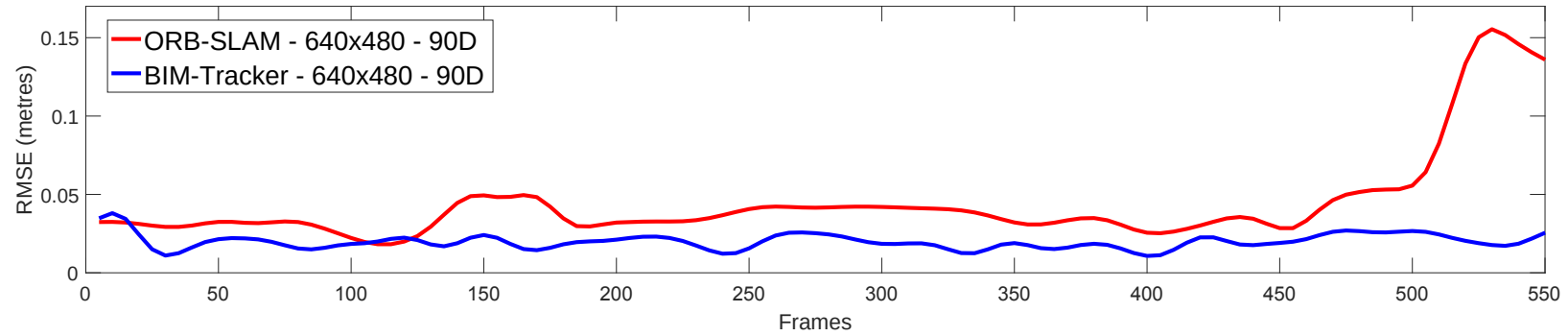


Figure 3.12: The per frame RMS errors with respect to the ground truth for ORB-SLAM and BIM-Tracker for the synthetic baseline dataset (Set 2).

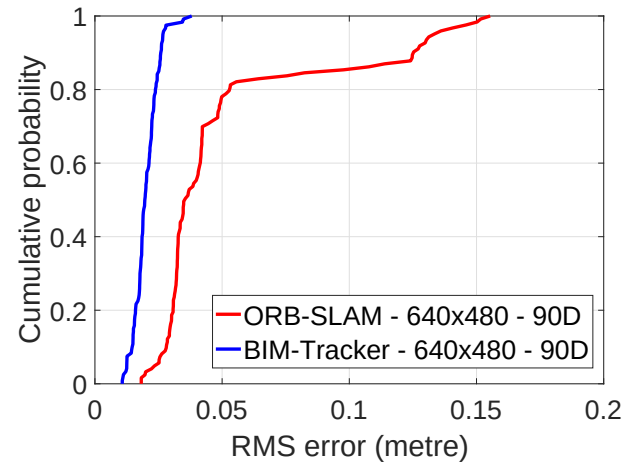


Figure 3.13: The cumulative distribution function (CDF) of RMS errors with respect to the reference trajectory for ORB-SLAM and BIM-Tracker.

indication that a camera having a narrow field of view (Set 4) estimates inaccurate poses near B and C of the trajectory. Moreover, the maximum errors are contained near the part F of the trajectory, which can again be explained by the presence of the railings of the staircase.

To evaluate the performance of BIM-Tracker in comparison with the state-of-the-art algorithms, the results are compared with ORB-SLAM (Mur-Artal et al., 2015), which is a feature-based monocular SLAM approach. The intrinsic parameters of the camera were encoded along with the default parametrisation² of the ORB-SLAM. The obtained trajectory from ORB-SLAM was scaled and oriented manually by comparing with the ground truth trajectory.

Figure 3.12 shows the per frame root mean square (RMS) error of the trajectories estimated by ORB-SLAM and BIM-Tracker with respect to the reference trajectory. In contrast to ORB-SLAM, from Figure 3.12 it is evident that the error of estimation does not increase with the length of the trajectory for BIM-Tracker, and therefore the drift in the trajectory is eliminated. Figure 3.13 shows the CFD of the RMS errors for ORB-SLAM and BIM-Tracker with respect to the reference trajectory and indicates that the BIM-Tracker is more accurate as compared to ORB-SLAM.

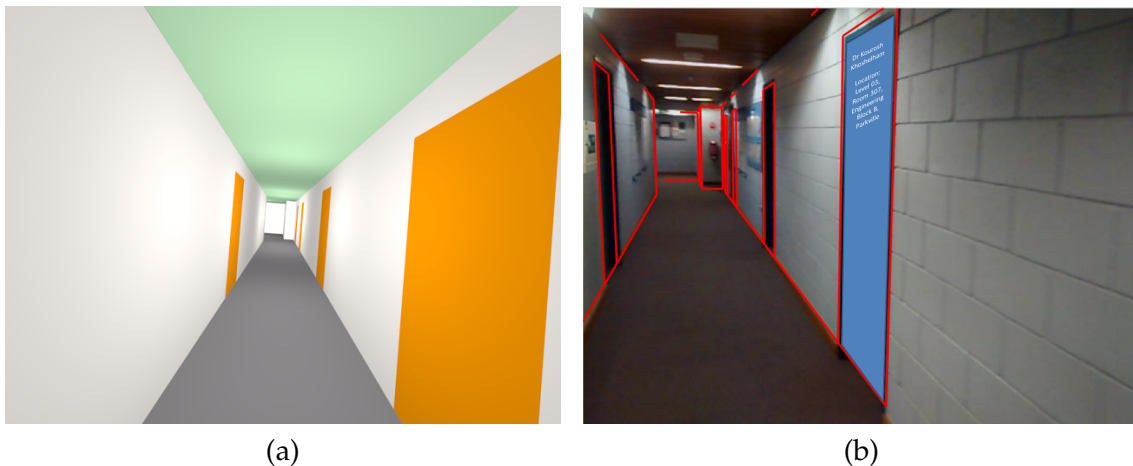


Figure 3.14: (a) The low level-of-detail model used for the tracking, containing only walls, floor, ceiling and doors. The geometry of the wall-door intersection was kept, and other details, such as the frame of the door was simplified (b) Projection of a banner for evaluating the estimated pose for augmented reality applications

²https://github.com/raulmur/ORB_SLAM/blob/master/Data/Settings.yaml

3.3.3 Experiment with real data

An experiment was conducted with the real dataset using the configurations of l_s , d_{search} , d_{thr} and d_{min} , same as Set 2, due to identical image resolutions. To correct for lens distortion of the images, Eq. 3.4 was used.

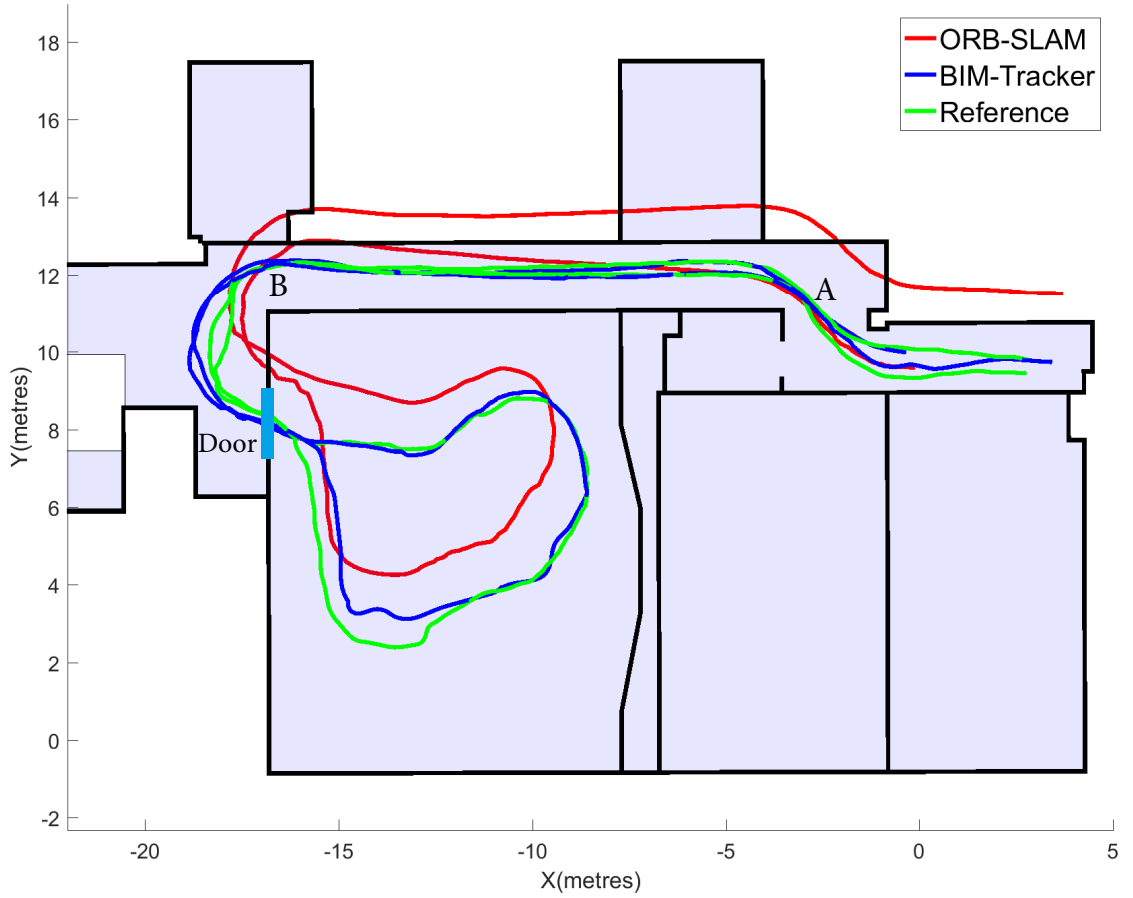


Figure 3.15: The reference trajectory and the trajectories estimated by ORB-SLAM and BIM-Tracker. The door is represented in light blue colour.

The dynamic nature of some of the structural elements of the building, such as a door (could be open or close), creates ambiguities between the 3D building model and the images from a real camera. Also, there are ambiguities created due to the deviation of the design type and the actual design of the building elements, for example, the design of the frame of a door. However, the geometry of the structural elements remains constant, such as the edge between a wall and a door. Hence, to reduce the ambiguities, a low level-of-

detail 3D building model is used containing only the geometries of the main structural elements of the building. Figure 3.14 (a) shows the low level-of-detail model used for the tracking containing only the main structural elements of the corridor.

Figure 3.14(b) shows that the estimated pose by using real images and the low level-of-detail model is sufficiently accurate to project a banner on a door demonstrating the potential of BIM-Tracker for augmented reality applications.

The results are compared with ORB-SLAM for real image dataset. Also, a reference trajectory is created in Agisoft PhotoScan Professional[®] software that uses bundle adjustment for estimating and refining camera poses. The 3D coordinate of the reconstructed environment were provided manually for performing accurate bundle adjustment, resulting in the re-projection error of 0.36 pixels, and the reconstruction error of 4.65 millimetres.

Figure 3.15 shows the estimated trajectory from the real dataset by ORB-SLAM, our approach and the reference trajectory, where the starting and finishing locations were different. The camera was traversed back and forth in the hallways and circled inside a room. The length of the generated trajectory is approximately 75 metres. It is observed that for all the frames the estimated trajectory by BIM-Tracker is within the navigable space of the corridor and very close to the reference trajectory. Moreover, Figure 3.15 indicates that there is no accumulation of errors even after taking turns near Point A and B and traversing the room, as there are overlaps at parts A and B of the trajectory as well as near the door. In contrast, ORB-SLAM clearly suffers from a drift after a turn near Point A, and the estimated trajectory lies outside the navigable space of the corridor.

3.3.4 Computational times

The average time required for processing each frame was approximately one second on a desktop computer having an i7 processor at the speed of 2.6 GHz and 16 GB memory. Table 3.4 shows the average time requirement of each step of the localisation. The approach was implemented in MATLAB without any hardware optimisation. It is expected that implementation on a suitable platform will improve the computational times for real-time operations.

Table 3.4: The average time for each step of localisation

Process	time (sec)
Visible edge detection	0.11
Image edge detection	0.03
Point sampling	0.01
Search for correspondences	0.01
MSAC (selection of best set)	0.37
MSAC (updating correspondences)	0.19
MSAC (pose estimation by least-squares)	0.16
Kalman filtering	0.01
Total average time	0.89

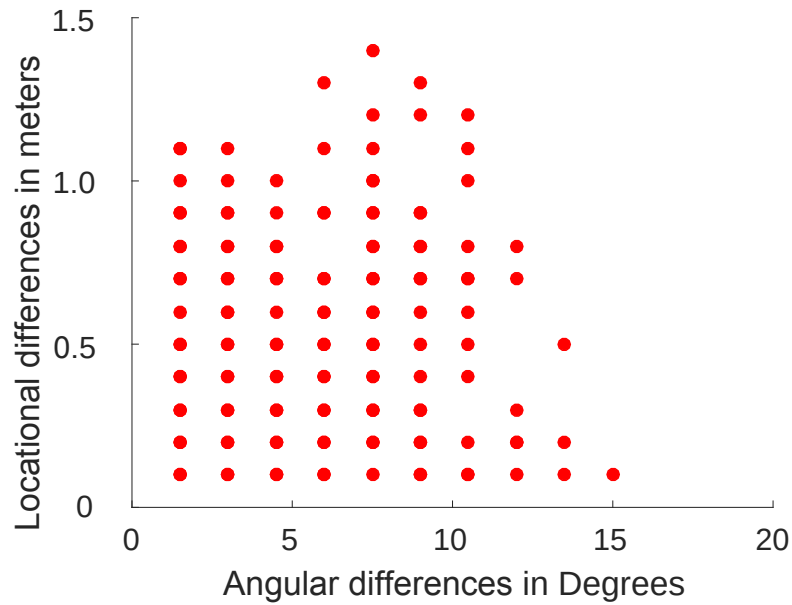


Figure 3.16: The region of convergence for pose initialisation. A red dot indicates convergence, whereas the absence of red dot indicates the region of failure of tracking. Euclidian distances are used to calculate the locational differences and Euler angles are used to calculate the angular differences.

3.3.5 Region of convergence

In an indoor environment, a small change in the camera pose results in large displacements of the back-projected model edges. Considering the clutter present due to the edges of the image, a wrong initial camera pose can lead to failure in the tracking. Thus, experiments are conducted with the real data to analyse the region of convergence for different values of the initial pose. Since ground truth was not available for the real data, the true poses of the camera were estimated manually for a few random frames. Subsequently, the poses are selected randomly in the interval of $[\pm 1500 \text{ millimetres}]$ for translation and $[\pm 20^\circ]$ for rotation from the true poses of the camera. For convenience, the Euler angles were converted to a single Euler angle (Section 3.3.2). Figure 3.16 shows the region of convergence for pose initialisation with respect to the previous pose as a function of Euler angles and Euclidean distances. From Figure 3.16 it is observed that the approach can handle combined rotational and translational deviation in the range of approximately $\pm 9^\circ$ and $\pm 700 \text{ millimetres}$ respectively for each frame.

This experiment also provides an insight into the minimum frame rate required for normal operation. For example, assuming the average walking speed of a human to be 2 metres per seconds the tracking can be performed successfully at a minimum frame rate of three FPS.

3.3.6 Discussions and future directions

Although BIM-Tracker is quite robust against motion blur, occlusions and illumination variations, it fails under notoriously challenging conditions. Firstly, in the presence of heavy motion blur, the edges extracted by the edge detector might be missed, noisy or displaced by a few pixels. Therefore, wrong 3D-2D correspondences will be generated that results in the estimation of a bad pose. Figure 3.17 (a) shows one of such poses where the image blur caused the failure of tracking. Although the Kalman filter can provide a good initialisation for a jerky frame, the uncertainty of the pose increases for prolonged motion blur thus leading to a failure, and this usually happens during long turns.

Secondly, the narrow FoV of the camera ($\approx 56^\circ$), results in the low number of visible

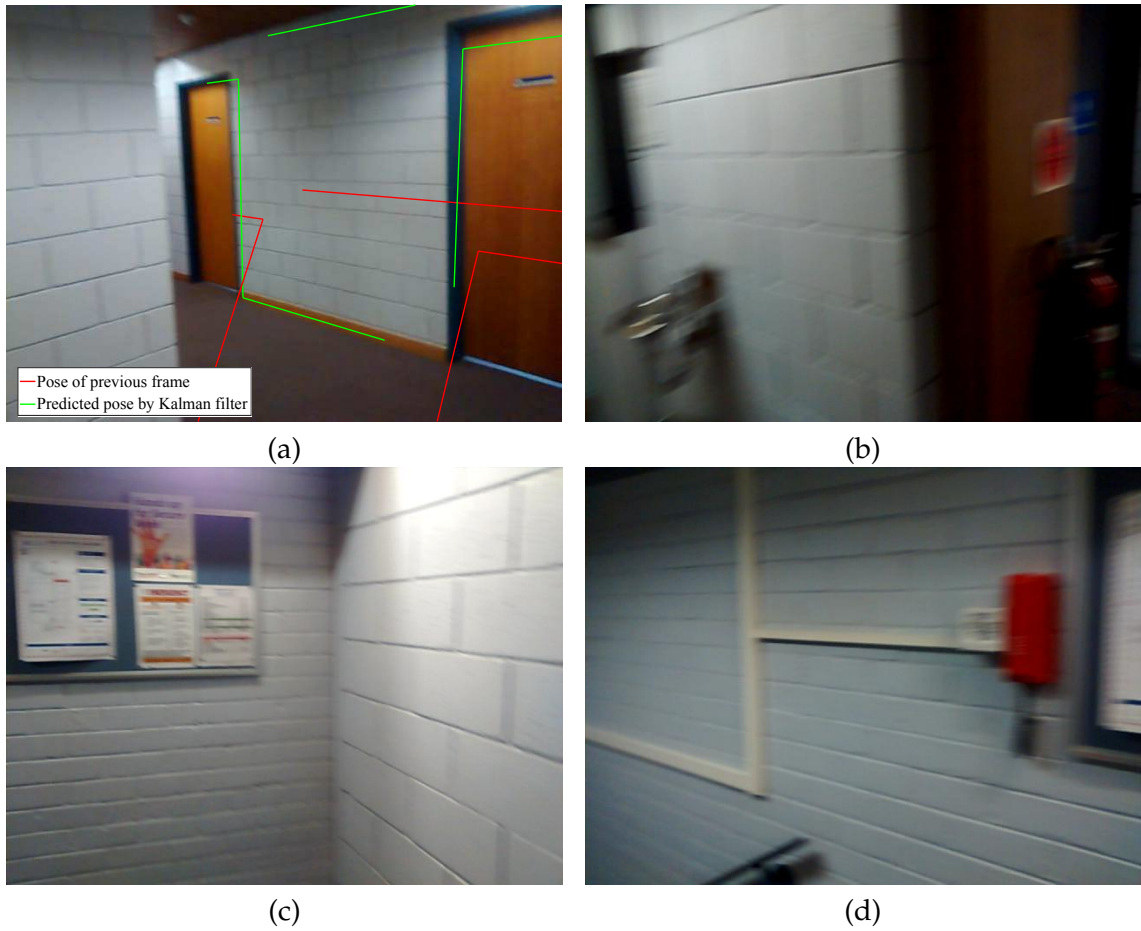


Figure 3.17: (a) The effect of Kalman filter. The Kalman filter was able to predict a better pose (green colour) compared to the pose estimated from the previous frame by Gauss-Newton optimisation (red colour) for a blurry frame. (b) A severely blurry frame during a turn having one model edge visible (c) A frame having one model edge visible (d) A frame having no model edge visible.

edges observed in the images when the camera is close to the wall. For instance, consider Figures 3.17 (b)-(d) where only one or none model edges are visible, the tracking failed completely after a few frames, and the estimated trajectory was the prediction of the Kalman filter with increased uncertainty. Moreover, the uncertain camera poses causes the wrong rendering of the model edges that results in complete failure of the tracking. Thirdly, during sharp rotations, there is a large displacement of the image edges. If the search distance d_{search} is less than the displacement, very few correct correspondences are made that results in the estimation of bad camera poses. On the other hand, using

a long search length is computationally expensive and often not necessary. Lastly, an inaccurate 3D building model can impact the camera pose estimation and can lead to tracking failure, for instance, incorrect locations of walls, floor or roof in the 3D building model.

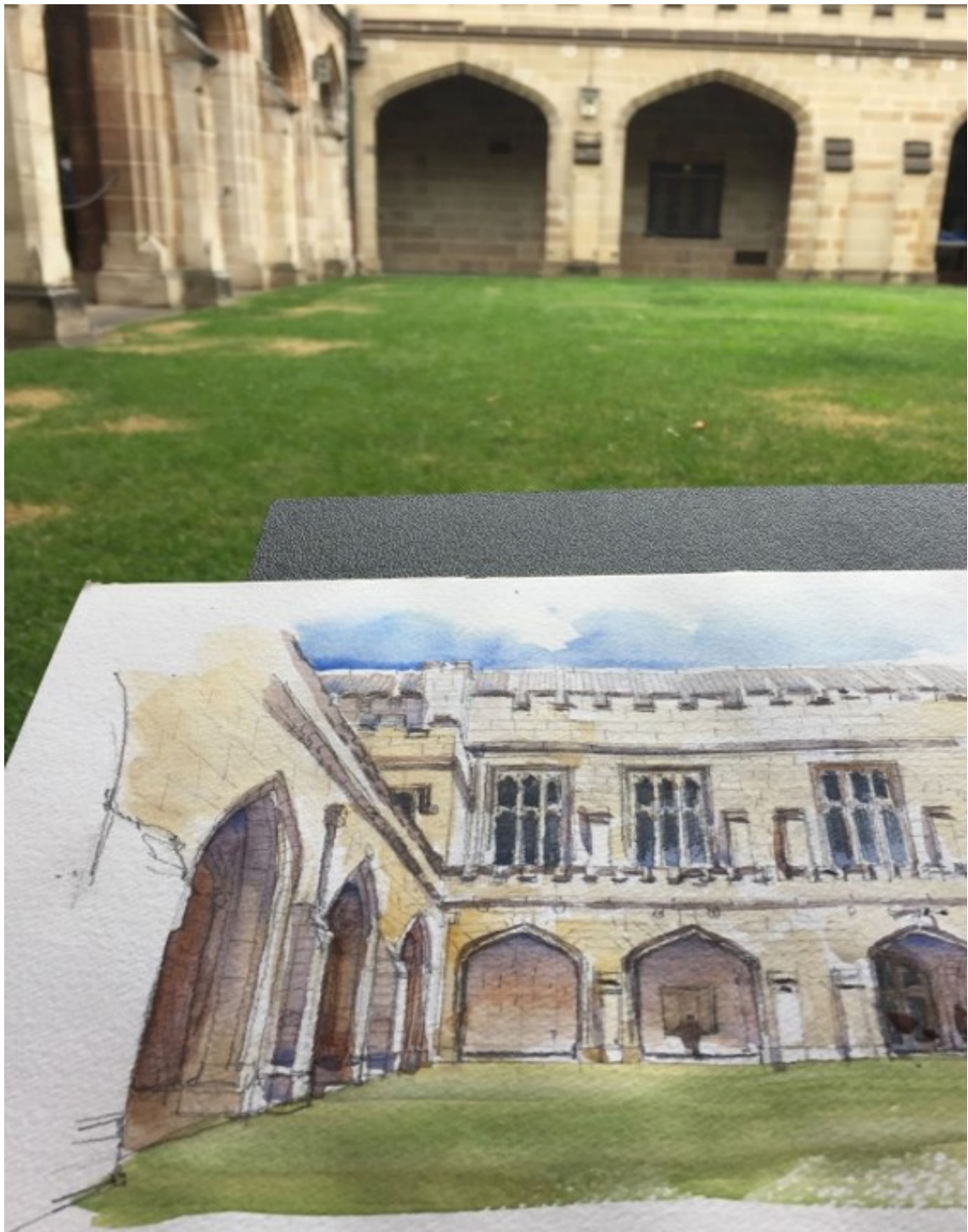
The main limitation of BIM-Tracker is the inability to recover the camera pose after a complete loss of tracking as discussed in the above paragraph. Integration of additional sensors such as the data from an inertial measurement unit (IMU) can improve the robustness of BIM-Tracker in the event of complete failure of tracking. However, that does not eliminate the necessity of an initial pose for starting the tracking, and a suitable reinitialisation process is desirable. Hence, as a part of the future work, a proper reinitialisation process (such as the work of [Kendall et al., 2015](#)) is proposed to be incorporated with BIM-tracker along with the integration of an IMU.

3.4 Conclusions

An approach is developed for performing infrastructure-independent indoor localisation by performing a model-based visual tracking. Evaluation of the approach suggests that a localisation accuracy of 10 centimetres can be achieved using a low level-of-detail 3D building model derived from a BIM. Moreover, there is no accumulation of error, which makes this approach suitable for indoor localisation tasks for long periods. Experiments with photo-realistic synthetic data suggest that a higher resolution of the image and a moderate FoV of the camera is suitable for accurate localisation. Additionally, the approach is robust against occlusions and motion blur of the camera.

A detailed evaluation of the trajectory reveals a systematic error in the localisation when there is no building structures in the FoV of the camera. Experiments with real datasets suggest that the pose estimated by the approach is suitable for augmented reality applications. The achievable accuracy is sufficient to localise within the navigable space of a corridor. Additionally, the range of convergence suggests that the approach can handle rotational and translational deviations which are sufficient for indoor localisation. Motion blur due to jerky camera motion for the real dataset was identified as the major

challenge in addition to the lack of visible edges for continuous tracking. Future work includes the integration of IMU data for increased robustness during sharp turns and automatic pose initialisation at the start of the tracking and in the event of a tracking failure.



"... Transfer learning is the application of gained knowledge to solve different but related problems ..."

≈ Old Quadrangle, The University of Melbourne, Australia
(Courtesy: <http://stephaniebower.blogspot.com/2017/05/>)

Chapter 4

BIM-PoseNet: Camera localisation using a 3D building model and deep learning

Existing approaches for indoor localisation using images either require an initial location or need first to perform a 3D reconstruction of the whole environment using structure-from-motion (SfM) approaches, which is challenging and time-consuming for large indoor spaces. In this chapter, a visual localisation approach is proposed to eliminate the requirement of image-based reconstruction of the indoor environment by using a 3D building model. A deep convolutional neural network (CNN) is fine-tuned using synthetic images obtained from the 3D building model to regress the camera pose. Results of the experiments indicate that the proposed approach can be used for indoor localisation in real-time with an accuracy of approximately 2 metres.

4.1 Introduction

As discussed in Section 2.3.2, the visual approaches that are independent of initial location are image retrieval or pose regression from a database of images or image features (Piasco et al., 2018). These approaches either require a database of images with known pose (Arandjelovic and Zisserman, 2012; Radenović et al., 2016) or additional data sources such as 3D point clouds (Irschara et al., 2009) or depth data of the scene (Shotton et al., 2013). However, the creation of a database of images with known pose and acquisition of additional data for large indoor spaces is a challenge due to the cost, time and post-processing involved.

A viable solution to this problem is a pose regression network, such as *PoseNet* (Kendall et al., 2015), which is fine-tuned with real images of known poses that are obtained us-

ing the structure-from-motion (SfM) (Furukawa and Ponce, 2010) approaches. The major limitation of *PoseNet* and its following approaches (Kendall and Cipolla (2016); Kendall et al. (2017); Walch et al. (2017); Clark et al. (2017)) is the requirement of a 3D reconstructed model derived from the SfM approaches. The SfM approaches require capturing images of the whole indoor space in advance, which is a laborious task. In addition, the SfM approaches are slow and error-prone.

The objective of this research is to investigate whether pose estimation can be done by fine-tuning a pre-trained network using synthetic images derived from a 3D building model rather than geotagged real images or images from the SfM approaches. A solution is proposed in this chapter to indoor camera pose regression based on a 3D wire-frame model (referred to as *3D building model* henceforth), which is derived from a building information modelling (BIM). The proposed approach eliminates the requirement of performing the SfM approach and does not need a database of images, point clouds or depth data. Furthermore, it is showed that the approach can work with low level-of-detail 3D building models. The following are the main contributions:

1. The proposed approach eliminates the requirement of performing SfM approaches, which is the prerequisite for the existing visual localisation approaches that provide the initial location. This is achieved by fine-tuning a pre-trained deep convolutional neural network (CNN) with synthetic images derived from the 3D building model with known camera poses, and testing the network with real images.
2. The localisation accuracy is improved by pre-processing the fine-tuning and test images and accuracy of approximately 2 metres is achieved in real-time. The results are compared with a test benchmark that was performed on real images.
3. The effects of level-of-detail of the 3D building model used for fine-tuning the deep CNN network is assessed.
4. A new dataset is introduced that is derived from the corridor of a building, covering an area of 43 m^2 , which extends the existing room-level indoor datasets (Shotton et al., 2013; Handa et al., 2014).

4.2 Methodology

The proposed approach is based on *PoseNet* architecture (Kendall et al., 2015). Figure 4.1 shows the design of the approach, where slightly modified versions of *GoogLeNet* (Szegedy et al., 2015) are fine-tuned individually with different sets of synthetic images with known pose, rendered from a 3D building model. Before fine-tuning, the weights of the network are updated from a pre-trained network of *GoogLeNet* that is trained on *Places* dataset (Zhou et al., 2014), which is a large collection of images of different places. Subsequently, the fine-tuned networks are tested with real images captured in the indoor environment to evaluate their pose regression ability.

4.2.1 Network architecture and fine-tuning

As shown in Figure 4.1 the architecture of *PoseNet* proposed by Kendall et al. (2015) is adapted, which is essentially *GoogLeNet* with 23 layers and containing six *inception* modules. An additional fully connected layer having a dimension of 2048 is added before the final regression layer, which serves as the localisation feature vector. The softmax classification layers are replaced by affine regressors, and the last layer is modified to produce two outputs having a dimension of three (location) and four (orientation).

During the fine-tuning phase, the inputs to the network are the synthetic images and their known pose at which they are rendered in the 3D building model. At the test phase, the pose of the images is unknown and is to be estimated by the network in the coordinate system of the 3D building model. The pose vector p can be represented by Eq. 4.1.

$$p = [x, q] \quad (4.1)$$

where $x = [X, Y, Z]$ is a vector representing the location of the camera and $q = [w, p, q, r]$ is a vector representing the orientation of the camera in quaternion format.

Kendall et al. (2015) show that using the weights of a pre-trained network on *Places* dataset is suitable for fast convergence as compared to networks pre-trained on other datasets such as *ImageNet* or networks that have random initialisation. Hence, in this research, the weights of the *GoogLeNet* pre-trained on *Places* dataset is updated. The

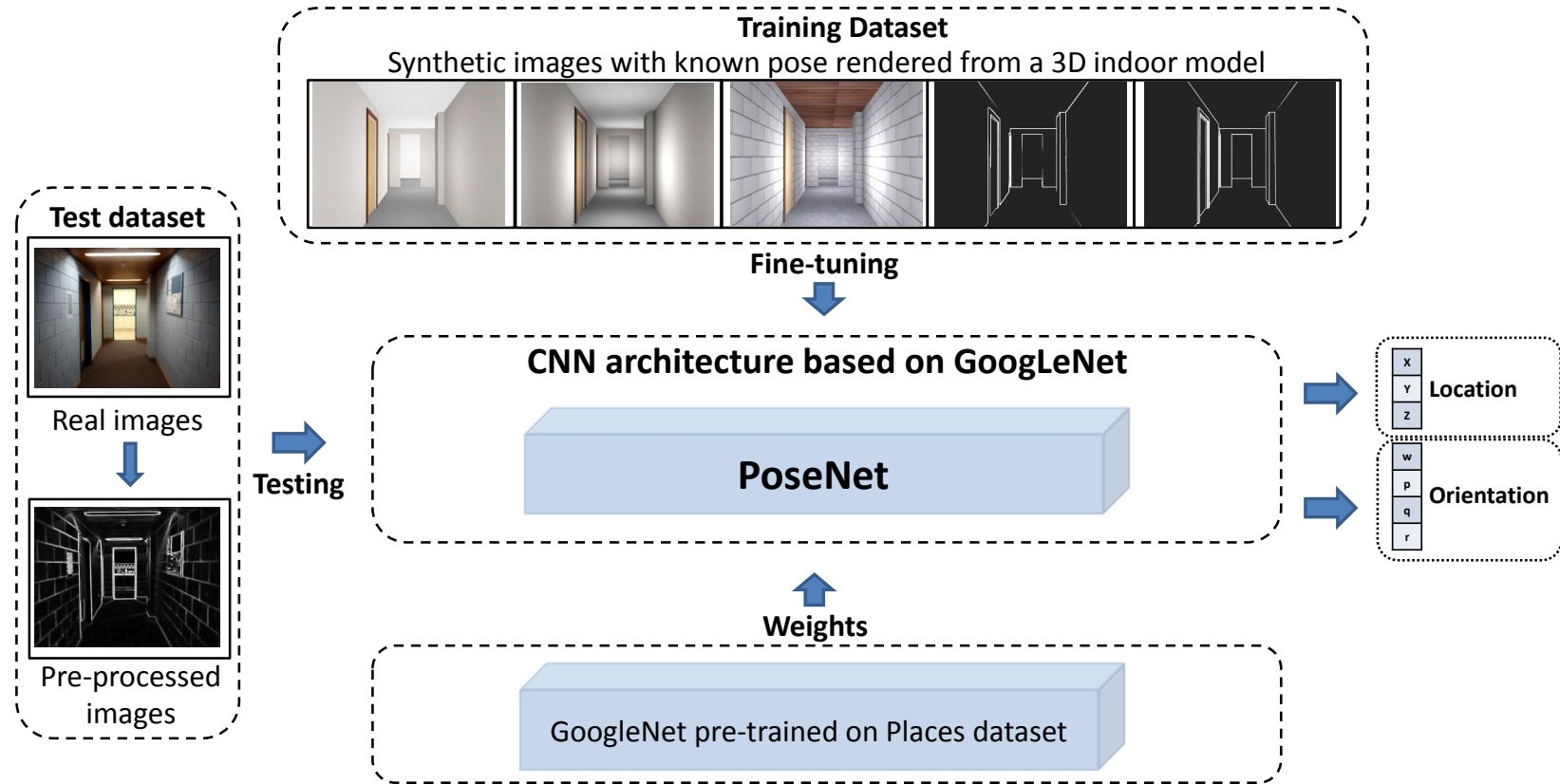


Figure 4.1: The design of BIM-PoseNet.

weights of the pre-trained network are suitable for the specific task of classifying places, and hence can not be used directly for the task of pose regression. Hence, the network is fine-tuned using the objective loss function suggested by [Kendall et al. \(2015\)](#):

$$loss(I) = ||\hat{x} - x||_2 + \beta \left\| \hat{q} - \frac{q}{||q||} \right\|_2 \quad (4.2)$$

where $\hat{x}$ and $\hat{q}$ are the observed value of location and orientation vector for an image I and β is a scaling factor to balance the error of location and orientation vectors. The value of β is dependent on the indoor space and is experimentally determined.

4.2.2 Use of 3D building model

The present research evaluates a pre-trained deep CNN fine-tuned with the different synthetic images with known pose to regress the unknown pose of real images. A 3D building model is used to generate different sets of synthetic images of the indoor space as described in Section [4.3.1](#).

The deep networks learn a hierarchy of increasingly complex features at each layer. Therefore, the lower layers learn generic low-level image features such as edges and textures. The deeper layers can differentiate high-level features such as windows, doors, roads, sky and patches of grasses. Moreover, these high-level features serve as image landmarks that the networks use for localisation.

[Peng et al. \(2015\)](#) train linear classifiers on features derived by deep CNNs from a few rendered synthetic images. The authors show that features derived from deep CNNs are invariant to colour, texture, pose and context (background). In other words that means if a network is invariant to an object's texture (such as a bird's texture), then it will have similar activations of neurons for the object with or without texture. The network *hallucinates* the right texture when given a texture-less object's shape. Therefore, if a network fine-tuned with synthetic images can recognise objects with different colour and texture in real images, then it is believed that a network fine-tuned with synthetic images with known pose should be able to estimate the pose of real query images.

The benefit of fine-tuning a pre-trained network is the requirement of only a few fine-tuning images compared to millions of images to train a network from scratch. A pre-trained network learns generic features that can be used for other tasks as well by leveraging transfer learning. As rendering millions of images is a time consuming and computationally expensive task, a pre-trained network is used and fine-tune with the synthetic dataset for pose regression. Therefore, different sets of synthetic images are created to assess the effects of different fine-tuning data on the pose regression ability of the network. The question is whether the network can learn relevant high-level geometrical features in the synthetic images that will be present in the real images for accurate pose regression. The relevant high-level geometrical features could be the doors, ceiling tube-lights or edges corresponding to a wall-floor intersection.

Fine-tuning a network that was initially trained for classifying real images, for pose regression with synthetic images, could be challenging. The task is more complicated than it seems, as now the network has to regress continuous pose values by looking into entirely different image examples that have similar geometry. For instance, consider the case, where the network sees synthetic examples, such as Figure 4.2 (a) - (c) during fine-tuning and has to make pose predictions on the real image, such as those shown in Figure 4.4 (a).

A relevant question is whether different model renderings and processing the real images to make them similar to the synthetic images will increase the pose estimation accuracy. To test this, the synthetic training images and the real test images are transformed to edge gradient magnitude (*gradmag*) images. Converting images to edge *gradmag* comes at the cost of loss of information such as colour and texture, but on the other hand, the main geometrical features of the images are preserved. The effect of pre-processing the synthetic and real images is evaluated on the performance of the pose regression network.

4.3 Experiments and results

Three experiments were conducted to evaluate the pose regression ability of a network fine-tuned with synthetic images on real images:

- Experiment 1: This experiment consisted of creating a baseline accuracy using real images for comparing the results with our experiments. Additionally, the optimum value of β for the scene was determined that is subsequently used as a tuning parameter for the next experiments and is explained in Section 4.3.2.
- Experiment 2: This experiment consisted of two sub-experiments to evaluate different renderings for the generation of synthetic images. In the first sub-experiment (Section 4.3.3), five networks with synthetic datasets are fine-tuned and tested with the respective synthetic datasets. In the second sub-experiment (Section 4.3.3), five networks are fine-tuned with synthetic image datasets, and their transfer learning ability was assessed by testing with real images directly.
- Experiment 3: In the last experiment, the effect of the level-of-detail of the 3D building model on the accuracy of the localisation was explored in Section 4.3.4.

The networks were implemented using Caffe (Jia et al., 2014) library on Linux. The fine-tuning loss was minimised using the Adagrad gradient descent optimisation algorithm with a fixed learning rate of 10^{-3} for 160 epochs. An NVIDIA GTX980M graphics processor unit (GPU) with 4GB of memory was used to fine-tune the networks using a batch size of 40. The average time for fine-tuning each network was approximately one-hour using NVIDIA CUDA[®] deep Neural Network library (cuDNN). The fine-tuning and testing images were resized to a resolution of 320x240 pixels, and a crop of dimension 224x224 was applied. The mean of the fine-tuning dataset was subtracted from the input images at test time.

4.3.1 Dataset used

In the current research, a 3D building model is used that was used in our previous works (Khoshelham et al., 2017; Ramezani et al., 2017; Khoshelham et al., 2018; Tran et al., 2019).

The 3D building model was derived from a BIM of the third floor of Block B of the Department of Infrastructure Engineering at the University of Melbourne, Australia. The 3D building model consists mainly of a part of the corridor having an approximate coverage of 230 m^2 . Figure 3.5 shows the plan view of the 3D building model. The BIM contains the main building elements including walls, floors, ceilings, doors, ceiling tube-lights, and stairs, but not details such as material, fabrication, assembly and installation information (such as mounting details). As per BIM specifications, the level-of-detail of the BIM used in this experiment can be considered as LoD 300 (Volk et al., 2014; Porwal and Hewage, 2013).

Synthetic image dataset

In the literature, the ground truth poses of the real images are derived from SfM approaches (Kendall et al., 2015; Clark et al., 2017; Walch et al., 2017; Kendall and Cipolla, 2016). The images are usually taken along a trajectory to cover the indoor or outdoor spaces. To generate the synthetic dataset, the same concept is imitated and a trajectory is defined in the 3D building model. The points in the trajectory serve as the ground truth poses of the synthetic images. The height of the trajectory was kept in the range of 1.5 – 1.8 metres from the floor, which is the usual height of holding a camera in hand. Artificial point light sources were placed near the actual locations of the ceiling tube-lights, to illuminate the corridor (except for edge render). The length of the trajectory was approximately 30 metres.

A virtual camera was moved along the trajectory to render four different sets of synthetic images in Blender¹, as shown in Figure 4.2. The intrinsic parameters of the virtual camera were kept identical to the real camera (Section 4.3.1). As pointed out by Wu et al. (2017), to address the challenge of an inadequate number of fine-tuning images, additional images are rendered at each point of the trajectory. In theory, synthetic images at all possible locations and orientations should be captured in the indoor space. However, this will increase the time required for image rendering and for fine-tuning the network.

¹Blender is an open-source 3D computer graphics software and finds its applications in video games, animated films, simulation and visual art. For more information please visit: www.blender.org

Table 4.1: The details of each set of synthetic data with the rendering times of each frame.

Rendering model	Resolution	Render time on CPU (sec)	Render time on GPU (sec)	No. of images rendered	Pseudonym
Cartoonish	640x480	0.59	–	2500	<i>Syn-car</i>
Photo-realistic	640x480	109.26	10.11	2500	<i>Syn-pho-real</i>
Photo-realistic textured	640x480	111.78	10.26	2500	<i>Syn-pho-real-tex</i>
Gradient magnitude of Cartoonish	640x480	0.62	–	2500	<i>Gradmag-Syn-car</i>
Edge render	640x480	0.48	–	2500	<i>Syn-edge</i>



Figure 4.2: Different sets of synthetic images rendered from the 3D building model.

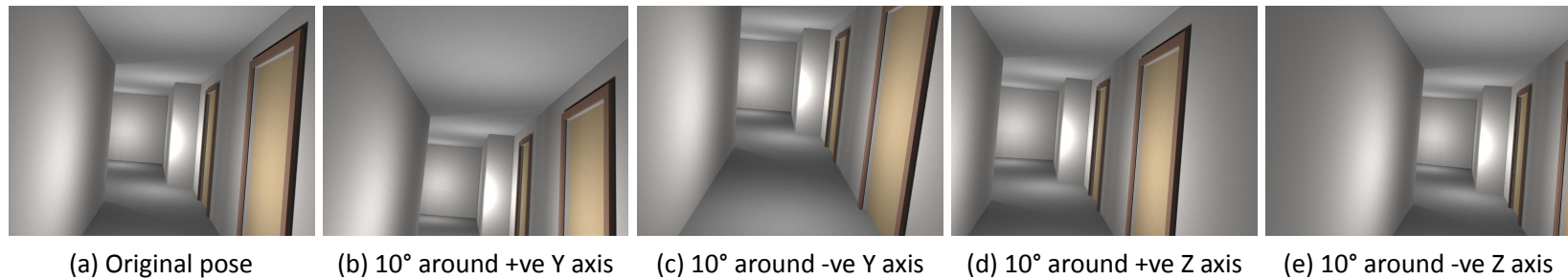


Figure 4.3: Generation of additional images on the trajectory by rotating the camera along Y and Z axes.

Therefore, images are rendered along the trajectory at 0.05 metres interval and $\pm 10^\circ$ tilt, and this has been shown in Figure 4.3. A total number of 2500 images were generated for each dataset.

As shown in Figure 4.2, the main difference between the cartoonish (*Syn-car*) and photo-realistic images (*Syn-pho-real*) is the model of rendering. Photo-realistic rendering is a physics-based model that traces the path of light rays in the scene, which is similar to the real world, whereas the cartoonish rendering roughly traces the path of light rays. The photo-realistic textured images (*Syn-pho-real-tex*) were rendered by adding repeating synthetic textures to the 3D building model, such as the textures of brick, carpet and wooden ceiling. The realism of the photo-realistic rendering comes at the cost of rendering times which is shown in Table 4.1. However, the rendering times of the photo-realistic datasets were considerably reduced with the help of a GPU. Note that the naming convention used for the datasets (e.g. cartoonish) is according to Blender terminology.

An additional dataset (*Gradmag-Syn-car*) was derived from the cartoonish images by taking the edge gradient magnitude of the images and suppressing weak edges below a threshold. The edge rendered images (*Syn-edge*) were generated by rendering only the edges of the 3D building model, without taking into account the lighting conditions. This dataset is similar to the *Gradmag-Syn-car* dataset, however, does not contain the effect of illumination of the scene, such as reflections and shadows. Table 4.1 shows the pseudonyms of the respective datasets.

Real image dataset

A smartphone camera (Motorola G 1st generation with a field-of-view (FoV) of 56.32°) was used to record a video sequence of the corridor. The intrinsic parameters of the camera were estimated by calibrating the camera. The focus of the camera was fixed to reduce any out-of-focus images. Additionally, the exposure of the camera was set to a constant to limit the underexposed images. A total number of 950 images of 640x480 pixels resolution (Figure 4.4 (a)) were acquired at a constant 30 frames per second. The edge *gradmag* images (Figure 4.4 (b)) were generated by pre-processing the real images, and suppressing weak edges that were below a threshold value.

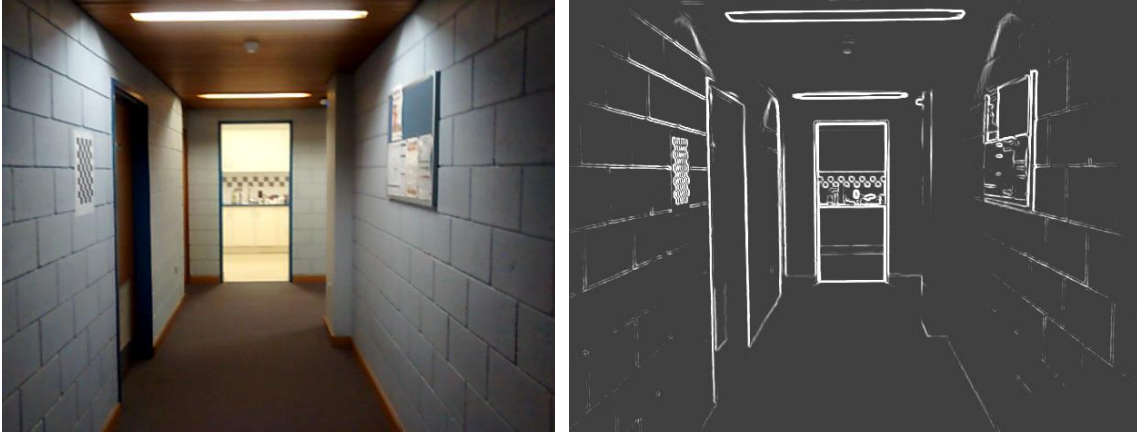


Figure 4.4: A real image (left) and its gradient intensity image (right).

4.3.2 Baseline performance using real images

To obtain a baseline performance of the network trained with real images and tested with real images, a network is fine-tuned with real images, and ground truth poses. The ground truth poses of the images sequence was estimated by using the SfM approaches of the Agisoft PhotoScan Professional[®] software. The image sequence was randomly partitioned into two halves for creating fine-tuning and test set. The coverage of the reconstructed mesh was 43 m^2 , the re-projection error was 0.91 pixels, and the total error of SfM was 2.60 centimetres.

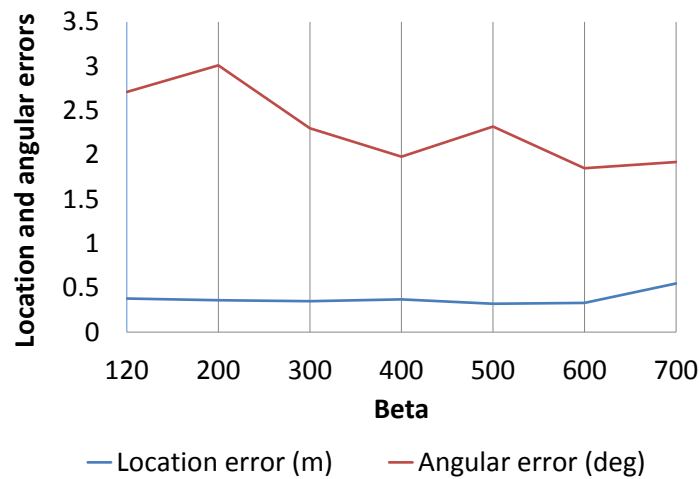


Figure 4.5: The variation of the locational and angular errors for the baseline experiment to determine the ideal value of β for the scene.

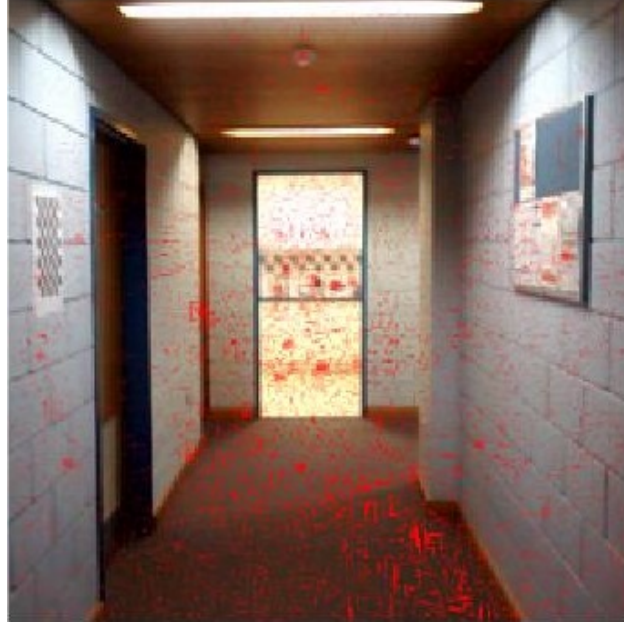


Figure 4.6: The saliency map predicted by a network fine-tuned with real images, overlaid on the test image. The red colour indicated the pixels in the image that the network considers important for pose regression.

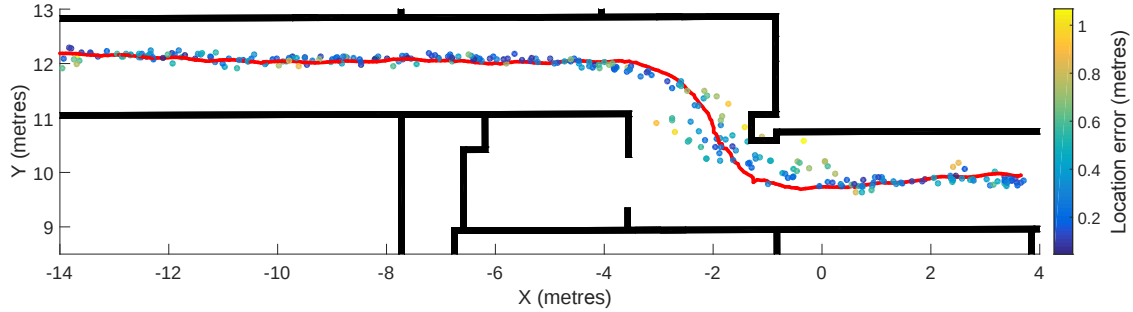


Figure 4.7: Validation results for the baseline experiment in which the network was trained with real images and tested with real images. The red line represents the ground truth trajectory estimated by the SfM approaches. The estimated locations by the fine-tuned network corresponding to the points in the trajectory are plotted in dark blue to yellow colours, where the colour provides a measure of error of the points.

To find a suitable value of the parameter β (introduced in Section 4.2.1), a grid search is performed in the expected range with the fine-tuning set. The optimum value of β is the ratio between the expected error of location and orientation at the end of fine-tuning. Hence for indoor scenes where the location errors are small, the value of β lies in the

range of 120 to 750 (Kendall et al., 2015). From Figure 4.5, a β value of 600 is selected, after which there is a increase in the location errors.

Figure 4.6 shows the saliency map of a real image as predicted by the fine-tuned network. The saliency maps (Simonyan et al., 2013) are an indicator of the sensitivity of image pixels or parts of an image that the network considers vital for the task of localisation. It is observed that the network considers the floor, roof and edges corresponding to wall-floor intersection important for the pose regression task. Figure 4.7 shows the distribution of the estimated points by the network fine-tuned with a β value of 600. The median of validation error is 0.33 metres and 1.85° and is summarised in Table 4.2. It is observed that the largest localisation errors can be related to part of the trajectory that has a turn. It could be the effect of motion blur (explained in Section 4.3.3), that degrades the quality of the images.

Table 4.2: The test errors of the fine-tuned networks on real images (training dataset with 475 images) and tested with real images (test dataset with 475 images).

Fine-tuned on	Validated with	Error (metres, degrees)
Real images	Real images	0.33 m, 1.85°

4.3.3 Fine-tuning with synthetic images

To evaluate the performance of the network when trained with synthetic images, two experiments were performed using five networks that were fine-tuned on the synthetic datasets, namely *Syn-car*, *Syn-pho-real*, *Syn-pho-real-tex*, *Gradmag-Syn-car* and *Syn-edge*. The value of β was kept at a constant value of 600 for the five networks.

Localisation accuracy for synthetic images

In this experiment, the five synthetic datasets were randomly partitioned into two halves (Table 4.3) for creating fine-tuning and validation sets. Subsequently, the fine-tuning errors of the networks were determined by testing the five networks with their validation sets. Figure 4.8 shows the estimated locations of the synthetic images by the five fine-tuned networks and Table 4.4 summarises the validation errors.

Table 4.3: The number of synthetic images used for fine-tuning the five networks to validate with synthetic images.

Total synthetic images	Fine-tuning images	Validation images
2500	1250	1250

Table 4.4: The fine-tuning errors for the five networks with respect to validation datasets.

Fine-tuned on	Validated with	Fine-tuning error (metres, degrees)	Shown in Figure 4.8
<i>Syn-car</i>	<i>Syn-car</i>	0.37m, 2.12°	(a)
<i>Syn-pho-real</i>	<i>Syn-pho-real</i>	0.33m, 1.83°	(b)
<i>Syn-pho-real-tex</i>	<i>Syn-pho-real-tex</i>	0.31m, 1.71°	(c)
<i>Gradmag-Syn-car</i>	<i>Gradmag-Syn-car</i>	0.40m, 2.37°	(d)
<i>Syn-edge</i>	<i>Syn-edge</i>	0.35m, 2.88°	(e)

From Figure 4.8 it is observed that the distribution of the estimated locations follows the same trend observed in the baseline experiment (Figure 4.7), for example, the largest error can be backtracked to the part of the trajectory that has a turn. It is noticed that the maximum localisation errors for a few points on the trajectory for the synthetic datasets are higher (approximately three metres) as compared to the baseline experiment (approximately one metre). However, from Table 4.4 it is observed that the validation errors of the five fine-tuned networks on synthetic data are competitive with the baseline condition. This experiment provides an insight into the ability of the network to adapt to diverse fine-tuning data.

It is interesting to note that the networks fine-tuned with *Gradmag-Syn-car* and *Syn-edge*, perform similarly to the network fine-tuned on the real images (baseline experiment). This fact indicates that a network fine-tuned with edge images (or edge *gradmag*) of a scene, which does not contain any texture or colour information, is sufficient for the task of pose regression.

Localisation accuracy for real images

In this experiment, the five networks fine-tuned with synthetic data were tested with the two sets of real images (Table 4.6) to estimate the cross-domain localisation errors by leveraging transfer learning. Wu et al. (2017) state that the fine-tuned networks exhibit

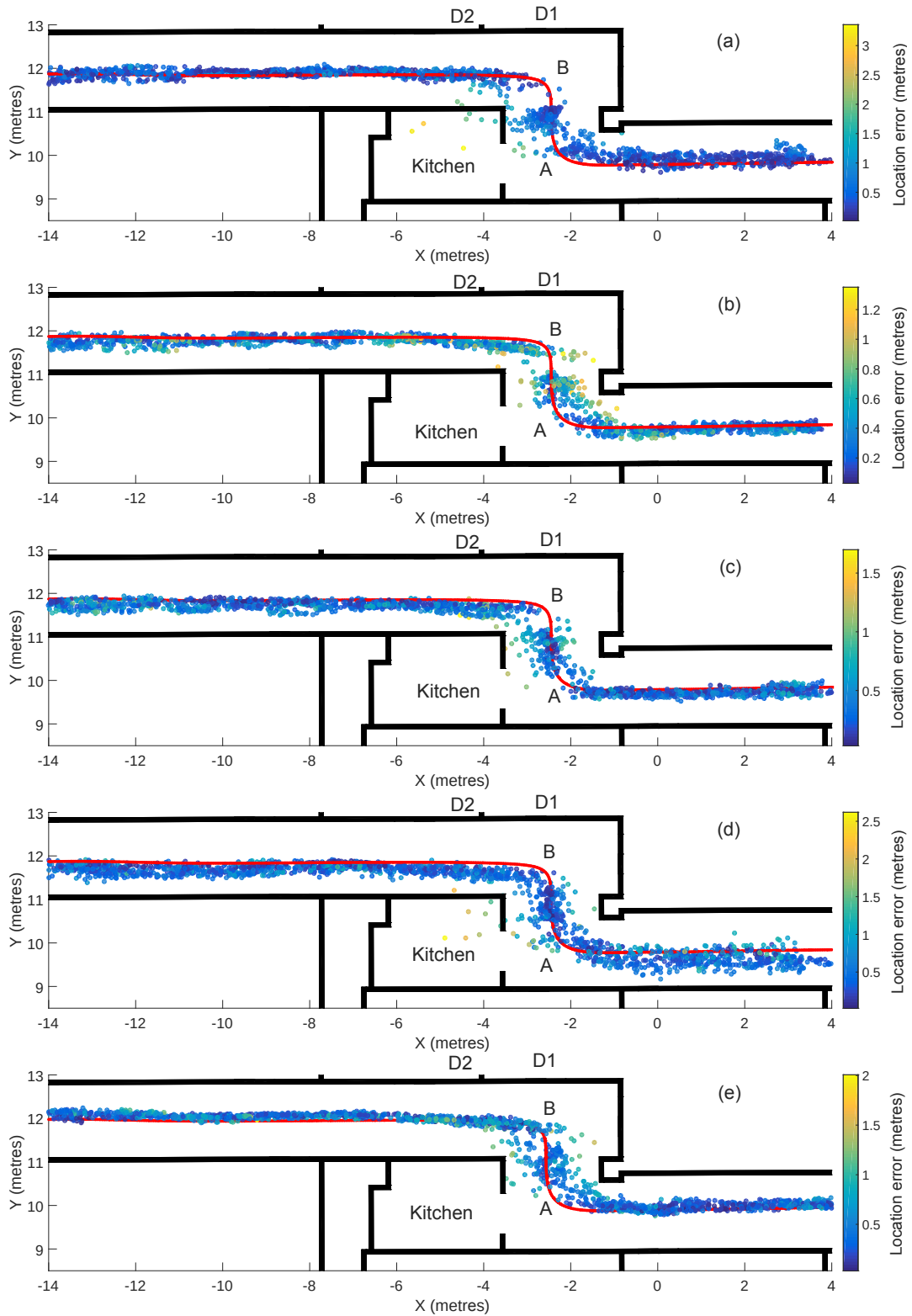


Figure 4.8: The validation results for the networks (a) fine-tuned with *Syn-car* and tested on *Syn-car* (b) fine-tuned with *Syn-pho-real* and tested on *Syn-pho-real* (c) fine-tuned with *Syn-pho-real-tex* and tested on *Syn-pho-real-tex* (d) fine-tuned with *Gradmag-Syn-car* and tested on *Gradmag-Syn-car* (e) fine-tuned with *Syn-edge datasets* and tested on *Syn-edge datasets*. D1 and D2 represent the locations of two doors in the corridor.

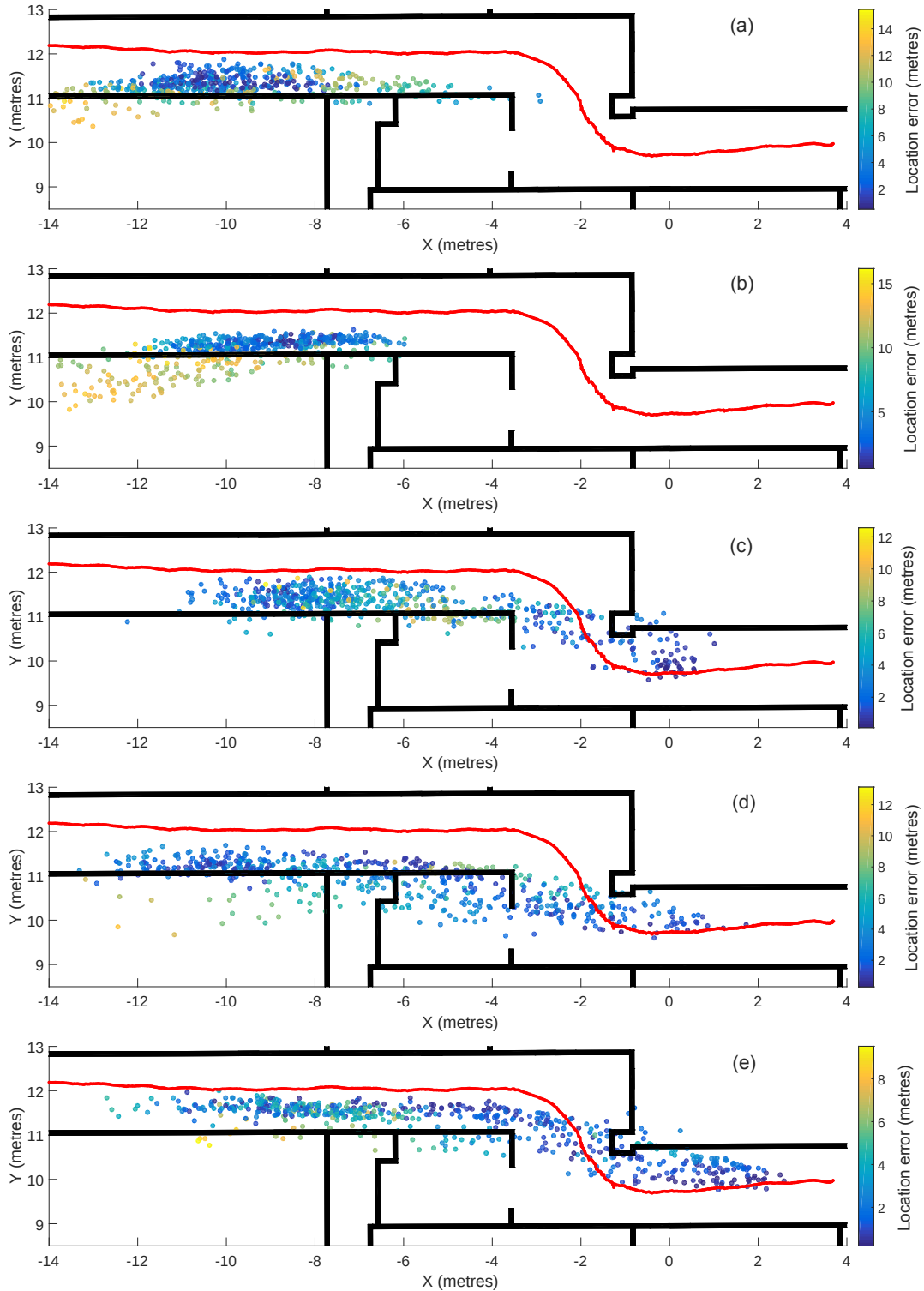


Figure 4.9: Estimated locations by the network (a) fine-tuned with *Syn-car* and tested on real images (b) fine-tuned with *Syn-pho-real* and tested on real images (c) fine-tuned with *Syn-pho-real-tex* and tested on real images (d) fine-tuned with *Gradmag-Syn-car* and tested on *gradmag* of real images (e) fine-tuned with *Syn-edge* and tested on edge *gradmag* of real images.

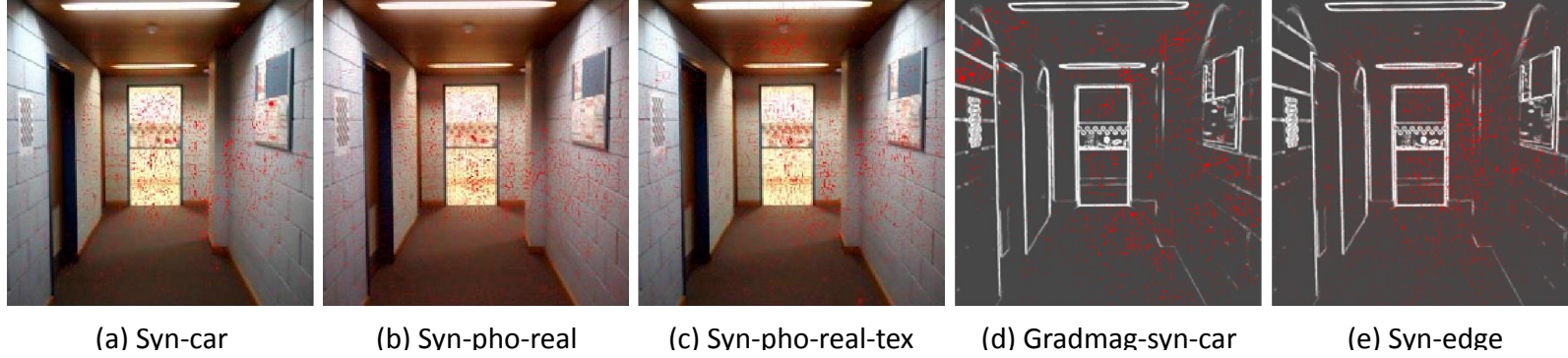


Figure 4.10: The saliency maps of (a) - (c) a real image and (d) - (e) edge *gradmag* of real image predicted by networks fine-tuned with synthetic image datasets. The red colour shows the pixels in the image the network considers important for pose regression.

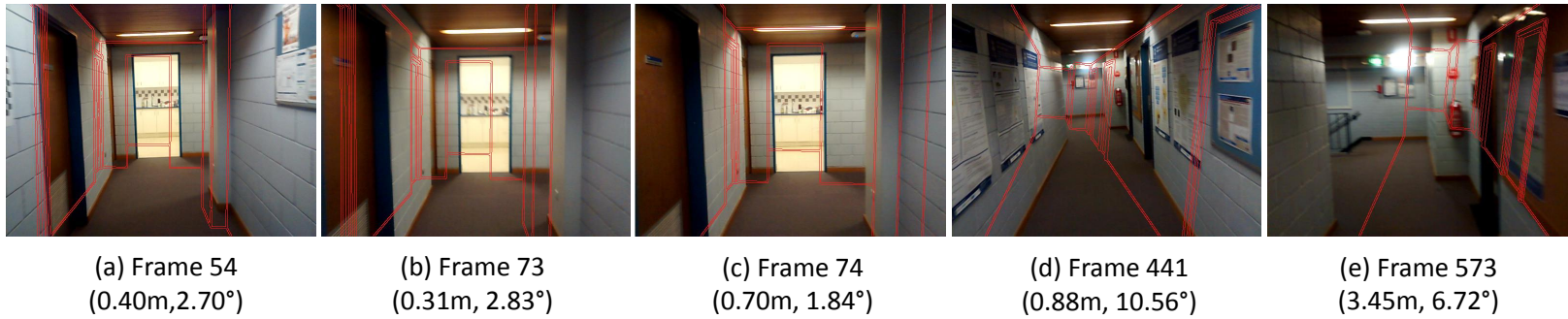


Figure 4.11: Overlay of the predicted pose on real images by the network fine-tuned on *Syn-edge* dataset. The values in the bracket represent the errors.

Table 4.5: The test errors of the three networks on real images (third column) and the test errors on the edge *gradmag* of the real images (fourth column).

Fine-tuned with	Validated with	Error for real images (metres, degrees)	Shown in Figure 4.9
<i>Syn-car</i>	Real images	6.25 m, 37.16°	(a)
<i>Syn-pho-real</i>	Real images	5.99 m, 11.33°	(b)
<i>Syn-pho-real-tex</i>	Real images	3.06 m, 12.25°	(c)
<i>Gradmag-Syn-car</i>	Gradmag of real images	2.63 m, 6.99°	(d)
<i>Syn-edge</i>	Gradmag of real images	1.88 m, 7.73°	(e)

a good transfer learning ability for synthetic images. They derive synthetic images from a 3D reconstruction of the scene from the SfM approaches that contain real textures of the environment. However, this is different in our case, where any texture information is not used (*Syn-car*, *Syn-pho-real*, *Gradmag-Syn-car* and *Syn-edge*) or repetitive synthetic textures are used to populate the scene (*Syn-pho-real-tex*).

Table 4.6: The number of synthetic images used to fine-tune the five networks for testing with real images/ gradmag of real images

Synthetic images used for fine-tuning	Real/ Real gradmag images for testing
2500	950

As the three synthetic datasets namely the *Syn-car*, *Syn-pho-real* and *Syn-pho-real-tex* are similar in appearance to the real images, the three networks are tested with real images directly. However, for the *Gradmag-Syn-car* and *Syn-edge* datasets, the edge *gradmag* of the real images are used for testing. Figure 4.9 shows the predicted locations of the real images, tested with networks fine-tuned on synthetic datasets and Table 4.5 shows the errors of the localisation.

From Figure 4.9 (a) - (b) and Table 4.5 it can be seen that the performance of the networks fine-tuned with the two synthetic datasets (*Syn-car* and *Syn-pho-real*) for localising with real images are poor. However, for the network fine-tuned with *Syn-pho-real-tex* dataset (Figure 4.9 (c)), the results are better, and the reason might be the high realism of the images as compared to the two other datasets. Additionally, the distribution of the estimated locations is uneven, and clusters are formed near one part of the trajectory, which indicates the presence of bias (or a shift).

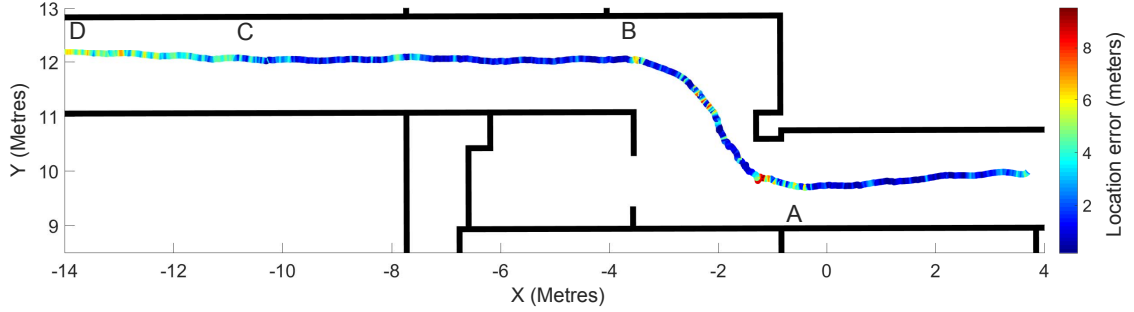


Figure 4.12: Visualising error for the different parts of the trajectory. The colour represents the magnitude of the errors. The maximum error is present (red colour) in the part AB of the trajectory. In the part CD of the trajectory, the errors are consistently larger compared to other parts of the trajectory.

Figure 4.9 (d) - (e) shows the estimated locations of the edge *gradmag* of the real images by networks fine-tuned with *Gradmag-Syn-car* and *Syn-edge* datasets. It is observed that there is an improvement in the distribution of the estimated locations. From Table 4.5 it is observed that the test errors reduce as compared to testing with the real images directly. The test errors of the network fine-tuned with the *Syn-edge* dataset is the least. Although the bias is reduced by using the edge *gradmag* images for fine-tuning and testing, it is not eliminated.

Figure 4.10 (d) - (e) shows the saliency maps of the networks fine-tuned on *Gradmag-Syn-car* and *Syn-edge* datasets respectively. It can be seen that both the networks consider patches of floor and roof important for the task of localisation, however, there are two differences. Firstly, the network fine-tuned with *Gradmag-Syn-car* dataset additionally considers patches of brick textures (top left part and right part of the image) important. As brick textures are repeating structures throughout the indoor spaces and using them as features for localisation causes an ambiguity. This fact can explain the poor performance as compared to the network trained with the *Syn-edge* dataset. Lastly, the network fine-tuned on *Syn-edge* dataset considers the edges corresponding to the pillar (on the right) more important. It is, however, interesting to note that both the networks do not consider ceiling tube-lights as important features for localisation.

Figure 4.11 shows the predicted pose of the camera by the network fine-tuned with the *Syn-edge* dataset, by overlaying the 3D building model on the real images. On a

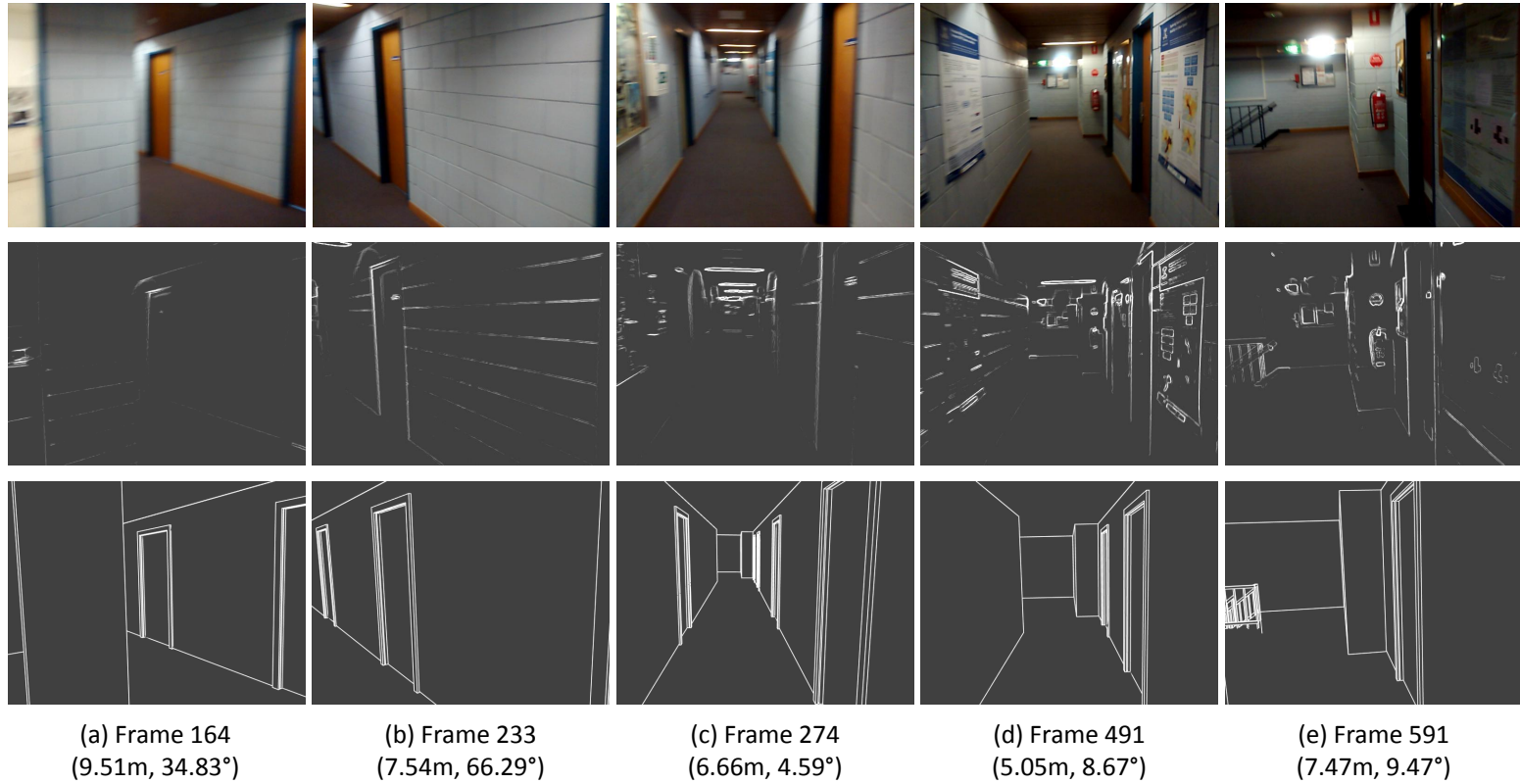


Figure 4.13: Visualising frames with high errors. The first row of images shows the real images of the frames. The second row shows the respective edge *gradmag* of the images and the third row shows the images of the *Syn-edge* dataset with the same pose of the virtual camera. Frame 164, 233 and 274 shows the effects of motion blur, which results in low values of edge *gradmag* for the blurry areas. Frame 491 and 591 shows the effects of artefacts in the edge *gradmag* of real images, due to structural differences, such as the presence of posters, notice boards and fire-extinguisher. Errors are shown in brackets.



Figure 4.14: Generation of a low level-of-detail 3D building model (Right) from the 3D building model derived from BIM (Left) by simplifying geometries. The colours are added to the roof (green), walls (white), doors (orange) and floor (gray) for better visualisation.

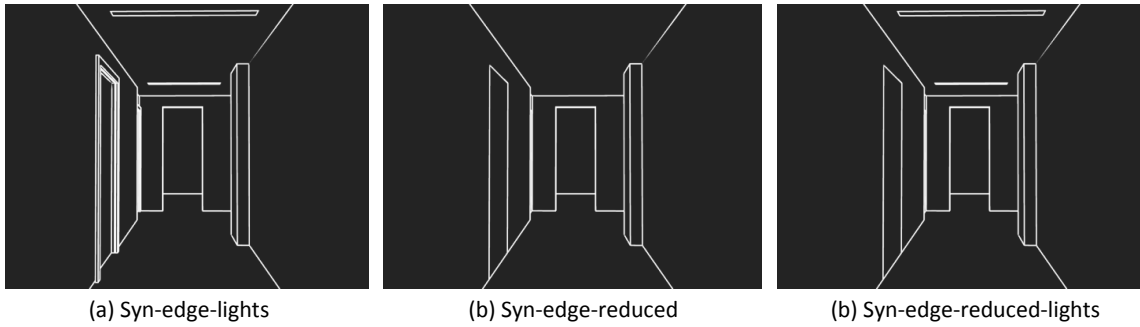


Figure 4.15: Different sets of synthetic images rendered to evaluate the effects of level-of-detail of a 3D building model on the errors of localisation.

Table 4.7: The test errors the three fine-tuned networks on edge *gradmag* of real images to determine the effects of level-of-detail of the 3D building model on their pose regression abilities.

Fine-tuned with	Validated on	Error for gradmag of real images (metres, degrees)	Shown in Figure 17
<i>Syn-edge-lights</i>	Gradmag of real images	2.44m, 7.80°	(a)
<i>Syn-edge-reduced</i>	Gradmag of real images	2.32m, 8.16°	(b)
<i>Syn-edge-reduced-lights</i>	Gradmag of real images	2.46m, 8.89°	(c)

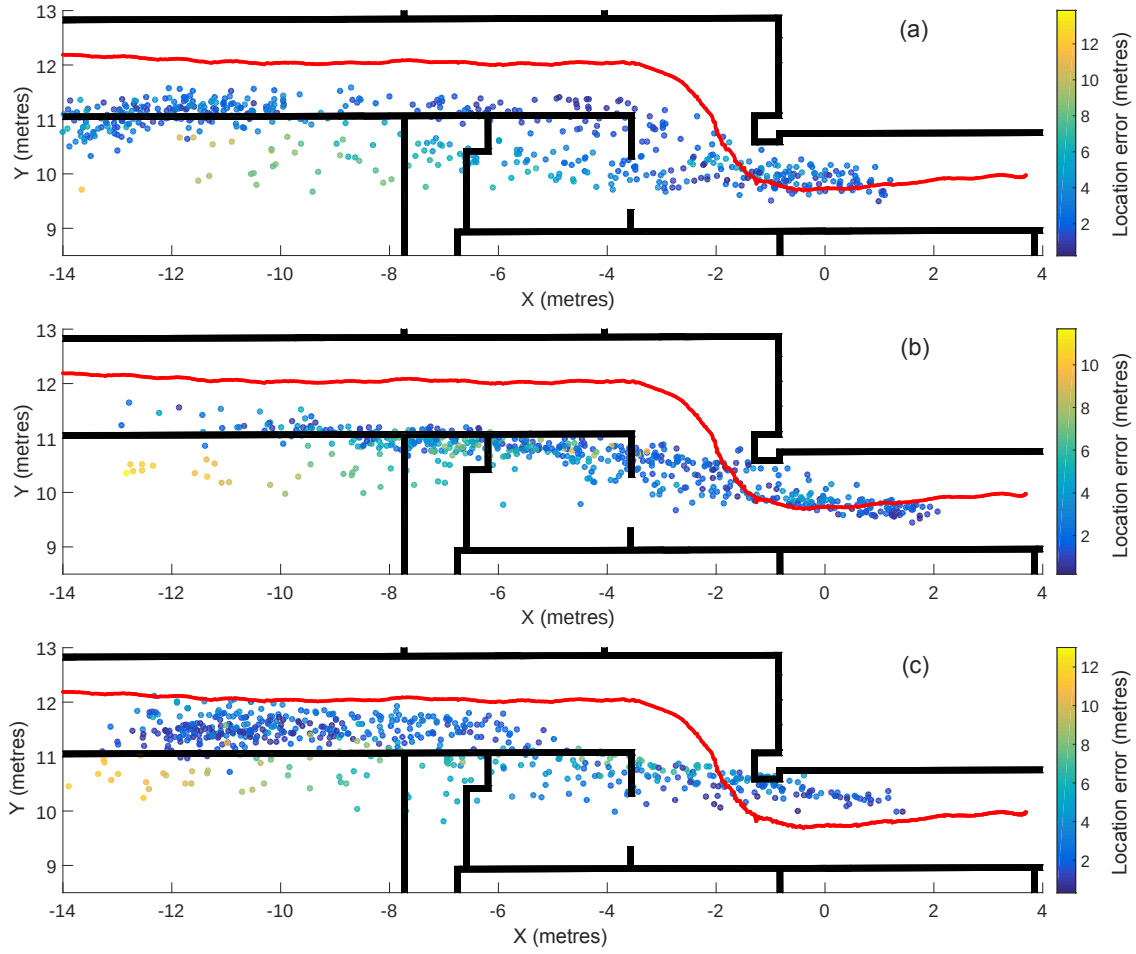


Figure 4.16: The results of the estimated locations of real images and edge *gradmag* of real images on fine-tuned networks on (a) *Syn-edge-lights* (b) *Syn-edge-reduced* and (c) *Syn-edge-reduced-lights*

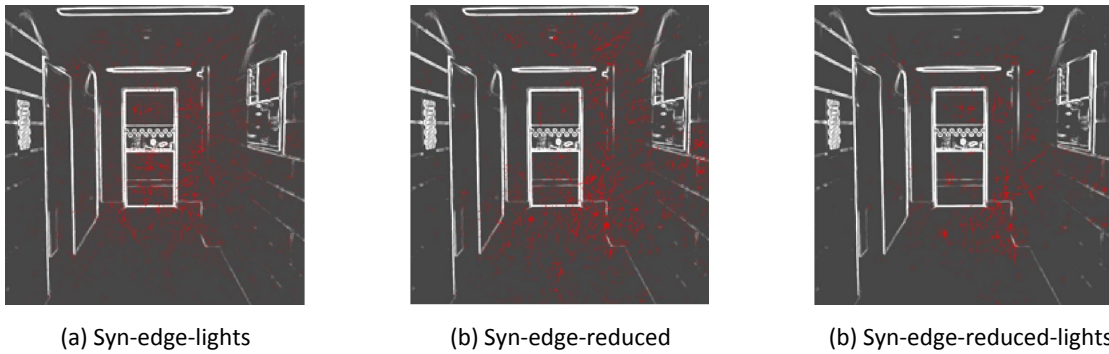


Figure 4.17: The saliency maps of the edge *gradmag* of real images as predicted by the networks fine-tuned with (a) *Syn-edge-lights* (b) *Syn-edge-reduced* (c) *Syn-edge-reduced-lights* datasets. The red colour represents the image pixels that the network considers important for the task of pose regression.

closer look at the distribution of the errors along the trajectory, as shown in Figure 4.12, it is observed that the maximum errors (up to 9.5 metres) are present in the part AB of the trajectory which is a turn. Additionally, it is noticed that for the part CD of the trajectory, the errors are consistently higher (in the order of three to eight metres), as compared to the other parts of the trajectory.

Examining images with large localisation errors reveals that these images contained a significant amount of motion blur or artefacts in the real image. Motion blur in the real images result in low values of edge *gradmag* in the test images which results in information loss (Figure 4.13 (a) - (c)), hence explains the large errors for the part AB of the trajectory.

Artefacts are introduced in the edge *gradmag* of the real images due to the presence of external elements that are not present in the 3D building model, such as posters and notice boards. Additionally, due to the structural difference between the real scene and the 3D building model, some artefacts are introduced in the real images. These structural differences are a result of errors in modelling the indoor spaces. Lighting of the scene plays a vital role in the appearance of the scene, and this adds more artefacts in the real images, such as the top left of Figure 4.13 (d). These artefacts result in large location errors for the part CD of the trajectory. The large errors can be accounted for the occlusions by the artefacts, which cover the parts of the images that the network considers important for localisation.

Differently, the high errors might be a result of the learnt features for each network, which might not be suitable for pose regression with real images. This fact is reflected in the saliency maps of the real images as predicted by the fine-tuned networks (Figure 4.10 (a) - (c)), which considers other parts of the image more important as compared to the saliency map for the baseline experiment (Figure 4.6). However, the closest saliency map with the baseline experiment is for the network fine-tuned with *Syn-pho-real-tex* dataset, where it considers parts of the floor, roof and side walls are important for the pose regression. This fact additionally explains the better results of the *Syn-pho-real-tex* dataset as compared to the other two synthetic datasets.

4.3.4 Effects of level-of-detail of 3D building models on localisation

So far the experiments were conducted using a single 3D building model, without considering the effects of the level-of-detail of the models. In practice, 3D building models are generated at different levels of detail. For example, a 3D model obtained from a 2D floor plan by extruding the walls will have a very low level-of-detail. Therefore, a door might just be represented by a rectangle. Hence, to assess the effects of the level-of-detail on the pose regression ability of the fine-tuned networks, three 3D building models were derived from the original model.

The first dataset called *Syn-edge-lights* was created by adding ceiling tube-lights to the original 3D building model, keeping all other building elements identical to the model used to render *Syn-edge* dataset. As the saliency map shown in Figure 4.10 (e) indicate that the network trained on *Syn-edge* do not consider the ceiling tube-lights important for localisation, and therefore, this dataset is created to assess the effects of ceiling tube-lights on the localisation performance of the network.

The second 3D building model was derived from the original model to create a low level-of-detail 3D building model, using which the *Syn-edge-reduced* dataset was created. The low level-of-detail 3D building model was created by retaining only the geometries of the main building elements, such as walls, floors, doors, ceiling and ceiling tube-lights. The geometry of the wall-door intersection was kept, and other details, such as the frame of the door was simplified. Figure 4.14 shows the same view of a virtual camera for the 3D building model derived from the BIM and the low level-of-detail 3D building model generated from it.

The third 3D building model was created by adding ceiling tube-lights to the low level-of-detail model, using which the *Syn-edge-reduced-lights* dataset was created. This dataset was generated to simulate the effects of the presence of the ceiling tube-lights in the low level-of-detail 3D building model, in case the auxiliary information is available. Sample synthetic images of the three datasets are shown in Figure 4.15. As per BIM specifications the low level-of-detail 3D building models used in these experiments can be considered to be derived from a LoD 200 BIM (Volk et al., 2014; Porwal and Hewage, 2013).

Subsequently, three networks were fine-tuned with the three modified datasets and were tested with the edge *gradmag* of the real images. Table 4.7 suggests that there is an increase in the localisation error for all the three networks compared to the network fine-tuned on *Syn-edge* dataset. In addition, Figure 4.16 shows that the distribution of the estimated points is worse as compared to the network fine-tuned with the *Syn-edge* dataset (Figure 4.9 (e)). Figure 4.17 shows the saliency maps of the three fine-tuned networks.

It is surprising that the networks that contained the ceiling tube-lights in the fine-tuning datasets (*Syn-edge-lights* and *Syn-edge-reduced-lights*) do not consider the ceiling tube-lights important. Additionally, the two networks consider roof as less important compared to the other network trained on the *Syn-edge-reduced* dataset. The saliency map of the network fine-tuned with the *Syn-edge-reduced* dataset is similar to the saliency map of the network fine-tuned on the *Gradmag-Syn-car* dataset.

The results indicate that the presence of the ceiling tube-lights in the fine-tuning dataset does not improve the localisation performance. Additionally, comparing errors from Table 4.2 and Table 4.7 (1.88 vs 2.32 metres) it can be concluded that a high level-of-detail of a 3D building model is desirable.

4.3.5 Discussion and future directions

Due to the constraint nature of indoor spaces, there are several instances where two places might appear similar due to symmetry or repetitiveness. Consider the case of a 90° turn in a corridor, especially if there are not enough distinguishing features in the FoV of the camera. The problem is referred to as *perceptual aliasing* (Lowry et al., 2016), and currently, is a limitation of the proposed approach. Figure 4.18 shows an example of perceptual aliasing. Figure 4.8 shows the images that are associated near a 90° turn (near point B) have the highest errors. In contrast, images near point A has some unique features (kitchen door without a door frame) in the FoV of the camera, hence explaining low error compared to point B.

In the literature, to address the problem of perceptual aliasing, a sequence of images (or video) is used by exploiting the spatio-temporal information of the consecutive frames

such as the works of [Walch et al. \(2017\)](#), [Clark et al. \(2017\)](#) and [Valada et al. \(2018a\)](#). In addition, Bayesian CNN have been used to model the uncertainty of the estimated location by the networks ([Kendall and Cipolla, 2016](#)), which can be used to measure the confidence of the location estimations by the network.



Figure 4.18: Left: Door D1 and Right: Door D2. An example of perceptual aliasing resulting in large errors in two test images containing two different doors whose appearance is identical.

The proposed approach has a potential to replace the existing infrastructure-dependent indoor localisation approaches, considering the easy availability of the 3D building models. However, the only limiting factor is the accuracy of localisation for applications. An accuracy of two metres is suitable for indoor localisation applications such as guiding personnel to a general location ([Taneja et al., 2011](#)) and is suitable for pedestrian navigation. However, this accuracy is not sufficient for robot navigation applications such as unmanned aerial vehicles or unmanned ground vehicles, since the width of a corridor or the dimensions of a room could be less than the error of localisation by our approach.

It is evident from Section 4.3.4 that there is a significant decrease in localisation performance when using a low level-of-detail 3D building model (1.88 vs 2.32 metres). However, this accuracy in localisation is still suitable for pedestrian navigation. Thus, the proposed approach can *potentially* work with 3D models derived from 2D floor plans, even without the knowledge of the ceiling tube-lights, in the case of unavailability of a BIM. The verification of the same will be a part of a future work.

In contrast to approaches that use a central server for processing data and infrastructure for communicating with the users, such the approach of [Kumar et al. \(2016\)](#), the

proposed approach is independent of a processing server or any infrastructure to communicate with it. The processing time for each image for the proposed approach is 5 milliseconds during test time on a GPU or 0.625 seconds on a CPU (i7 @ 2.6 GHz). The GPUs on the modern ubiquitous mobile devices can be utilised for real-time processing. Alternatively, the real-time localisation can be performed at lower update rates on CPU alone. Another interesting direction for future research is to explore miniature networks that are suitable for mobile platforms such as [Müller et al. \(2017\)](#) or [Howard et al. \(2017\)](#), with less computational expenses.

There is still a room for improvement in the current approach which is left as a part of future studies. Firstly, the semantic knowledge derived from a BIM can be incorporated into the approach, such as the work of [Valada et al. \(2018b\)](#), where the real test images can be segmented at the test phase. Secondly, constraints from the 3D building models can be used during fine-tuning of the networks, so that the estimated locations of the camera is always inside the navigable area of the indoor space. Lastly, the fine-tuned networks could be combined with model-based visual tracking approaches to increase the accuracy and the robustness for performing indoor localisation.

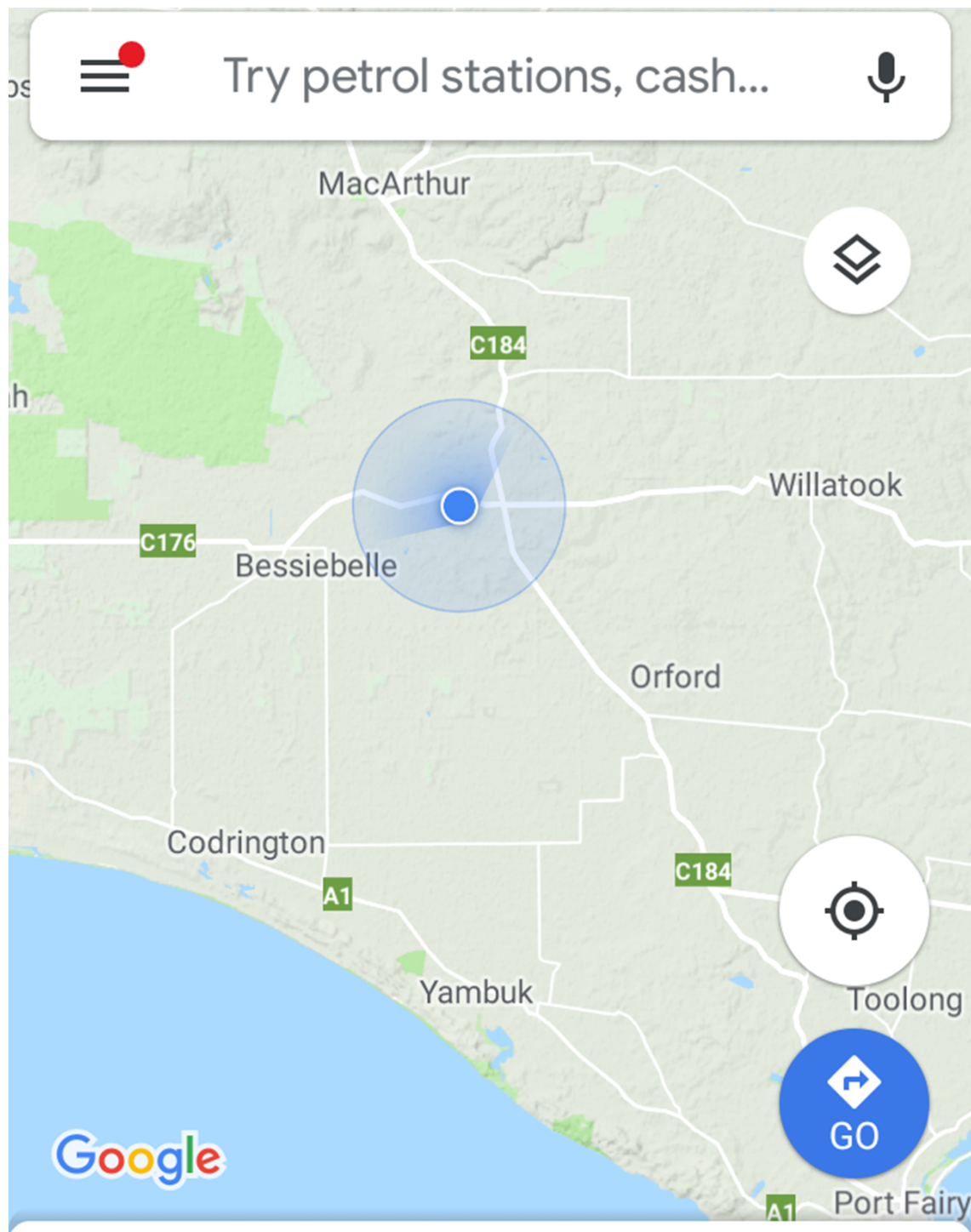
4.4 Conclusions

In this chapter, a deep CNN pose regressor trained by synthetic images is proposed that solves the problem of indoor localisation. The presented approach does not require any additional infrastructure nor a point cloud, 3D image-based reconstructed models, image database or depth of the environment. Synthetic images derived from a 3D building model has been used to fine-tune a deep CNN for camera pose regression in an indoor space.

The validation results of the networks fine-tuned with synthetic images suggest that the networks retain the camera pose information accurately even for the *gradmag* images that contain no colour or texture, but only structural edges. The main finding of the research is that the deep CNN regressor fine-tuned by synthetic edge images can successfully estimate the pose of real images represented as edge gradient magnitude, achieving

a localisation accuracy of approximately two metres. A higher level-of-detail 3D building model is desirable for better results, but the proposed approach can work with low level-of-detail 3D building models as well.

The localisation errors are mainly caused by the motion blur, the structural differences of the 3D building model and the real indoor space, and the presence of artefacts in the real image. Alternatively, the errors could be an effect of the features learnt by the network for localisation. Perceptual aliasing due to symmetry and repetitiveness of indoor spaces remains a challenge for this approach. The outcomes of the present study suggest the approach has the potential to provide initial locations in real-time for the infrastructure-independent localisation approaches such as pedestrian dead reckoning (PDR), simultaneous localisation and mapping (SLAM) and visual odometry.



"... the uncertainty plays an important role in localisation ..."

≈ Bessie Belle, Victoria, Australia
(Courtesy: Google maps mobile)

Chapter 5

Bayesian BIM-PoseNet: Modelling localisation uncertainty of BIM-PoseNet

In this chapter, Bayesian BIM-PoseNet is presented to model the uncertainty of camera pose estimates by BIM-PoseNet. A deep Bayesian convolutional neural network is fine-tuned with synthetic images generated from the 3D building model to estimate the camera pose of real images. The uncertainty of the estimated camera poses is modelled by sampling the outputs of the Bayesian network fine-tuned with synthetic images. The results of the experiments indicate that a localisation accuracy of 2 metres can be achieved using the proposed approach. The modelled uncertainties are correlated with the actual errors of the pose estimates, thus providing confidence with which the camera pose estimates can be trusted.

5.1 Introduction

The discussions in the previous chapter revealed that one of the limitations of BIM-PoseNet that restricts wide application is its poor accuracy, which is approximately 2 metres, and the maximum errors were in the order of 10 metres. Uncertainty of localisation provides an indication of confidence and trust in estimated locations in the absence of ground truth. As discussed in Section 2.3.3, uncertainty modelling in pose regression networks has been done based on real images by previous workers. The results of those studies show a correlation between the errors and the modelled uncertainties. Therefore, the uncertainty of camera pose regression of real images is modelled with a Bayesian neural network (Kendall and Cipolla, 2016) using synthetic images generated from a 3D building model. This approach, called Bayesian BIM-PoseNet eliminates the

requirement of real images with their ground truth poses during the fine-tuning phase (similar to BIM-PoseNet). The results are compared with a benchmark that is created by using real images and structure-from-motion (SfM) approaches. The following are the main contributions:

1. The uncertainty of the estimated camera poses from a network fine-tuned with synthetic images is modelled to show the correlation with actual errors.
2. The proposed approach eliminates the requirement of performing image-based reconstruction of indoor spaces by utilising synthetic images rendered from a 3D building model.
3. A detailed analysis of the trajectory is performed to identify the factors contributing to large errors and uncertainties.

5.2 Methodology

Bayesian BIM-PoseNet uses a pre-trained Bayesian convolutional neural network (CNN) fine-tuned with different types of synthetic images with known camera poses to estimate the unknown camera poses of real images. Figure 5.1 shows the overview of the approach, where a modified architecture of GoogLeNet (Szegedy et al., 2015) is used. The network is fine-tuned with a synthetic dataset containing several types of images generated by graphical rendering from the 3D building model. Subsequently, the fine-tuned networks are tested with real images for its pose regression ability and uncertainty of the network is modelled to check the correlation with actual errors. Besides, the factors contributing large errors and uncertainties are identified for networks that are fine-tuned with synthetic images and tested on real images.

5.2.1 Rendering synthetic images from a 3D building model

In the literature, the ground truth camera poses are estimated using SfM approaches for fine-tuning the networks (Kendall et al., 2015; Walch et al., 2017; Clark et al., 2017; Kendall and Cipolla, 2016). In this research, the ground truth camera poses are known,

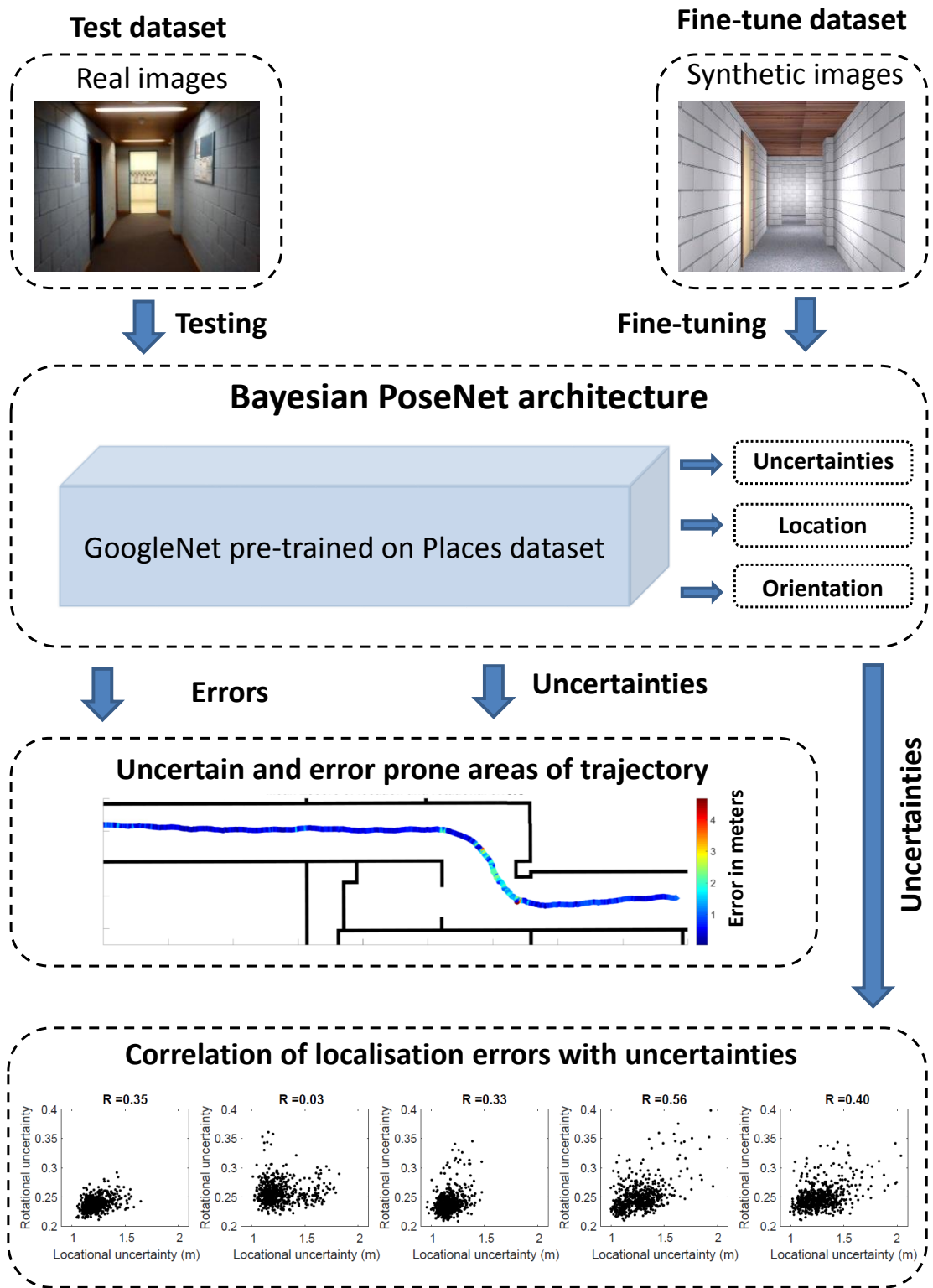


Figure 5.1: The design of Bayesian BIM-PoseNet.

and a virtual camera is used to render synthetic images. The details of the creation of the synthetic dataset are described in Section 4.3.1.

A pre-trained deep CNN requires only a few fine-tuning images as compared to millions of images required for training from random weights. As generating millions of synthetic images is an overwhelming task, a pre-trained network is fine-tuned with 2500 synthetic images and perform pose regression by leveraging transfer learning. A relevant question is whether the fine-tuned network can learn relevant high-level geometrical features from the synthetic images for camera pose regression with real images. The relevant high-level features in the context of indoor spaces could be doors, walls and ceilings.

Additionally, another question is whether processing the real images to make them similar to the synthetic images will improve the pose regression ability of the network. To examine the concept, the real and the synthetic images are converted to edge gradient magnitude (gradmag) images, hence transforming them both to a common image feature space. Consequently, the networks are fine-tuned with edge gradmag of synthetic images and test with edge gradmag of real images.

5.2.2 Fine-tuning a Bayesian network

Bayesian neural networks model the uncertainty in neural networks by gathering distributions over the network weights (MacKay, 1992). Drop-out (Srivastava et al., 2014) is a common practice that is used to avoid over-fitting a network while training with limited training data. Sampling a Bayesian network with randomly dropped out connection at test time can be considered as a way of getting Monte Carlo samples from the posterior distribution of the estimations (Kendall and Cipolla, 2016). Moreover, as the posterior distribution is not directly traceable for such a network, a variational inference usually is used to approximate the distribution of the weights (Gal and Ghahramani, 2016a).

A 24 layer deep Bayesian CNN (Kendall and Cipolla, 2016) is adopted, where the softmax classifier (Szegedy et al., 2015) is replaced with a fully connected layer of dimension 2048. The fully connected layer is connected to a 7-dimensional affine pose regressor that regresses the camera pose p :

$$p = [x, q] \quad (5.1)$$

where x is the location of camera $[X, Y, Z]$, q is the quaternions defining camera rotation $[q_0, q_1, q_2, q_3]$

The weights are taken from a pre-trained network trained on Places dataset (Zhou et al., 2014). The Places dataset contains millions of images of different locations and is suitable for training networks to classify places. Using the weights of a pre-trained network eliminates the tedious task of training the network with millions of images. Drop-outs are performed only before the convolution layers that had randomly initialised weights, in order to improve the pose regression ability of the network. The network is subsequently fine-tuned with synthetic images using the objective loss function:

$$loss(I) = ||\hat{x} - x||_2 + \beta \left\| \hat{q} - \frac{q}{||q||} \right\|_2 \quad (5.2)$$

where $\hat{x}$ is the estimated location, $\hat{q}$ is the estimated orientation, I is the Image and β is the scaling factor to balance errors of x and q .

At the test time, the Bayesian network is sampled to obtain Monte Carlo drop-out samples, and the pose is estimated by taking a mean of the samples. The number of samples required for optimum performance of the network is 40, beyond which there is no evidence of improvement (Kendall and Cipolla, 2016). The distribution of the samples follows a unimodal Gaussian distribution for both location and rotation (Kendall and Cipolla, 2016), and the distributions contains only one peak corresponding to the maxima. The locational and rotational uncertainties are expressed as the trace of the unimodal Gaussian's covariance matrices and are represented by numbers. Note that the locational uncertainties have the unit of metre and the rotational uncertainties are dimensionless scalars (product of two quaternions).

5.3 Experiments and results

Two experiments were performed to model the uncertainty of the network's pose regression on real images when fine-tuned with synthetic images. The first experiment

consists of creating a baseline performance using real images, and to determine the optimum value of weighting factor β . The second experiment was performed to measure the performance of the network fine-tuned with different types of synthetic images on real images. In addition, the experiment involved fine-tuning two networks with edge gradmag of synthetic images and testing with the gradmag of the real images.

Caffe (Jia et al., 2014) deep learning library was used for the experiments. The loss was minimised with a learning rate of 10^{-3} using Adagrad gradient descent optimisation for 160 epochs on an NVIDIA GTX980M graphics processor unit (GPU) with 4GB memory. The images were resized to 320x240 pixels, and a crop of 224x224 pixels was applied subsequent to mean subtraction during fine-tuning and testing.

5.3.1 Dataset

Two datasets were used in the experiments, namely a synthetic image dataset and a real image dataset. The synthetic image dataset that was used in the experiments of BIM-PoseNet were used in the experiment of Bayesian BIM-PoseNet. The dataset contained five different type of synthetic images obtained from the 3D building model, and is shown in Figure 4.2 . The methodology and the details of the image generation from the building information modelling (BIM) is explained in details in Section 4.3.1. The real image dataset that was used in the experiments of BIM-PoseNet was used for the current experiments. The dataset consisted of images of a corridor of a building captured with a camera of a smartphone. Figure 4.4 shows a real image and its gradient intensity image and other details of the dataset is summarised in Section 4.3.1.

5.3.2 Baseline performance using real images

A baseline performance is created by fine-tuning a network with real images (not gradmag) of known camera poses and tested with real images again. The process of generation of ground truth camera poses is described in Section 4.3.2. The image sequences were randomly portioned into fine-tuning and validation sets, containing equal number of images. The suitable value of the parameter β was estimated by performing a grid

search in the expected range of 120 to 750 for indoor environments (Kendall et al., 2015). Figure 5.2 shows the median location and rotation errors with the variation of β . The location errors are calculated as the Euler distances from the ground truth and the rotational errors in radians are calculated by $2\cos^{-1}(\langle \hat{q}q \rangle)$, where $\langle . \rangle$ denotes the inner product of two vectors.

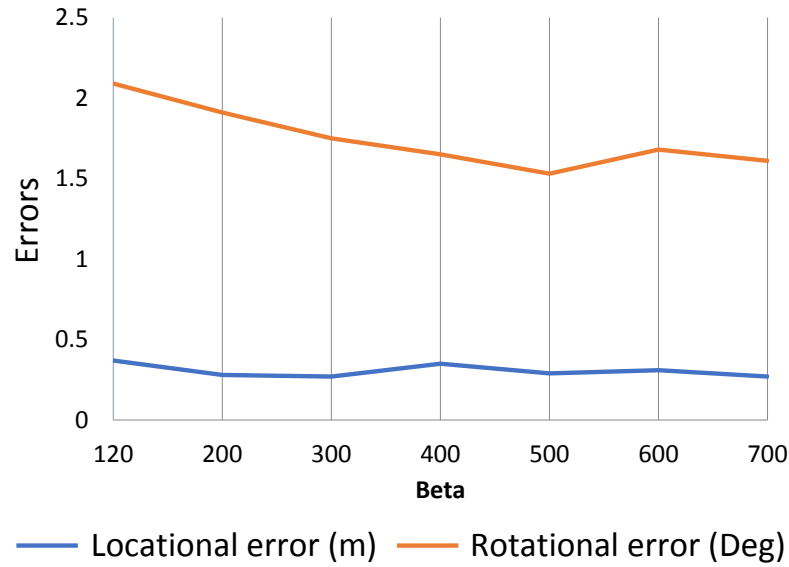


Figure 5.2: The variation of locational and rotational errors with β .

Figure 5.3 shows the distribution of location and rotation uncertainties corresponding to the different values of β . The correlation of the locational and rotational uncertainties are estimated by using the correlation coefficient R which is defined as:

$$R = \frac{cov(U_l, U_r)}{\sigma_{U_l}\sigma_{U_r}} \quad (5.3)$$

where cov is the covariance of two random variables, U_l is the Uncertainty of location, U_r is the uncertainty of rotation, σ_{U_l} is the standard deviation of U_l and σ_{U_r} is the standard deviation of U_r .

R has a range of $[-1,1]$. The sign of the coefficient denotes the positive or negative correlation, whereas the magnitude indicates the strength of the relation of two random variables. Figure 5.3 indicates that a strong positive correlation for the value of $\beta = 500$. Therefore, a β value of 500 is selected for our experiments, as the ideal network should

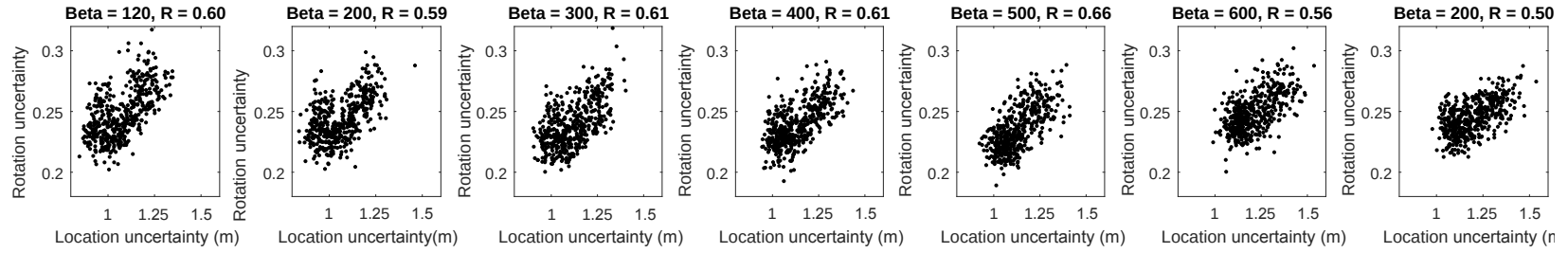


Figure 5.3: The variation of locational and rotational uncertainties with β . R denotes correlation coefficient

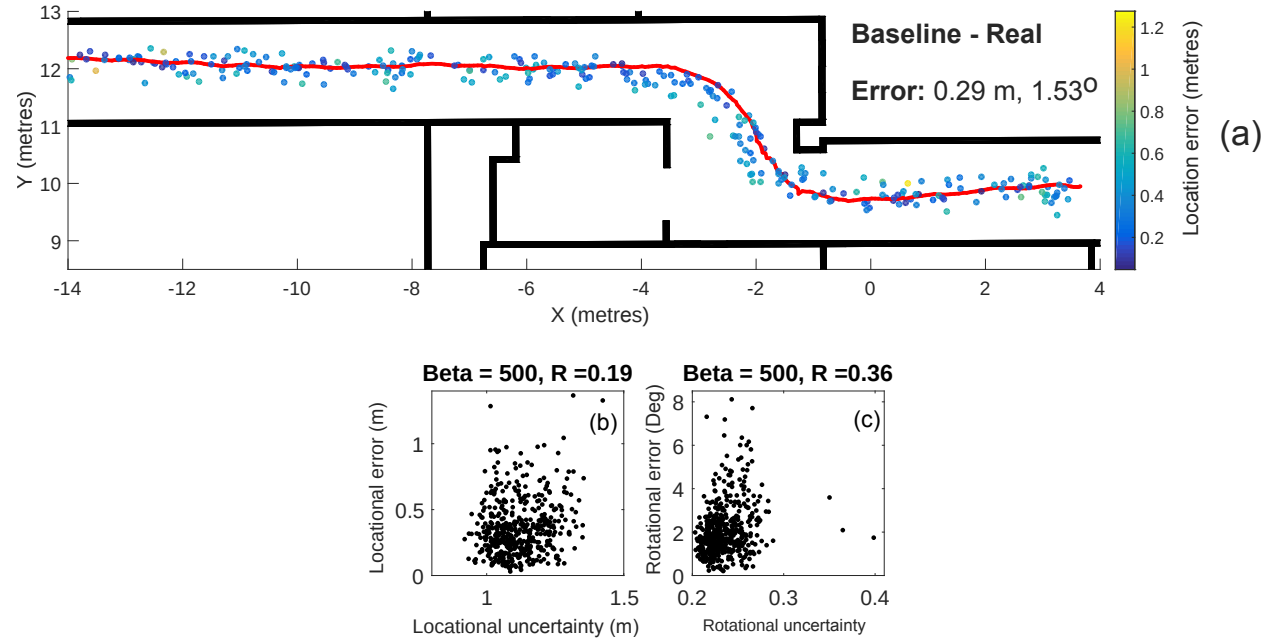


Figure 5.4: Baseline performance of a network fine-tuned and validated using real images with $\beta = 500$. (a) The distribution of the estimated camera poses. The red line shows the ground truth trajectory estimated using SfM approaches. The colours represent the magnitude of the errors corresponding to each point. The locational and rotational errors are in metres and degrees respectively. (b) locational uncertainty vs locational errors (c) rotational uncertainty vs rotational errors

have correlated locational and rotational uncertainties. Figure 5.4 (a) shows the distribution of the estimated camera poses along with the locational and rotational errors and it is observed that the largest errors are near the 90° turns of the corridor. Figure 5.4 (b) and (c) shows the variation of locational uncertainty with location errors and rotational uncertainty with rotational errors respectively.

5.3.3 Estimating uncertainty with synthetic images

In this experiment, five networks were fine-tuned with synthetic datasets and were tested with two types of real images, to estimate the best type of image suitable for performing cross-domain pose regression. Moreover, it is investigated whether the uncertainties of the estimated camera poses from a network fine-tuned with synthetic images are correlated with actual errors. The location and rotation errors, uncertainties and the distribution of the estimated points are accounted.

Three datasets namely Syn-car, Syn-pho-real and Syn-pho-real-tex are similar in appearance to the real images. Therefore networks fine-tuned with the three datasets are tested directly with real images. However, the two networks fine-tuned with Gradmag-Syn-car and Syn-edge datasets are tested with edge gradmag of real images. Figure 5.5 (a) - (e) shows the distribution of the estimated camera poses for real images with errors by the networks. Figure 5.5 (f) - (j) shows the locational and rotational uncertainties of the estimated camera poses.

In Figure 5.5 the networks fine-tuned on edge gradmag images (Gradmag-Syn-car and Syn-edge) perform better than the other networks. There is approximately two-fold accuracy improvement for the networks fine-tuned on edge gradmag images (Gradmag-Syn-car and Syn-edge), compared to Syn-car, Syn-pho-real and Syn-pho-real-tex images. The best camera pose estimations (both locational and rotational) came from the network fine-tuned with Gradmag-Syn-car dataset and tested with edge gradmag of real images, followed by the network fine-tuned with Syn-edge images.

Additionally, the distribution of the estimated locations from the network fine-tuned with Gradmag-Syn-car images (Figure 5.5 (d)) is best amongst other image types, followed by Syn-edge images (Figure 5.5 (e)). Figure 5.5 (i) - (j) show higher correlation

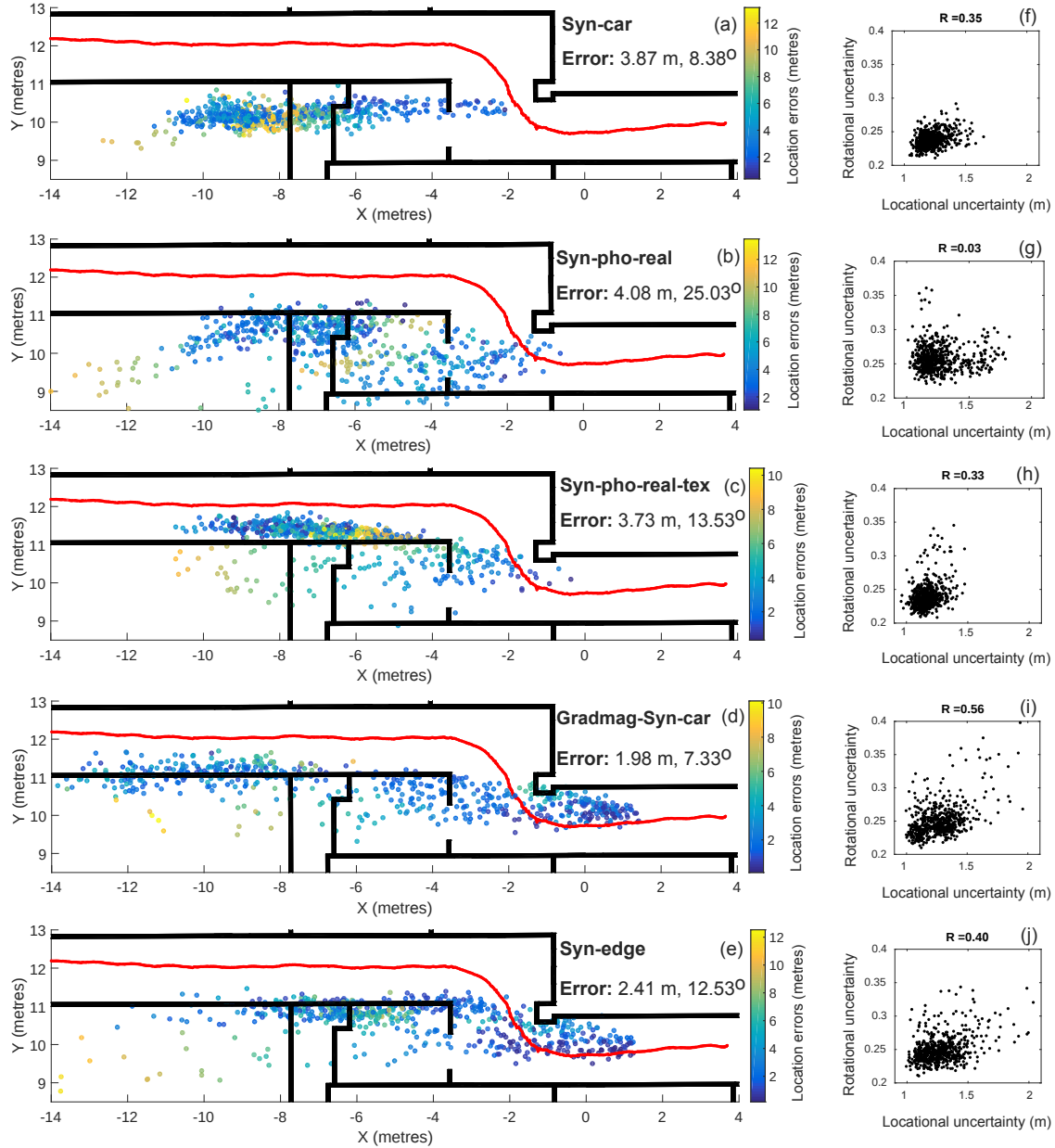


Figure 5.5: The results of the network (a) fine-tuned with Syn-car and tested on real images (b) fine-tuned with Syn-pho-real and tested on real images (c) fine-tuned with Syn-pho-real-tex and tested on real images (d) fine-tuned with Gradmag-Syn-car and tested on edge gradmag of real images (e) fine-tuned with Syn-edge and tested on edge gradmag of real images. Uncertainty plots for (f) Syn-car (g) Syn-pho-real (h) Syn-pho-real-tex (i) Gradmag-Syn-car and (j) Syn-edge datasets.

for locational and rotational uncertainties, for both the edge gradmag images, Gradmag-Syn-car being the most correlated.

It is observed from Figure 5.5 that the locational errors for the network fine-tuned with Syn-pho-real-tex is least, as compared to Syn-car and Syn-pho-real images, but the errors of all the three networks are comparable. The possible explanation could be the presence of the synthetic textures in Syn-pho-real-tex dataset that make this type of images appear similar to the real image. However, the rotational errors of network fine-tuned with Syn-car dataset are least, followed by networks fine-tuned with Syn-pho-real-tex and Syn-car images.

From Figure 5.5 (a) - (c) it is noted that the distribution of the estimated locations by the networks fine-tuned with Syn-car and Syn-pho-real are poor as compared to Syn-pho-real-tex images dataset. For all three networks, the estimated poses result in cluster formation away from the ground truth trajectory. Additionally, there exists a shift in the estimated camera locations, indicating the presence of a bias. Regarding uncertainties, from Figure 5.5 (f) - (h), it is seen that the network fine-tuned with Syn-car and Syn-pho-real-tex has the similar correlation for locational and rotational uncertainties, as compared to very low correlation for Syn-pho-real dataset. This fact explains the high errors of the Syn-pho-real images as compared to other image types.

Figure 5.6 shows the overlay of the 3D building model on five selected test images using the camera poses estimated by the network fine-tuned with Gradmag-Syn-car dataset and Table 5.1 shows the errors and uncertainties of the corresponding camera poses. The locational and rotational errors in those frames varied from 0.24 - 1.61 metres and 2.77° - 7.70° respectively. The range of locational and rotational uncertainties was 1.00 - 1.21 metres and 0.22 - 0.25 respectively.

Figure 5.7 (a) - (b) show the trend of the locational errors and uncertainties respectively for visualising the most error-prone and uncertain areas of the trajectory. Figure 5.7 (c) shows the images corresponding to the large estimation errors. In Figure 5.7 (a) - (b) it can be seen that the most substantial errors and uncertainties are present in part BC of the trajectory, which corresponds to a 90° turn and a few large errors exists near Point D, E and F.

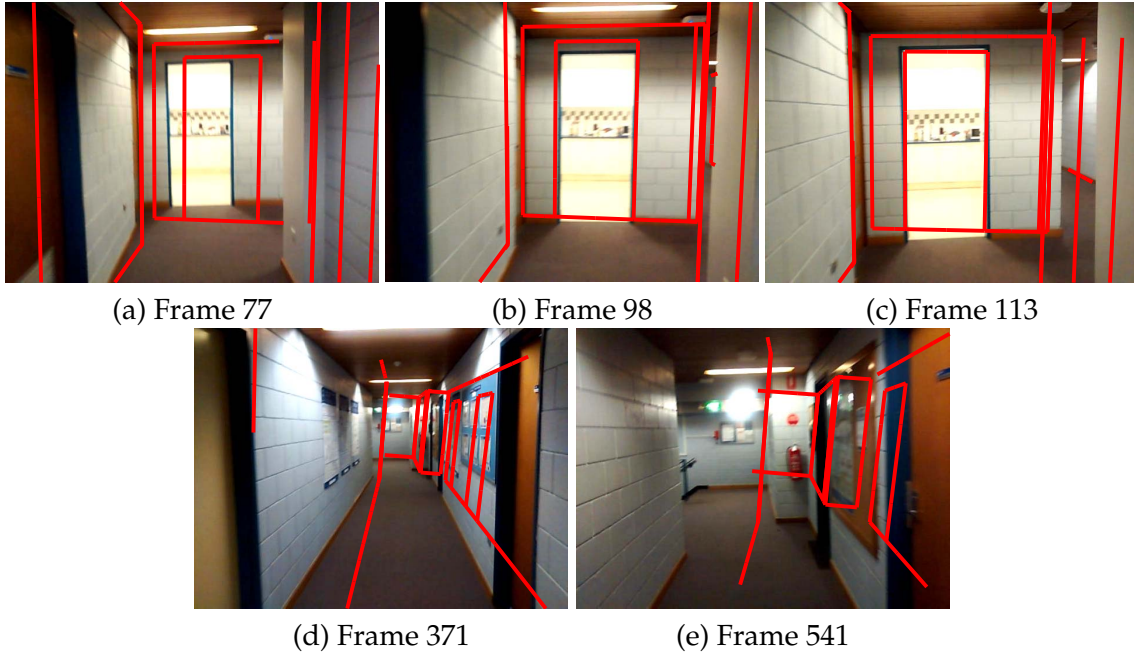


Figure 5.6: Predicted camera poses by a network fine-tuned with Gradmag-Syn-car dataset and tested with edge gradmag of real images.

Frame	Location error	Location uncertainty	Rotational error	Rotational uncertainty
77	0.55 m	1.00 m	5.74°	0.22
98	0.24 m	1.20 m	2.77°	0.24
113	0.46 m	1.10 m	5.31°	0.22
371	1.61 m	1.19 m	5.30°	0.25
541	1.05 m	1.21 m	7.70°	0.23

Table 5.1: Errors and uncertainties corresponding to Figure 5.6. The unit of the rotational uncertainty is scalar dimensionless number.

The images with large errors, as seen in Figure 5.7 (c), contain significant amount of either motion blur or artefacts. Motion blur results in low values of edge gradmag of the real images, which results in loss of information and explain the large errors in Points B, C and D of the trajectory. Artefacts are objects that are present in the real image but not in the 3D building model, such as posters and notice boards and results in higher errors for Points A, D, E and F. Additionally, artefacts are also introduced due to the structural difference of the 3D building model and the actual building, which are introduced as a result of errors in modelling the building.

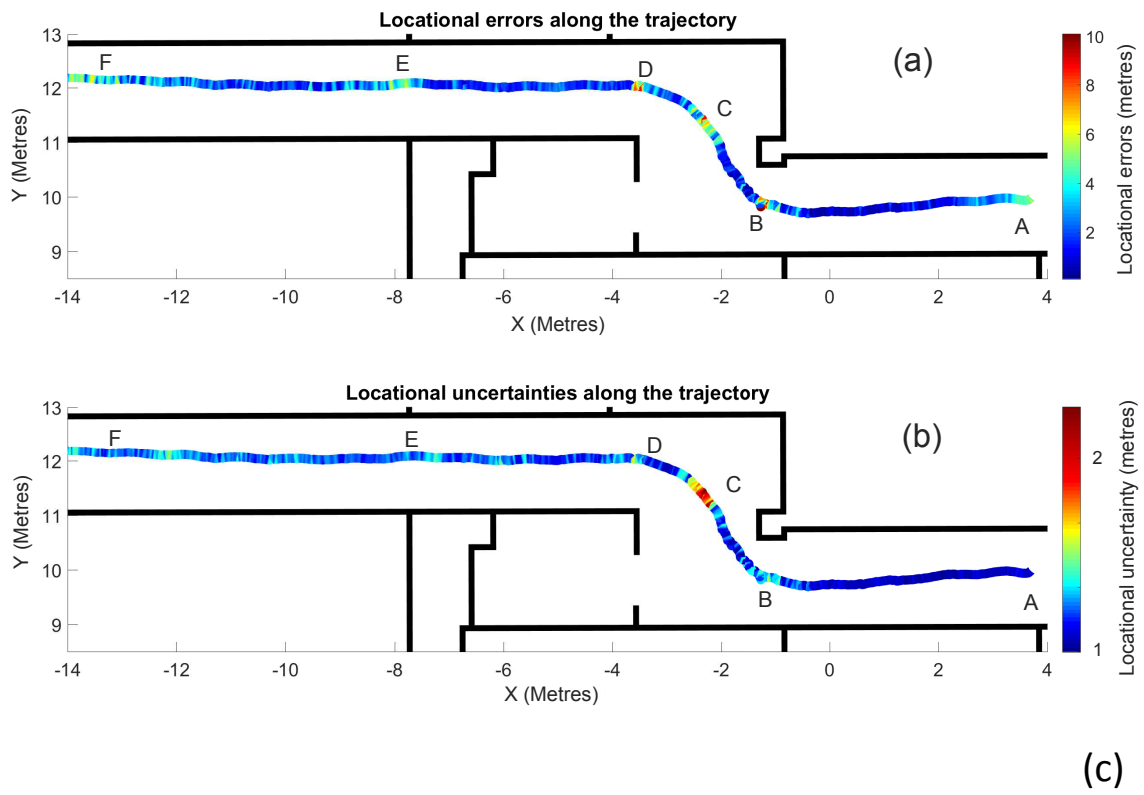


Figure 5.7: (a) shows the average normalised errors of the trajectory (b) shows the average normalised uncertainties of the trajectory and (c) shows corresponding points on the trajectory with high errors.

Table 5.2 shows the errors and uncertainties of the frames shown in Figure 5.7 (c) with high errors. The locational and rotational errors in those frames varied from 6.74 - 10.10 metres and 9.01° - 62.10° respectively. The range of locational and rotational uncertainties was 1.20 - 1.75 metres and 0.24 - 0.35 respectively. Comparing Table 5.2 with Table 5.1 indicate that larger uncertainties are associated with images having larger errors.

Frame	Location error	Location uncertainty	Rotational error	Rotational uncertainty
10	6.97 m	1.69 m	10.05°	0.25
167	10.10 m	1.41 m	62.10°	0.27
240	6.93 m	1.75 m	42.73°	0.35
277	8.07 m	1.57 m	19.02°	0.25
405	6.74 m	1.20 m	23.34°	0.27
591	8.19 m	1.37 m	9.01°	0.24

Table 5.2: Errors and uncertainties corresponding to Figure 5.7 (c)

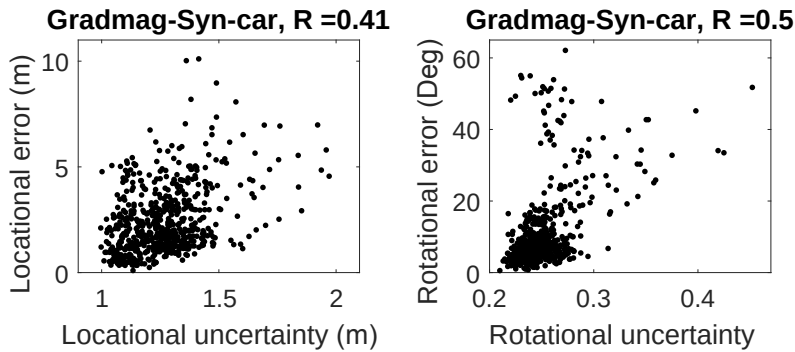


Figure 5.8: (a) Location uncertainty vs error (b) Rotational uncertainty vs errors for Gradmag-Syn-car images.

Figure 5.8 shows the correlation of the locational errors with locational uncertainties and rotational errors with rotational uncertainties. Figure 5.8 (a) indicates the locational uncertainties are more correlated with the location errors as compared to baseline experiment with real images in Figure 5.4 (b). Similar trend is observed for rotational errors and uncertainties when comparing Figure 5.8 (b) with 5.4 (c). Thus, the estimated uncertainties can be used as a measure of errors and are particularly useful in the absence of ground truth, as it provides the confidence of the camera pose estimations.

5.3.4 Discussion and future directions

In indoor spaces, specifically with repetitive structures, two places might appear similar, and it is impossible to differentiate between them without some auxiliary information. This problem is referred to as perceptual aliasing (Lowry et al., 2016) and is also a limitation of the current approach. For example, consider frame 240 in Figure 5.7 (c), where the camera is close to the wall, and only one door is visible. As all the doors and walls in the building have the same appearance, it creates confusion for pose estimation. This phenomenon can explain the high errors and uncertainties near part Point C of the trajectory in Figure 5.7. Similarly, the high errors in the benchmark experiment with real images can be explained based on this fact.

The proposed approach is independent of any central processing server and hence does not need any additional infrastructure for operation, in addition, the network is very lightweight concerning storage (approximately 50 MB). The processing time for each image on a GPU is approximately 67 milliseconds and 9.07 seconds on an i7 CPU. Fortunately, most of the modern pervasive devices contain a GPU that can be used to accelerate the processing.

The achievable accuracy by the localisation approach is suitable for pedestrian navigation (Taneja et al., 2011) but is not sufficient for robotic applications; however, there is still room for improvement. A potential future direction to overcome the perceptual aliasing could be the use of Kalman filter to integrate the uncertainties derived from the network with the spatio-temporal information from a video sequence, or the use of long-short-term-memory (LSTM) to exploit spatio-temporal information alone. On the other hand, to address the computational challenge, another future direction for the research could be exploring mini-networks that are made explicitly for operation in mobile devices.

5.4 Conclusions

The uncertainty of camera pose regression of BIM-PoseNet is modelled by fine-tuning a Bayesian neural network with different types of synthetic images generated from a 3D

building model, and regress the camera pose of real images. The proposed approach eliminates the requirement of performing SfM approaches before operation. The experiments indicate that a localisation accuracy of approximately two metres can be achieved by representing real images as edge gradient magnitude images. The localisation errors are the result of the presence of motion blur, artefacts in the images and perceptual aliasing. Moreover, the uncertainties associated with the estimated locations are correlated with the localisation errors, thus providing the measure with which the estimated locations can be trusted.



"... the spatio-temporal dependency of consecutive images improves localisation performance ..."

≈ Block B, Department of Infrastructure Engineering, The University of Melbourne, Australia
(Courtesy: Agisoft PhotoScan Professional®)

Chapter 6

Recurrent BIM-PoseNet: Learning pose estimation from synthetic image sequences

A limitation of BIM-PoseNet and Bayesian BIM-PoseNet is the low accuracy that occurs as a result of estimating the pose of each image frame independently. In this chapter, a visual localisation approach called Recurrent BIM-PoseNet is proposed that exploits the spatio-temporal information from synthetic image sequences to improve the 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 Recurrent BIM-PoseNet estimates a smoother trajectory with smaller inter-frame error as compared to existing methods. The achievable accuracy with Recurrent BIM-PoseNet is 1.6 metres, which is an improvement of approximately thirty per cent compared to the existing approaches.

6.1 Introduction

Previous chapters revealed that BIM-PoseNet and Bayesian BIM-PoseNet eliminate the requirement of 3D image-based reconstruction by making use of a 3D indoor model. These approaches fine-tune a deep CNN using synthetic images rendered from a building information model (BIM), to regress the camera pose of real images. However, the camera pose for each image frame is estimated, without considering the spatial dependency between the consecutive frames. The per-frame error varies largely, thus resulting in high variance of distances between consecutive frames. Another limitation of pose regression methods based on a single image is the so-called perceptual aliasing where indoor locations with similar appearances result in large pose estimation errors.

Therefore, a relevant question is whether the performance of approaches that use synthetic images for visual localisation, such as BIM-PoseNet (Acharya et al., 2019a) and Bayesian BIM-PoseNet (Acharya et al., 2019c), can be improved by applying the constraint of spatio-temporal dependencies between consecutive images. Besides, the uncertainty of camera pose plays a vital role in the estimation, as it provides the confidence of the predicted camera poses in the absence of ground truth. Ideally, localisation errors should be correlated with the estimated uncertainties. The uncertainty of camera pose have been modelled using synthetic images (Acharya et al., 2019c). However, that approach estimates the localisation uncertainty from a single image without considering the spatio-temporal continuity of image sequences. Therefore, another relevant question is explored as to whether the uncertainty of camera pose can be modelled by using a sequence of synthetic images.

In this chapter, Recurrent BIM-PoseNet is proposed, which is a deep Bayesian recurrent CNN to localise a single image, using a short sequence of images. Previous methods that use a single image for pose regression (e.g. Kendall et al., 2015) give pose estimates that vary largely for consecutive images resulting in a jagged trajectory. 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. Smoothness is defined as having consistent inter-frame distances, so that there are no sudden *jumps* between two consecutive camera poses.

The network is fine-tuned using synthetic image sequences generated from a BIM, therefore eliminating the requirement of 3D image-based reconstruction of indoor spaces. At the test time, the camera poses of a sequence of real image is regressed. It is shown that Recurrent BIM-PoseNet results in a smoother trajectory with smaller inter-frame errors as compared with BIM-PoseNet (Acharya et al., 2019a) and Bayesian BIM-PoseNet (Acharya et al., 2019c). Additionally, the estimated uncertainties show correlation with the localisation errors, suggesting the suitability of the approach for modelling uncertainty. Moreover, the proposed method does not need any color or texture information to perform localisation. The following are the contributions:

1. The proposed approach eliminates the requirement of 3D image-based reconstruction for camera pose regression. In the proposed approach, a network is trained by synthetic images to estimate the pose of real images. The proposed method improves the localisation accuracy of previous pose regression networks.
2. The spatio-temporal information of the synthetic image sequences is utilised to generate a smoother trajectory when tested with real image sequences.
3. The uncertainty of camera pose estimation is modelled, and shows correlation with the actual errors obtained from a ground truth trajectory.

In Section 6.2, the theory and methodology of Recurrent BIM-PoseNet are explained, which is followed by experiments and results in Section 6.3. Section 6.4 concludes with a summary of the outcomes of the research and possible future directions.

6.2 Methodology

The design of the proposed approach is shown in Figure 6.1. 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 architecture of Recurrent BIM-PoseNet is described in Section 6.2.1, fine-tuning and loss function are described in Section 6.2.2, and Section 6.2.3 explains the uncertainty modelling for camera pose estimations. The implementation details are presented in Section 6.3.1.

6.2.1 Deep learning Architecture

The proposed network consists of a Bayesian CNN module and an LSTM module. The LSTM (Hochreiter and Schmidhuber, 1997) 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

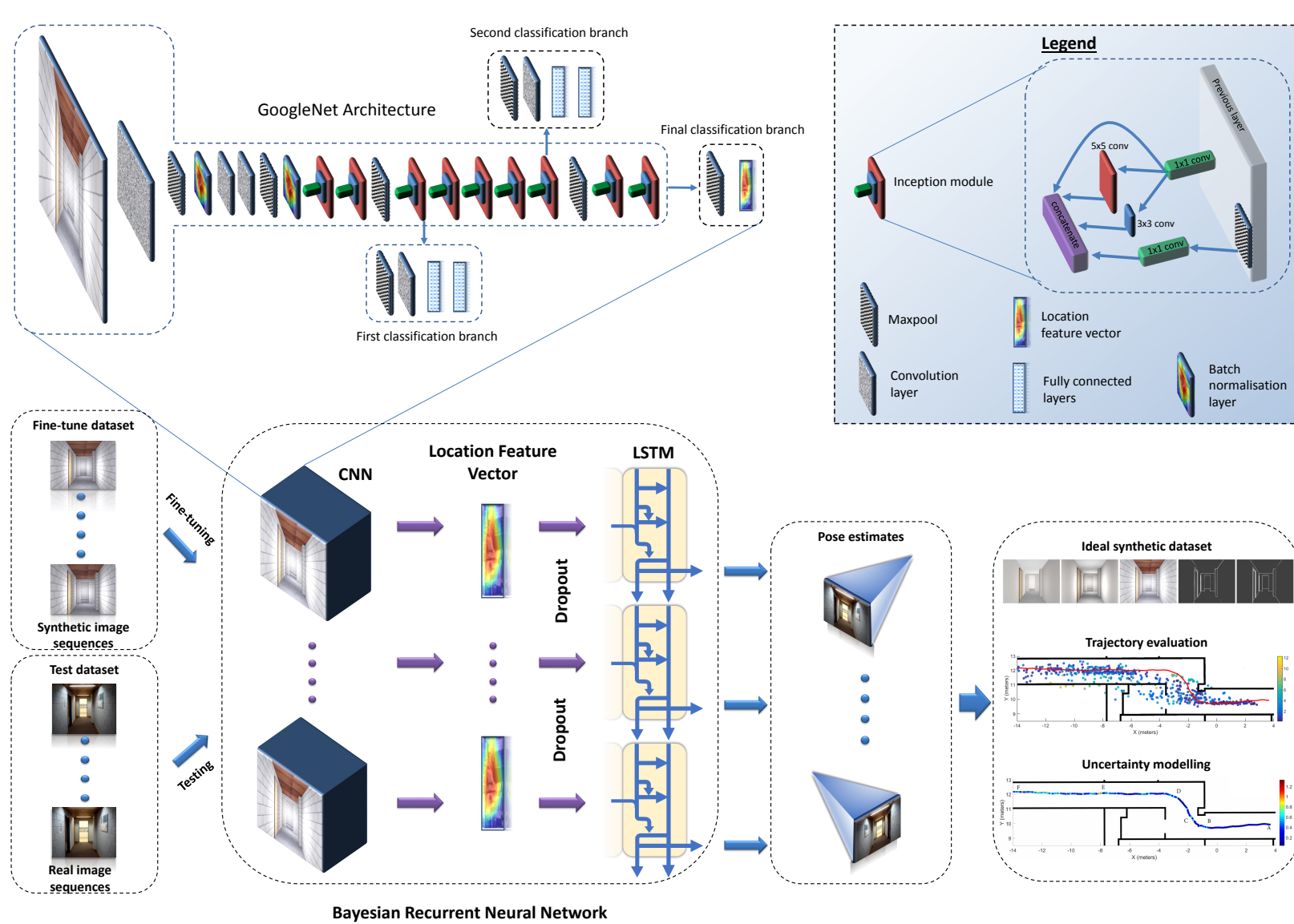


Figure 6.1: The design of the proposed approach. The network used is GoogleNet (Szegedy et al., 2015) containing nine inception modules. The output of the final classification branch yields the location feature vector which serves as input to the LSTM layer.

data in and out of the memory cells that is regulated by the input and forget gates. The activations of the LSTM (h_t) at time step t for a location feature vector v_t are calculated by the following equations:

$$\begin{aligned}
 f_t &= \sigma(W_f v_t + U_f h_{t-1} + b_f) \\
 i_t &= \sigma(W_i v_t + U_i h_{t-1} + b_i) \\
 o_t &= \sigma(W_o v_t + U_o h_{t-1} + b_o) \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_h(W_c v_t + U_c h_{t-1} + b_c) \\
 h_t &= o_t \circ \sigma_h(c_t)
 \end{aligned} \tag{6.1}$$

where, f_t is the forget gate's activation vector, i_t is the input gate's activation vector, o_t is the output gate's activation vector, c_t is the cell state vector and h_t is the hidden state vector. W, U and b are the weights and the bias vectors of the LSTM that are learnt during the training process. σ and σ_h denote the sigmoid function activation and hyperbolic tangent function respectively, and $\circ$ denotes the Hadamard product of two vectors.

The CNN module uses a deep CNN called GoogleNet (Szegedy et al., 2015) for image feature extraction. Similar to PoseNet (Kendall et al., 2015) the softmax layers are removed, and a 2048-dimensional dense layer is added, which acts as the location feature vector. The initial weights of the network are updated from GoogleNet pre-trained on the Places image dataset (Zhou et al., 2014) due to its suitability for the task of scene classification and localisation (Kendall et al., 2015).

In the literature, sliding window has been used to generate samples from the data, which are used in conjunction with LSTMs to exploit the spatio-temporal information of consecutive samples, and subsequently to model the uncertainty of the estimations (Zhu and Laptev, 2017; Clark et al., 2017). Therefore, to capture the spatio-temporal information, sequences of consecutive images are generated by sliding a window of length n , from the total number of N images, and 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 module. The output of the hidden states of the LSTM is utilised to regress the camera pose of the images in the sequence.

6.2.2 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:

$$p_t^T = [x_t^T, q_t^T] \quad (6.2)$$

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 .

The network is fine-tuned by minimising the following objective loss function (L) that represents the errors of the whole sequence of images by unrolling the network and performing back-propagation through time:

$$L = \sum_{t=1}^{t=n} \|\hat{x}_t^T - x_t^T\|_2 + \beta \|\hat{q}_t^T - q_t^T\|_2 \quad (6.3)$$

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.

6.2.3 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 (Gal and Ghahramani, 2015). 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. The trace of the covariance matrices of the camera pose samples is used to model uncertainty, which provides a good numeric measure of the uncertainty (Kwon et al., 2018; Depeweg et al., 2017). The multiple pose predictions by the network are utilised in the following manner to model the location and rotation uncertainties:

$$\begin{aligned}
U_{x_t}^T &= \text{sqr}t(\text{trace}(C(x_1^T, x_2^T, \dots, x_n^T))) \\
U_{q_t}^T &= \text{sqr}t(\text{trace}(C(q_1^T, q_2^T, \dots, q_n^T)))
\end{aligned} \tag{6.4}$$

where, $U_{x_t}^T$ and $U_{q_t}^T$ denotes the uncertainty of camera location and rotation, respectively for the T^{th} frame, and C denotes the covariance matrix. Subsequently, the correlation factor R (Acharya et al., 2019c) is defined to quantify the correlation of the estimated localisation uncertainty and localisation errors in the following equations:

$$\begin{aligned}
R_{LU/LE} &= \frac{\text{cov}(U_x^N, E_x^N)}{\sigma_{U_x^N} \sigma_{E_x^N}} \\
R_{RU/RE} &= \frac{\text{cov}(U_q^N, E_q^N)}{\sigma_{U_q^N} \sigma_{E_q^N}} \\
R_{LU/RU} &= \frac{\text{cov}(U_x^N, U_q^N)}{\sigma_{U_x^N} \sigma_{U_q^N}}
\end{aligned} \tag{6.5}$$

where $R_{LU/LE}$ denotes the correlation of estimated location uncertainties vs location errors, $R_{LU/RU}$ 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 $\sigma_{E_q^N}$ denotes 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.

6.3 Experiments and results

6.3.1 Implementation details

The networks were implemented in Keras (Chollet et al., 2015) using TensorFlow libraries (Abadi et al., 2015) 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 320x240 pixels, and a central crop of dimension 224x224 was applied. The mean of the fine-tuning dataset was subtracted from the input images during fine-tuning and test time.

6.3.2 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, the following experiments are designed.

- 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 6.3.4.
- 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 6.3.5.
- 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 6.3.6.

6.3.3 Dataset

Two datasets were used in the experiments, namely a synthetic image dataset and a real image dataset. More details of generating the dataset and the example images (Figure 4.2 and Figure 4.4) can be found in Section 4.3.1.

6.3.4 Experiment 1: Baseline performance using real image sequences

The process of generation of ground truth camera poses is described in Section 4.3.2. The value of the weighing parameter β in the loss function (Equation 6.3) is set to 600 based on our previous experiments (Acharya et al., 2019a). The dataset was partitioned into fine-tuning, validation and testing sets maintaining the sequence of the images.

Achievable accuracy

The length of the LSTM units is important for network performance. A longer length of LSTM units will lead to over-fitting, where the network will perform excellently for the validation data but will perform poorly for another 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, an experiment is performed 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, 512 LSTM units are used.

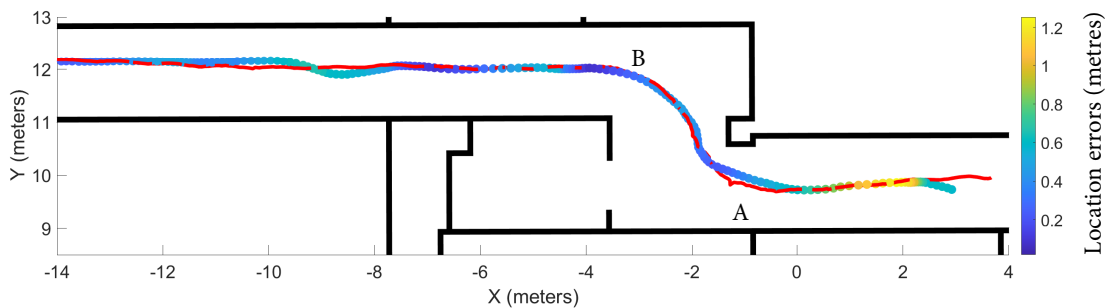


Figure 6.2: The predictions of Recurrent BIM-PoseNet fine-tuned and tested using a sequence of ten real images. The colour represents the magnitude of error of each point, and the red line denotes the ground truth trajectory.

To identify the ideal window length the networks were fine-tuned and tested using different window lengths of images. Four networks are fine-tuned 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 metres and 0.83° . Figure 6.2 shows the distribution of the predicted points by the network.

Estimated uncertainty

Figure 6.3 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 (Acharya et al., 2019c), 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.

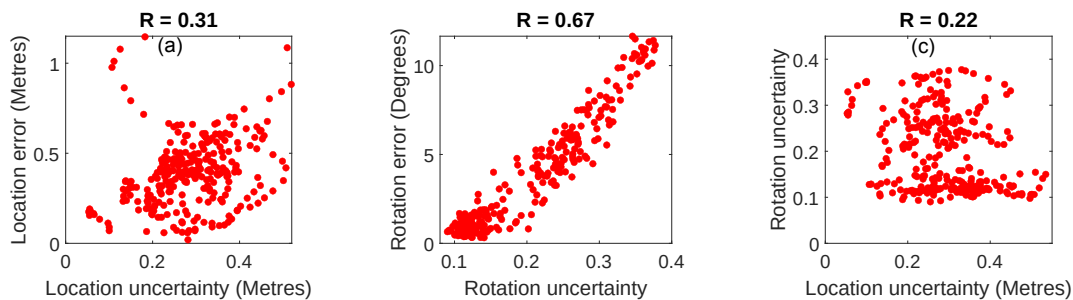


Figure 6.3: The modelled uncertainty by Recurrent BIM-PoseNet. (a) Estimated location uncertainty vs location errors (b) Estimated rotation uncertainty vs rotation errors (c) Estimated location uncertainty vs estimated rotation uncertainty. R denotes the correlation factor.

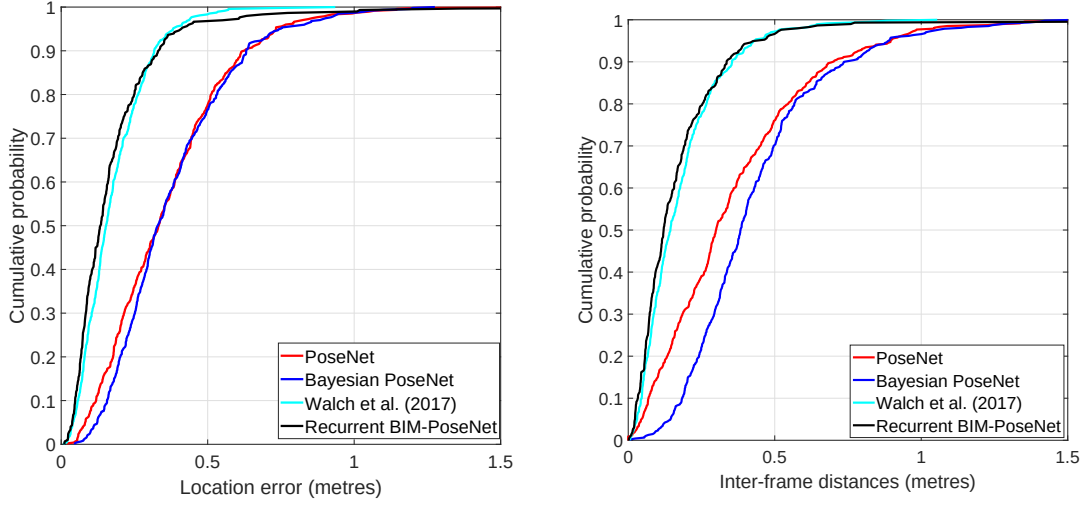


Figure 6.4: Cumulative distribution function (CDF) of the (a) errors and (b) inter-frame distances of the predictions by BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for the baseline performance using real images.

Table 6.1: The comparison of the median errors and inter-frame distances of BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for the baseline experiment.

Approach	Median errors (metres, Degrees)	Median inter-frame distances (metres)
BIM-PoseNet	0.33m, 1.85°	0.30m
Bayesian BIM-PoseNet	0.29m, 1.53°	0.38m
Recurrent BIM-PoseNet	0.17m, 0.83°	0.15m
Walch et al.(2017)	0.15m, 1.62°	0.14m

Comparison with previous approaches

Figure 6.4 shows the cumulative distribution function (CDF) of the location errors and the inter-frame distances for Recurrent BIM-PoseNet compared to BIM-PoseNet, Bayesian BIM-PoseNet and Walch et al. (2017). 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. Compared to BIM-PoseNet and Bayesian BIM-PoseNet, Recurrent BIM-PoseNet achieves smaller localisation errors and inter-frame distances for the predictions. However compared to the work of Walch et al. (2017), a similar trend is noticed in the accuracy and inter-frame distances (i.e. smoothness).

Table 6.1 summarises the median errors and the inter-frame distances for the four approaches. It can be seen that even by using a window of ten images, the location and rotation errors for Recurrent BIM-PoseNet approach are reduced by a factor of 2 as compared with BIM-PoseNet and Bayesian BIM-PoseNet. Besides, the effect of perceptual aliasing is reduced which was one of the main limitations of the BIM-PoseNet and Bayesian BIM-PoseNet. For instance, consider the part AB in Figure 6.2 of the trajectory, where the errors of the camera pose estimates are reduced compared to the previous approaches. Compared to Walch et al. (2017), comparable errors and inter-frame distances are noticed, with a slight improvement in the rotational errors.

6.3.5 Experiment 2: Performance of the network fine tuned using synthetic images

Recurrent BIM-PoseNet was fine-tuned with the synthetic image datasets namely, Syn-car, Syn-pho-real, Syn-pho-real-tex, Gradmag-Syn-car and Syn-edge. The value of the loss function parameter β was kept 600 for the fine-tuning. Naive (0 – 0.5) and variational dropouts (0 – 0.5) are used for the LSTM units (Gal and Ghahramani, 2016b) which improved the accuracy of camera pose regression for synthetic image sequences (However, for real image sequences adding dropouts to LSTM units showed no improvement, where all the connections were retained).

Similar to the experiments with real data, to identify the ideal length of the LSTM units for synthetic image sequences, networks are fine-tuned 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 (Acharya et al. (2019c)) 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 6.5 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

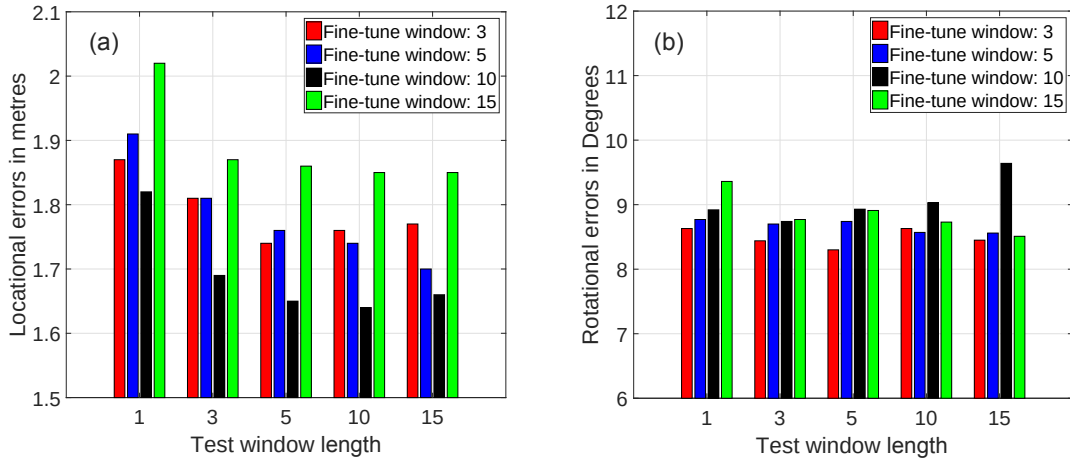


Figure 6.5: The effects of window length on the (a) location errors and (b) rotation errors for networks fine-tuned using Gradmag-Syn-car dataset and tested on Gradmag of real images.

real images, Figure 6.5 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 metres and 9.29° .

The fine-tuned networks can generalise well on a longer window of test image sequences than 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 it was fine-tuned. Figure 6.5 (b) shows that there is a slight decrease in the rotation errors with the increase in the length of the window of the image sequence.

Table 6.2: The comparison of the errors of BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for the five networks fine-tuned using the different synthetic image datasets and tested using real images.

Fine-tuned on	Tested on	BIM-PoseNet (metres, degrees)	Bayesian BIM PoseNet (metres, degrees)	Recurrent BIM-PoseNet (metres, degrees)	Walch et al. (2017) (metres, degrees)
Syn-car	Real images	6.25m, 37.16 ^o	3.87m, 8.38 ^o	3.97m, 15.20 ^o	4.09m, 22.28 ^o
Syn-pho-real	Real images	5.99m, 11.33 ^o	4.08m, 25.03 ^o	3.01m, 8.50 ^o	2.88m, 15.31 ^o
Syn-pho-real-tex	Real images	3.06m, 12.25 ^o	3.73m, 13.53 ^o	2.23m, 8.31 ^o	2.50m, 11.99 ^o
Gradmag-Syn-car	Gradmag of real	2.63m, 6.99 ^o	1.98m, 7.33 ^o	1.62m, 9.29 ^o	2.89m, 19.22 ^o
Syn-edge	Gradmag of real	1.88m, 7.73 ^o	2.41m, 12.53 ^o	1.87m, 11.15 ^o	1.90m, 12.42 ^o

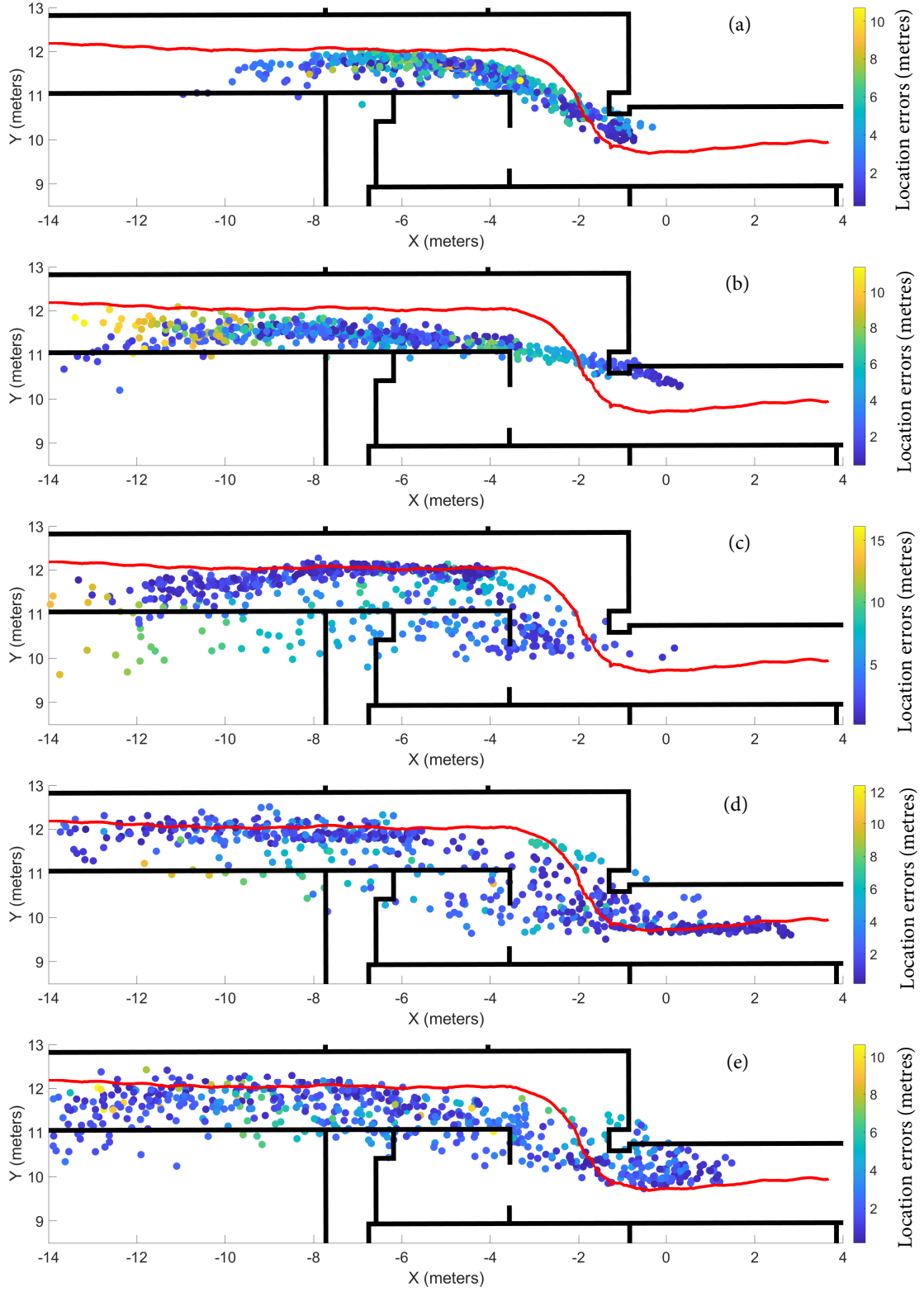


Figure 6.6: Estimated camera poses by the proposed Recurrent BIM-PoseNet (a) fine-tuned using Syn-car and tested on real images (b) fine-tuned using Syn-pho-real and tested on real images (c) fine-tuned using Syn-pho-real-tex and tested on real images (d) fine-tuned using Gradmag-Syn-car and tested on gradmag of real images (e) fine-tuned using Syn-edge and tested on edge gradmag of real images.

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 were used, and for testing a window of ten real images were used. Figure 6.6 shows the estimated trajectories by the networks and Table 6.2 shows the test errors of Recurrent BIM-PoseNet compared to the previous approaches.

The proposed Recurrent BIM-PoseNet achieves a better localisation accuracy than BIM-PoseNet, Bayesian BIM-PoseNet for all renderings of synthetic images. Compared to Walch et al. (2017) improvements are noticed for all renderings except for Syn-car and Syn-pho-real renderings. The improvement on rotation accuracy however is apparent only for some of the renderings. Table 6.2 also shows that among the different renderings of the synthetic images the network fine tuned using Gradmag-Syn-car images achieves the best localisation accuracy. This trend can also be seen in Figure 6.6 (d) that shows the improvement of the distribution of the estimated points by the networks fine-tuned using Gradmag-Syn-car images as compared to the other three networks. The test errors (fine-tuned with Gradmag-Syn-car and tested on gradmag of real) with Recurrent BIM-PoseNet are reduced to 1.62 metres, 9.29° , as compared to 2.63 metres, 6.99° for BIM-PoseNet, 1.98 metres, 7.33° for Bayesian BIM-PoseNet and 2.89 metres, 19.22° for Walch et al. (2017).

The synthetic Gradmag-Syn-car images achieve the best localisation performance and agrees with the results of Bayesian BIM-PoseNet, but is different from those of BIM-PoseNet and Walch et al. (2017), where Syn-edge images achieve the best performance. However, for all approaches the performance of Gradmag-Syn-car and Syn-edge is quite similar and noticeably better than that of the other types of synthetic images. Additionally, the bias or a shift in the location of the estimated points that was reported in

BIM-PoseNet and Bayesian BIM-PoseNet is greatly reduced, especially for the network fine-tuned using Gradmag-Syn-car dataset.

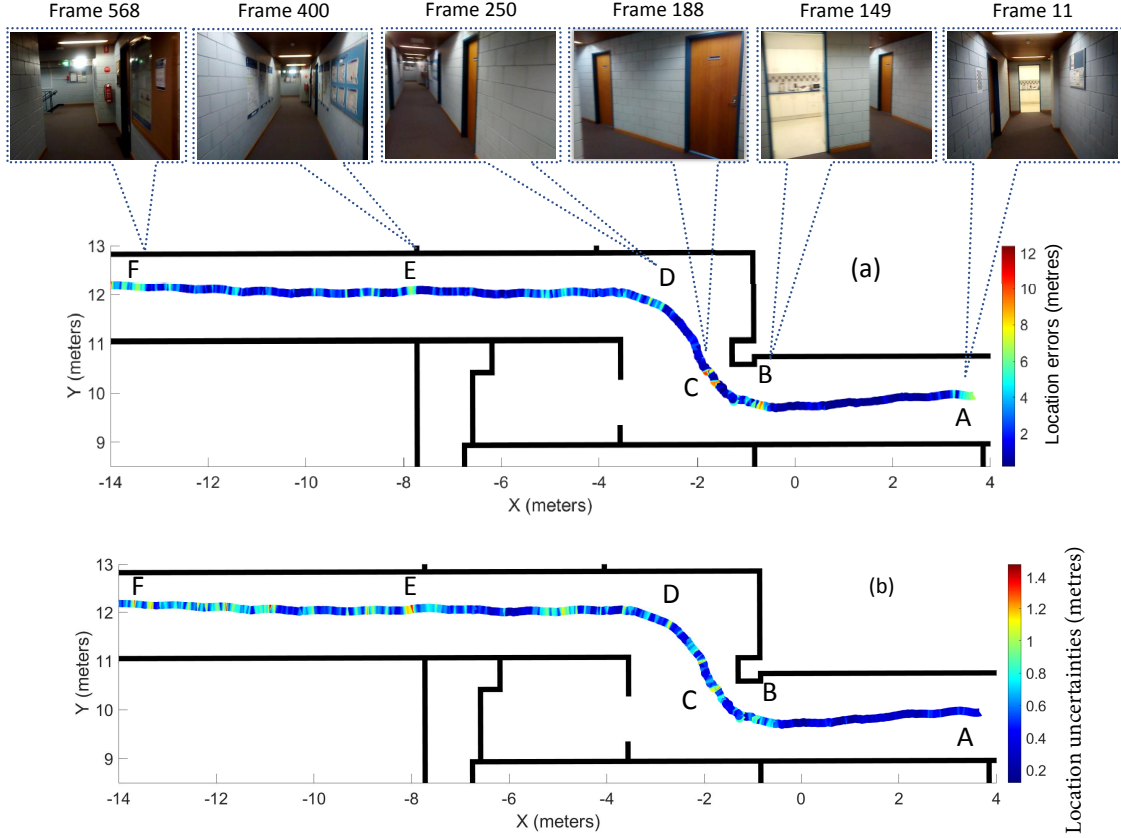


Figure 6.7: The distribution of the (a) Location errors and (b) Estimated location uncertainties along the trajectory for the network fine-tuned using Gradmag-Syn-car dataset. The first row shows some of the frames with large location errors near Point A - E. Table 6.3 shows the errors and estimated uncertainties of the frames.

Table 6.3: The comparison of the errors and estimated uncertainties for the frames of Figure 6.7.

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

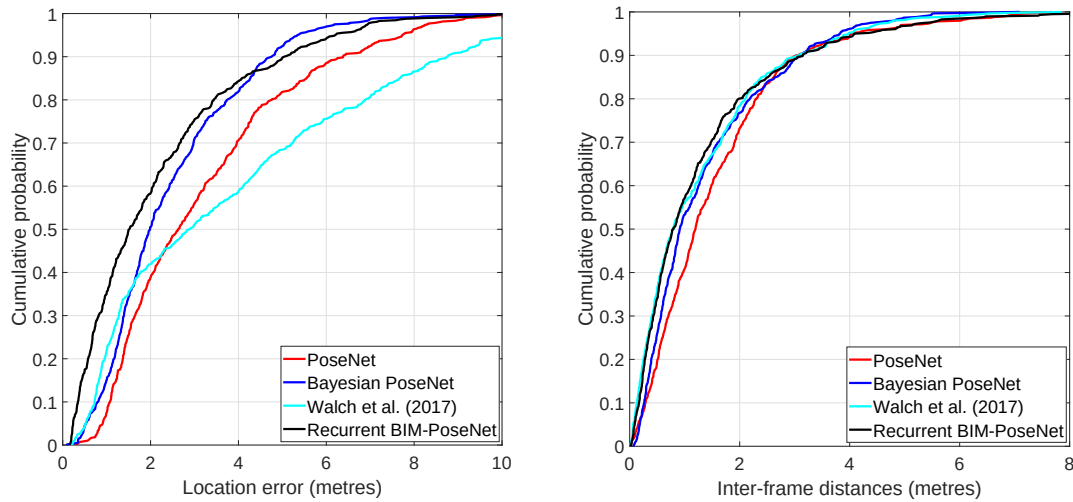


Figure 6.8: Cumulative distribution function (CDF) of the (a) localisation errors and (b) inter-frame distances of the predictions by BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet for Gradmag-Syn-car dataset.

Figure 6.7 (a) shows the distribution of the localisation errors and some of the frames that resulted in largest errors along the trajectory for the network fine-tuned on Gradmag-Syn-car dataset, to identify error-prone areas. Table 6.3 shows the localisation 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, Point A. Other error sources include the motion blur that results in larger errors for the Point 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 sources of errors. The errors due to the perceptual aliasing near Point D in Figure 6.7 (a), are reduced for Recurrent BIM-PoseNet (up to 6 metres) as compared to BIM-PoseNet (up to 8 metres) and Bayesian BIM-PoseNet (up to 10 metres), for the same part of the trajectory.

Figure 6.8 shows the cumulative distribution function (CDF) of the localisation errors and the inter-frame distances of Recurrent BIM-PoseNet for Gradmag-Syn-Car images, as compared with BIM-PoseNet, Bayesian BIM-PoseNet and Walch et al. (2017). The localisation errors are calculated as the Euclidean distances from the ground truth trajectory and the inter-frame distances are the Euclidean distances between two consecutive

frames. It is evident that Recurrent BIM-PoseNet outperforms BIM-PoseNet, Bayesian BIM-PoseNet and [Walch et al. \(2017\)](#) in terms of localisation errors. However, in terms of inter-frame distances, only a slight improvement is identified.

6.3.6 Experiment 3: Modelling uncertainty using synthetic images

Figure 6.9 shows the correlation between the estimated uncertainties and the actual errors for the networks. Figure 6.9 (a) - (i) shows 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, and 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. Also, the correlation of the estimated location uncertainties vs estimated rotation uncertainties are least as compared to the other datasets. Table 6.4 compares the correlation factor for the uncertainties of Bayesian BIM-PoseNet with Recurrent BIM-PoseNet.

Figure 6.9 (j) - (o) shows 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

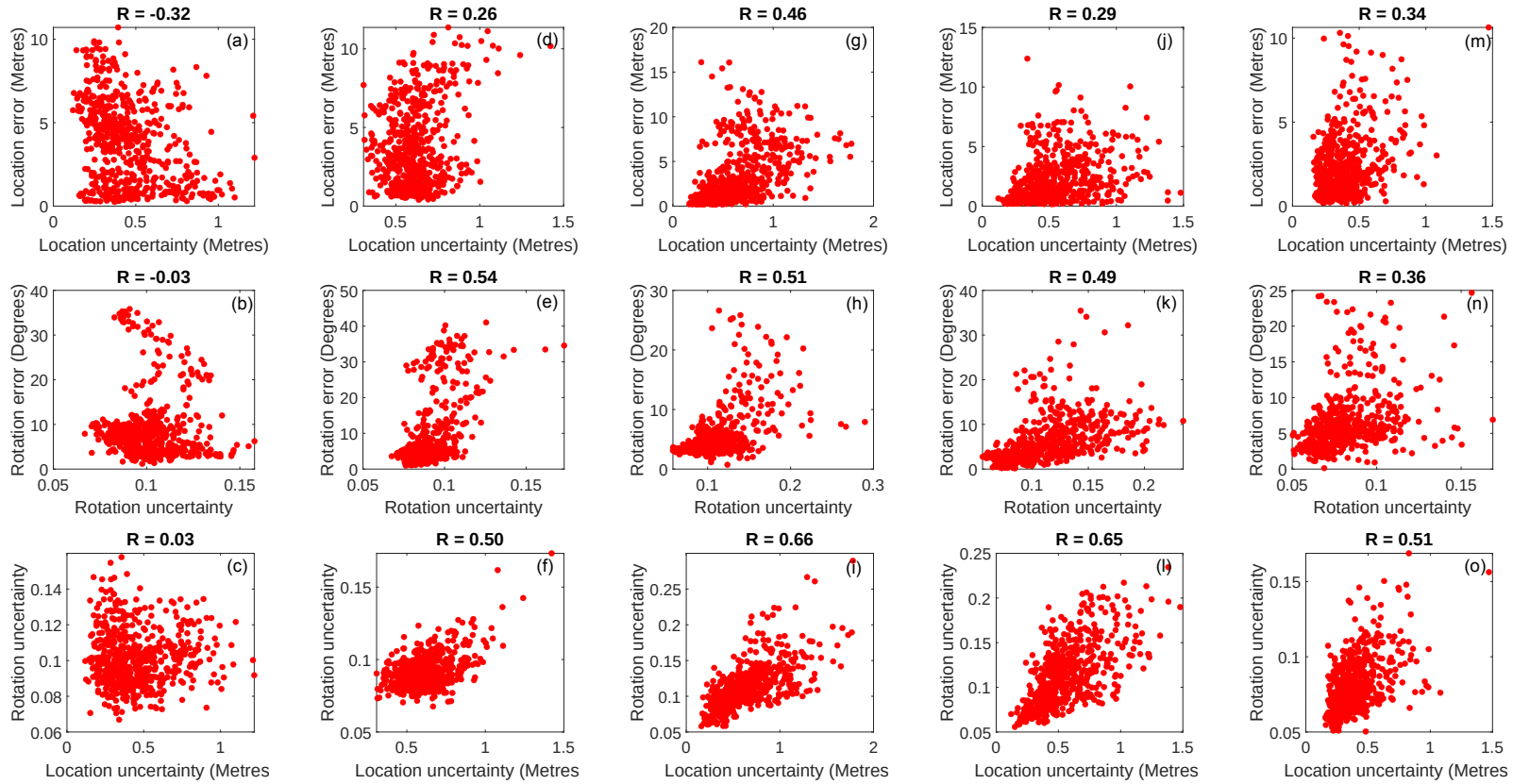


Figure 6.9: The uncertainty modelled by the network (a) - (c) fine-tuned using Syn-car and tested on real images, (d) - (f) fine-tuned using Syn-pho-real and tested on real images, (g) - (i) fine-tuned using Syn-pho-real-tex and tested on real images, (j) - (l) fine-tuned using Gradmag-Syn-car and tested 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.

Table 6.4: 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

estimated rotational uncertainties improve for both the datasets and is summarised in Table 6.4. 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 is better as compared to Bayesian BIM-PoseNet (Table 6.4).

Figure 6.7 (b) 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 6.7 (a)) 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 6.7. 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 Point E and F.

6.3.7 Computation times

Table 6.5 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.

Table 6.5: 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 (millisec)	Test time CPU (sec)
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 ([Sehgal and Kehtarnavaz, 2019](#)), 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.

6.4 Conclusions

A deep Bayesian recurrent CNN is proposed 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 metres 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, it is showed 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.

The accuracy of the proposed approach is suitable for many indoor applications such as pedestrian navigation, but, this accuracy might not be suitable for autonomous robots, such as unmanned ground vehicles. However, there is still some room of improvement 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 ([Long et al., 2015](#)) can be used.



"... Dawn or Dusk? ..."

≈ Masson Road, Melbourne School of Design, The University of Melbourne, Australia

Chapter 7

Conclusions

This thesis explores visual indoor localisation approaches that are infrastructure- independent, are suitable for mass-market applications, and can be applied to ubiquitous devices. Following that objective the thesis addresses the two challenges of the state-of-the-art visual indoor localisation approaches, using a 3D building model. The first addressed challenge is the accumulation of error for visual approaches like simultaneous localisation and mapping (SLAM) and visual odometry. The second addressed challenge is the requirement of 3D image-based reconstruction of the indoor space for camera pose regression approaches using convolutional neural networks (CNN). Consequently, four approaches namely BIM-Tracker, BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet are proposed in the thesis to reach the research objectives.

The proposed approaches are suitable for applications such as augmented reality and pedestrian navigation. Besides, these approaches are not prone to the accumulation of error and do not depend on the 3D image-based reconstruction of the indoor space for operation. The only requirement is a 3D building model that can be easily obtained from a building information model (BIM) that is commonly created and maintained by architects, construction engineers, and the facility managers. Therefore, the outcomes of the research point towards the use of infrastructure-independent visual localisation approaches that can be adapted by most of the ubiquitous devices, and are suitable for many mass-market applications. The following section outlines the main findings of the research and the respective achievement of the research objectives.

7.1 Achieved research objectives

The thesis fulfils the four research objectives and addresses the two challenges of the state-of-the-art visual localisation approaches. The first research objective of the thesis was to achieve centimetre level accuracy without any drift by integrating image information with a 3D building model. The objective is fulfilled with BIM-Tracker, where a drift-free visual localisation is performed by integrating image information with the 3D building model. The experiments with real data captured by a smartphone suggest that a localisation accuracy better than ten centimetres can be achieved by BIM-Tracker. Moreover, the estimated camera poses are suitable for augmented reality applications.

The second research objective was to perform camera pose regression of real images without performing a 3D image-based reconstruction of the indoor environment by using synthetic images obtained from the 3D building model. The objective is achieved with BIM-PoseNet where the requirement of 3D image-based reconstruction of indoor space is eliminated for camera pose regression approaches using CNNs, like PoseNet. The proposed BIM-PoseNet uses synthetic images generated from a 3D building model to perform camera pose regression with real images. It is identified that BIM-PoseNet can perform pose regression of real images by learning from synthetic edge gradmag images that do not contain any colour or texture, and the achievable accuracy is under two metres. In addition, it is identified that BIM-PoseNet can perform camera pose regression with low level-of-detail 3D building models, as well.

The third research objective was to model the uncertainty of camera pose estimates by BIM-PoseNet using a Bayesian CNN. The objective is fulfilled with Bayesian BIM-PoseNet that models the uncertainty of camera pose estimates with BIM-PoseNet. The results of the experiments suggest that the modelled uncertainties are correlated with the actual errors, and therefore the uncertainties provide confidence of the estimated camera pose. Besides, the best localisation results are obtained by learning from the synthetic edge gradmag images (similar to BIM-PoseNet), and accuracy of under 2 metres can be achieved.

The fourth research objective was to reduce the errors of pose regression with real images for the networks that learn from synthetic images, by exploiting the spatio-temporal

information of image sequences using recurrent CNNs. The objective is achieved with Recurrent BIM-PoseNet that reduces the errors of camera pose regression of BIM-PoseNet and Bayesian BIM-PoseNet by using the spatio-temporal information of synthetic image sequences. Results of the experiments indicate that a localisation accuracy under 1.6 metres can be achieved by Recurrent BIM-PoseNet using synthetic edge gradmag images (similar to BIM-PoseNet and Bayesian BIM-PoseNet). Besides, the modelled uncertainties of camera pose estimations show a correlation with the actual errors.

One of the major findings of the thesis is that the deep CNN pose regression networks learn more from structural information (edges) rather than texture. This information is repetitively validated with the experiments with BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet.

7.2 Limitations and future directions

The experiments with synthetic and real data in Chapter 3 reveal that BIM-Tracker is robust against moderate motion blur, occlusions and illumination variations. However, the main limitation of BIM-Tracker is the inability to recover the camera pose after a complete loss of tracking. The loss of tracking could be a result of extreme situations. For instance, BIM-tracker fails under severe motion blur that can be a result of a sudden rotation of the camera. Also, sharp rotations often result in the failure of tracking due to the wrong initialisation of the next frame. Another failure case arises when there are no visible building edges in the image due to the low field-of-view (FoV) of the camera, or when the camera is very close to the wall. Furthermore, the loss of tracking can result when the user takes the smartphone temporary for other purposes (e.g. making a call).

The future research directions for BIM-Tracker include the integration of approaches that can provide the initial camera pose in the event of a tracking loss, such as Recurrent BIM-PoseNet. Another possible future direction to mitigate the loss of tracking includes the integration of sensors present in the smartphones such as inertial measuring units (IMU) and using large FoV cameras. Additionally, one of the assumptions of BIM-Tracker is that the motion of the camera follows a constant velocity model, and therefore Kalman



Figure 7.1: An example of semantic segmentation (right image) of an indoor scene (left image), after [Zheng et al. \(2019\)](#).

filtering was used. In practice, this assumption might be valid for controlled robotic localisation applications, however, for pedestrian navigation this might not always be valid. Therefore, an additional research direction points towards the integration of extended Kalman filtering to improve the robustness of the proposed method. Furthermore, BIM-Tracker assumed that the camera intrinsic parameters are correctly obtained from the calibration process. However, calibration errors can affect the accuracy of BIM-Tracker and perhaps is another research direction.

Chapter 4 and 5 indicate that BIM-PoseNet and Bayesian BIM-PoseNet can learn from synthetic edge gradmag images that do not contain any texture or colour information, and can perform localisation with real images. However, the limitations of these approaches include the lack of precision of pose estimates, high variance of estimated distances between consecutive frames and high localisation errors (approximately two metres). The large errors are a result of artefacts, motion blur and perceptual aliasing. Chapter 6 proposes Recurrent BIM-PoseNet that uses spatio-temporal information of the synthetic edge gradmag images. This strategy reduces the effect of perceptual aliasing, and the median error is improved to 1.6 metres. However, the large errors of Recurrent BIM-PoseNet are due to motion blur and artefacts.

The other future directions include the use of semantic information from the BIM to generate segmented synthetic images which can be used to reduce the effects of the artefacts. Figure 7.1 shows an example of the use of segmented images. Next, in Chapter 4 it is demonstrated that BIM-PoseNet can perform localisation with a low level-of-detail

3D building model. Therefore, BIM-PoseNet can *potentially* work with 3D models derived from 2D floor plans by extruding the walls, and verifying the same can be a part of future work. Additionally, we have explored the localisation accuracies for BIMs having LoD 200 and LoD 300 in Chapter 4, and we leave the verification of the effects of LoD 350, LoD 400 and LoD 500 on the localisation accuracy as a part of future work.

Another future direction would be the use of constraints from the 3D building model so that the estimated camera poses are inside the navigable area of the indoor space. This constraint can be incorporated in the loss function that will be optimised during the fine-tuning process. Also, the localisation performance of *shallow* networks that operate real-time in smartphones and mobile platforms, such as SqueezePoseNet (Müller et al., 2017), can be evaluated as a part of future work.

The experiments performed in the thesis were conducted in a university corridor, and the future research can explore the applicability of all these methods in larger and less constrained indoor environments, such as large halls, or open spaces such as multi-storey shopping malls, or perhaps in outdoor environments. Additionally, for the scope of the research it was assumed that the BIM is a correct representation of the physical space, and there is no inconsistency (apart from measurement errors during creation of the BIM). However, in practice, due to the dynamic nature of the indoor spaces, this assumption might not always be valid. Although the proposed methods can handle this inconsistency to some extent, they might fail under certain situations, for instance in a construction site. Therefore, future experimentation is required to assess the performance of the proposed approaches in environments whose structures change with time.

7.3 Recommendation and applications

BIM-Tracker has localisation less than 10 centimetres, which makes it suitable for many applications, such as pedestrian navigation, augmented reality and autonomous robots, such as unmanned ground vehicles. For the ideal operation of BIM-Tracker, a medium field-of-view (approximately 90°) camera operating in a landscape orientation and capturing high-resolution images is recommended. Additionally, the 3D building model

that is used for the localisation needs to be accurate. Any discrepancy of the 3D building model with the indoor space will result in the artefacts in the image which can result in increased localisation errors or perhaps loss of tracking. Moreover, for a smooth operation of BIM-Tracker, the camera needs to be away from the walls to have a good view of the structural elements of the building.

BIM-PoseNet, Bayesian BIM-PoseNet and Recurrent BIM-PoseNet achieve localisation accuracy in the range of 1.6 – 2 metres, and can be used independently of BIM-Tracker. While this accuracy is suitable for pedestrian navigation, it is not suitable for autonomous robots. For ideal operation of these approaches, the 3D building model needs to be an accurate representation of the indoor space (similar to the case of BIM-Tracker). As the modelled uncertainty is an indication of the change in the scene, one of the interesting future applications using the uncertainty can be change detection. This detected change can perhaps be used to update the 3D building model automatically.

Regarding computational times, BIM-Tracker can be optimised to run on CPU in real-time by implementing in proper language (such as C++) and platform, and therefore can be utilised for robotic localisation in indoor spaces. Also, BIM-PoseNet and the related approaches can be optimised to run on real-time on platforms such as NVIDIA nano that contains on-board GPUs.

Bibliography

Bibliography

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- Acharya, D., Khoshelham, K., and Winter, S. Real-time detection and tracking of pedestrians in cctv images using a deep convolutional neural network. In *Proceedings of the 4th Annual Conference of Research@Locate*, volume 1913, pages 31–36, April 2017.
- Acharya, D., Yan, W., and Khoshelham, K. Real-time image-based parking occupancy detection using deep learning. In *Proceedings of the 5th Annual Conference of Research@Locate*, volume 2087, pages 33–40, April 2018.
- Acharya, D., Khoshelham, K., and Winter, S. BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150:245–258, 2019a.
- Acharya, D., Ramezani, M., Khoshelham, K., and Winter, S. BIM-Tracker: A model-based visual tracking approach for indoor localisation using a 3D building model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150:157–171, 2019b.
- Acharya, D., Singha Roy, S., Khoshelham, K., and Winter, S. Modelling uncertainty of single image indoor localisation using a 3d model and deep learning. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, IV-2/W5:247–254, 2019c.
- Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrani, S., Al-Ammar, M. A., and Al-Khalifa, H. S. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5):1–36, 2016. ISSN 1424-8220.

- Arandjelovic, R. and Zisserman, A. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 2911–2918, June 2012.
- Boochs, F., Schtze, R., Simon, C., Marzani, F., Wirth, H., and Meier, J. Increasing the accuracy of untaught robot positions by means of a multi-camera system. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–9, Sept 2010.
- Brahmbhatt, S., Gu, J., Kim, K., Hays, J., and Kautz, J. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2616–2625, June 2018.
- Briales, J. and Gonzalez-Jimenez, J. A minimal closed-form solution for the perspective three orthogonal angles (p3oa) problem: application to visual odometry. *Journal of Mathematical Imaging and Vision*, 55(3):266–283, 2016.
- Brki, B., Guillaume, S., Sorber, P., and Oesch, H. P. Daedalus: A versatile usable digital clip-on measuring system for total stations. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–10, Sept 2010.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, Dec 2016. ISSN 1552-3098.
- Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986. ISSN 0162-8828.
- Caron, G., Dame, A., and Marchand, E. Direct model based visual tracking and pose estimation using mutual information. *Image and Vision Computing*, 32(1):54 – 63, 2014. ISSN 0262-8856.
- Chen, H., Wang, X., Bai, H., Ros, S. J., and Plocher, T. Camera registration and video integration in 3d geometry model, June 14 2011. US Patent App. 14/122,324.

- Choi, C. and Christensen, H. I. Real-time 3d model-based tracking using edge and key-point features for robotic manipulation. In *2010 IEEE International Conference on Robotics and Automation*, pages 4048–4055, May 2010.
- Choi, C. and Christensen, H. I. 3d textureless object detection and tracking: An edge-based approach. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3877–3884, Oct 2012.
- Chollet, F. et al. Keras. <https://keras.io>, 2015.
- Clark, R., Wang, S., Markham, A., Trigoni, N., and 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)*, volume 3, July 2017.
- Comport, A. I., Marchand, E., and Chaumette, F. A real-time tracker for markerless augmented reality. In *Proceedings of the 2Nd IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '03*, pages 36–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2006-5.
- Costante, G., Mancini, M., Valigi, P., and Ciarfuglia, T. A. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE robotics and automation letters*, 1(1):18–25, 2016.
- David, P., DeMenthon, D., Duraiswami, R., and Samet, H. Simultaneous pose and correspondence determination using line features. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–424–II–431 vol.2, June 2003.
- David, P., DeMenthon, D., Duraiswami, R., and Samet, H. Softposit: Simultaneous pose and correspondence determination. In Heyden, A., Sparr, G., Nielsen, M., and Johansen, P., editors, *Computer Vision — ECCV 2002*, pages 698–714, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-47977-2.
- Davison, A. J. Real-time simultaneous localisation and mapping with a single camera.

- In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1403–1410 vol.2, Oct 2003.
- Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. *arXiv preprint arXiv:1710.07283*, 2017.
- Drummond, T. and Cipolla, R. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):932–946, July 2002. ISSN 0162-8828.
- Fischler, M. A. and Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6): 381–395, June 1981. ISSN 0001-0782.
- Fod, A., Howard, A., and Mataric, M. A. J. A laser-based people tracker. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 3024–3029, 2002.
- Furukawa, Y. and Ponce, J. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.
- Gal, Y. and Ghahramani, Z. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.
- Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016a.
- Gal, Y. and Ghahramani, Z. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016b.
- Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, Aug 2003. ISSN 0162-8828.

- Gee, A. P. and Mayol-Cuevas, W. Real-time model-based slam using line segments. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Nefian, A., Meenakshisundaram, G., Pascucci, V., Zara, J., Molineros, J., Theisel, H., and Malzbender, T., editors, *Advances in Visual Computing*, pages 354–363, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-48627-5.
- Glocker, B., Shotton, J., Criminisi, A., and Izadi, S. Real-time rgb-d camera relocalization via randomized ferns for keyframe encoding. *IEEE Transactions on Visualization and Computer Graphics*, 21(5):571–583, 2015.
- Goller, C. and Kuchler, A. Learning task-dependent distributed representations by back-propagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN’96)*, volume 1, pages 347–352. IEEE, 1996.
- Gomez-Ojeda, R. and Gonzalez-Jimenez, J. Robust stereo visual odometry through a probabilistic combination of points and line segments. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2521–2526, May 2016.
- Gomez-Ojeda, R. and Gonzalez-Jimenez, J. Geometric-based line segment tracking for hdr stereo sequences. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 69–74, Oct 2018.
- Gomez-Ojeda, R., Moreno, F. A., Scaramuzza, D., and Jiménez, J. G. PL-SLAM: a stereo SLAM system through the combination of points and line segments. *CoRR*, abs/1705.09479, 2017.
- Graves, A. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.
- Groves, P. D. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.
- Gu, F., Hu, X., Ramezani, M., Acharya, D., Khoshelham, K., Valaee, S., and Shang, J. Indoor localization improved by spatial context - a survey. *ACM Computing Surveys*, 52(3):64:1–35, 2019.

- Ha, I., Kim, H., Park, S., and Kim, H. Image retrieval using bim and features from pre-trained vgg network for indoor localization. *Building and Environment*, 140:23 – 31, 2018.
- Handa, A., Whelan, T., McDonald, J., and Davison, A. J. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, May 2014.
- Harle, R. A survey of indoor inertial positioning systems for pedestrians. *The IEEE Communications Surveys Tutorials*, 15(3):1281–1293, 2013.
- Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.
- Hofer, M., Wendel, A., and Bischof, H. Line-based 3d reconstruction of wiry objects. In *18th Computer Vision Winter Workshop*, pages 78–85, 2013.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., and Roy, N. Visual odometry and mapping for autonomous flight using an rgb-d camera. In Christensen, H. I. and Khatib, O., editors, *Robotics Research : The 15th International Symposium ISRR*, pages 235–252. Springer International Publishing, Cham, 2017. ISBN 978-3-319-29363-9.
- Irschara, A., Zach, C., Frahm, J. M., and 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)*, pages 2599–2606, June 2009.
- Isikdag, U., Zlatanova, S., and Underwood, J. A bim-oriented model for supporting indoor navigation requirements. *Computers, Environment and Urban Systems*, 41:112–123, 2013.

- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 675–678, 2014.
- Jose Tarrio, J. and Pedre, S. Realtime edge-based visual odometry for a monocular camera. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- Kendall, A. and Cipolla, R. Modelling uncertainty in deep learning for camera relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4762–4769, May 2016.
- Kendall, A., Grimes, M., and Cipolla, R. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, December 2015.
- Kendall, A., Cipolla, R., et al. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 8, July 2017.
- Khoshelham, K. and Ramezani, M. Vehicle positioning in the absence of gnss signals: Potential of visual-inertial odometry. In *2017 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4, March 2017.
- Khoshelham, K., Vilariño, L. D., Peter, M., Kang, Z., and Acharya, D. The isprs benchmark on indoor modelling. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42, 2017.
- Khoshelham, K., Tran, H., Díaz-Vilariño, L., Peter, M., Kang, Z., and Acharya, D. An evaluation framework for benchmarking indoor modelling methods. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4: 297–302, 2018.
- Khoshelham, K. Closed-form solutions for estimating a rigid motion from plane correspondences extracted from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:78 – 91, 2016. ISSN 0924-2716.

- Kitanov, A., Bisevac, S., and Petrovic, I. Mobile robot self-localization in complex indoor environments using monocular vision and 3d model. In *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, pages 1–6. IEEE, 2007.
- Koletschka, T., Puig, L., and Daniilidis, K. Mevo: Multi-environment stereo visual odometry. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4981–4988, Sep. 2014.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in neural information processing systems (NIPS)*, pages 1097–1105, Dec 2012.
- Kropp, C., Koch, C., Brilakis, J., and König, M. A framework for automated delay prediction of finishing works using video data and bim-based construction simulation. In *Proc. of the 14th International Conference on Computing in Civil and Building Engineering*, 2012.
- Kumar, A. K. T. R., Schufele, B., Becker, D., Sawade, O., and Radusch, I. Indoor localization of vehicles using deep learning. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, June 2016.
- Kwon, Y., Won, J.-H., Kim, B. J., and Paik, M. C. Uncertainty quantification using bayesian neural networks in classification: Application to ischemic stroke lesion segmentation. *arXiv preprint arXiv:1506.02158v6*, 2018.
- Lepetit, V. and Fua, P. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005. ISSN 1572-2740.
- Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., and Milford, M. J. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

- Lymberopoulos, D. and Liu, J. The microsoft indoor localization competition: Experiences and lessons learned. *IEEE Signal Processing Magazine*, 34(5):125–140, Sept 2017. ISSN 1053-5888.
- MacKay, D. J. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.
- Marchand, E., Bouthemy, P., and Chaumette, F. A 2d3d model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941 – 955, 2001. ISSN 0262-8856.
- Massiceti, D., Krull, A., Brachmann, E., Rother, C., and Torr, P. H. S. Random forests versus neural networks - what’s best for camera localization? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5118–5125, May 2017.
- Mautz, R. Indoor positioning technologies, 2012.
- Melekhov, I., Ylioinas, J., Kannala, J., and Rahtu, E. Image-based localization using hour-glass networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 879–886, Oct 2017a.
- Melekhov, I., Ylioinas, J., Kannala, J., and Rahtu, E. Relative camera pose estimation using convolutional neural networks. In *Proceedings of the Advanced Concepts for Intelligent Vision Systems*, pages 675–687, Feb 2017b.
- Müller, M., Urban, S., and Jutzi, B. Squeezeposenet: Image based pose regression with small convolutional neural networks for real time uas navigation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:49, 2017.
- Mur-Artal, R., Montiel, J. M. M., and Tards, J. D. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015. ISSN 1552-3098.
- Mur-Artal, R. and Tardós, J. D. Fast relocalisation and loop closing in keyframe-based slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 846–853. IEEE, 2014.

- Muralidharan, K., Khan, A. J., Misra, A., Balan, R. K., and Agarwal, S. Barometric phone sensors: More hype than hope! In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, HotMobile '14, pages 12:1–12:6, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2742-8.
- Nister, D., Naroditsky, O., and Bergen, J. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–652–I–659 Vol.1, June 2004.
- Parisotto, E., Chaplot, D. S., Zhang, J., and Salakhutdinov, R. Global pose estimation with an attention-based recurrent network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 350–359, June 2018.
- Peng, X., Sun, B., Ali, K., and Saenko, K. Learning deep object detectors from 3d models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1278–1286, 2015.
- Piasco, N., Sidib, D., Demonceaux, C., and Gouet-Brunet, V. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74:90 – 109, 2018.
- Porwal, A. and Hewage, K. N. Building information modeling (bim) partnering framework for public construction projects. *Automation in Construction*, 31:204–214, 2013.
- Qiu, K., Liu, T., and Shen, S. Model-based global localization for aerial robots using edge alignment. *IEEE Robotics and Automation Letters*, 2(3):1256–1263, July 2017. ISSN 2377-3766.
- Radenović, F., Tolias, G., and Chum, O. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–20, Sept 2016.
- Ramezani, M. and Khoshelham, K. Vehicle positioning in gnss-deprived urban areas by stereo visual-inertial odometry. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2018. ISSN 2379-8858.

- Ramezani, M., Acharya, D., Gu, F., and Khoshelham, K. Indoor positioning by visual-inertial odometry. In *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, volume 4, 2017.
- Ramezani, M., Khoshelham, K., and Fraser, C. Pose estimation by omnidirectional visual-inertial odometry. *Robotics and Autonomous Systems*, 105:26 – 37, 2018. ISSN 0921-8890.
- Reitmayr, G. and Drummond, T. Going out: Robust model-based tracking for outdoor augmented reality. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '06*, pages 109–118, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 1-4244-0650-1.
- Santos, D. R. D., Basso, M. A., Khoshelham, K., de Oliveira, E., Pavan, N. L., and Vosselman, G. Mapping indoor spaces by adaptive coarse-to-fine registration of rgb-d data. *IEEE Geoscience and Remote Sensing Letters*, 13(2):262–266, 2016.
- Scaramuzza, D. and Fraundorfer, F. Visual odometry [tutorial]. *IEEE Robotics Automation Magazine*, 18(4):80–92, Dec 2011. ISSN 1070-9932.
- Sehgal, A. and Kehtarnavaz, N. Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time apps. *Machine Learning and Knowledge Extraction*, 1(1):450–465, 2019.
- Serant, D., Julien, O., Ries, L., Thevenon, P., Dervin, M., and Hein, G. The digital TV case - Positioning using signals-of-opportunity based on OFDM modulation. *Inside GNSS*, 6(6):pp 54–62, November 2011.
- Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., and 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)*, pages 2930–2937, June 2013.
- Silveira, G., Malis, E., and Rives, P. The efficient e-3d visual servoing. *International Journal of Optomechatronics*, 2(3):166–184, 2008.

- Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–8, May 2013.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- Su, H., Qi, C. R., Li, Y., and Guibas, L. J. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2686 – 2694, December 2015.
- Svarm, L., Enqvist, O., Kahl, F., and Oskarsson, M. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1455–1461, 2017.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.
- Taneja, S., Akcamete, A., Akinci, B., Garrett Jr, J. H., Soibelman, L., and East, E. W. Analysis of three indoor localization technologies for supporting operations and maintenance field tasks. *Journal of Computing in Civil Engineering*, 26(6):708–719, 2011.
- Tarzia, S. P., Dinda, P. A., Dick, R. P., and Memik, G. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys '11*, pages 155–168, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0643-0.
- Teulire, C., Marchand, E., and Eck, L. 3-d model-based tracking for uav indoor localization. *IEEE Transactions on Cybernetics*, 45(5):869–879, May 2015. ISSN 2168-2267.
- Torr, P. and Zisserman, A. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138 – 156, 2000. ISSN 1077-3142.

- Tran, H., Khoshelham, K., and Kealy, A. Geometric comparison and quality evaluation of 3d models of indoor environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:29–39, 2019.
- Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. Bundle adjustment — a modern synthesis. In Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-44480-0.
- Vacchetti, L., Lepetit, V., and Fua, P. Combining edge and texture information for real-time accurate 3d camera tracking. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '04*, pages 48–57, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2191-6.
- Valada, A., Radwan, N., and Burgard, W. Deep auxiliary learning for visual localization and odometry. In *International Conference on Robotics and Automation (ICRA 2018)*. IEEE, 2018a.
- Valada, A., Radwan, N., and Burgard, W. Incorporating semantic and geometric priors in deep pose regression. In *Proceedings of the Workshop on Learning and Inference in Robotics: Integrating Structure, Priors and Models at Robotics: Science and Systems (RSS)*, 2018b.
- Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 4627–4635. IEEE, 2017.
- Volk, R., Stengel, J., and Schultmann, F. Building information modeling (bim) for existing buildings literature review and future needs. *Automation in construction*, 38:109–127, 2014.
- Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., and Cremers, D. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 627–637, Oct 2017.

- Wald, I., Boulos, S., and Shirley, P. Ray tracing deformable scenes using dynamic bounding volume hierarchies. *ACM Trans. Graph.*, 26(1), January 2007. ISSN 0730-0301.
- Wang, S., Clark, R., Wen, H., and Trigoni, N. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050, May 2017a.
- Wang, S., Wang, Y., Tang, J., Shu, K., Ranganath, S., and Liu, H. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, pages 391–400. International World Wide Web Conferences Steering Committee, 2017b.
- Witt, J. and Weltin, U. Robust stereo visual odometry using iterative closest multiple lines. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4171, Nov 2013.
- Wu, J., Ma, L., and Hu, X. Delving deeper into convolutional neural networks for camera relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651, May 2017.
- Wuest, H., Vial, F., and Stricker, D. Adaptive line tracking with multiple hypotheses for augmented reality. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '05*, pages 62–69, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2459-1.
- Xie, H., Gu, T., Tao, X., Ye, H., and Lv, J. Maloc: A practical magnetic fingerprinting approach to indoor localization using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14*, pages 243–253, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2968-2.
- Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., and Aboutanios, E. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Communications Surveys Tutorials*, 19(2):1327–1346, Secondquarter 2017.

- Ye, H., Gu, T., Tao, X., and Lu, J. B-loc: Scalable floor localization using barometer on smartphone. In *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, pages 127–135, Oct 2014.
- Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006. ISSN 0360-0300.
- Zhang, L. and Koch, R. Structure and motion from line correspondences: Representation, projection, initialization and sparse bundle adjustment. *Journal of Visual Communication and Image Representation*, 25(5):904 – 915, 2014. ISSN 1047-3203.
- Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., and Zhou, Z. Structured3d: A large photo-realistic dataset for structured 3d modeling. *CoRR*, abs/1908.00222, 2019.
- Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.
- Zhu, L. and Laptev, N. Deep and confident prediction for time series at uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110. IEEE, 2017.
- Zhu, Z., Oskiper, T., Samarasekera, S., Kumar, R., and Sawhney, H. S. Ten-fold improvement in visual odometry using landmark matching. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, pages 1–8, Oct 2007.
- Zisserman, M. A. A. Robust object tracking. In *Asian Conference on Computer Vision*, volume 1, pages 58–61, 1995.