A Monocular Indoor Localiser based on an Extended Kalman Filter and Edge Images from a Convolutional Neural Network

James Unicomb[1], Ravindra Ranasinghe*[1], Lakshitha Dantanarayana[1], and Gamini Dissanayake[1]

Abstract— **The main contribution of this paper is an extended Kalman filter (EKF) based algorithm for estimating the 6 DOF pose of a camera using monocular images of an indoor environment. In contrast to popular visual simultaneous localisation and mapping algorithms, the technique proposed relies on a pre-built map represented as an unsigned distance function of the ground plane edges. Images from the camera are processed using a Convolutional Neural Network (CNN) to extract a ground plane edge image. Pixels that belong to these edges are used in the observation equation of the EKF to estimate the camera location. Use of the CNN makes it possible to extract ground plane edges under significant changes to scene illumination. The EKF framework lends itself to use of a suitable motion model, fusing information from any other sensors such as wheel encoders or inertial measurement units, if available, and rejecting spurious observations. A series of experiments are presented to demonstrate the effectiveness of the proposed technique.**

I. INTRODUCTION

Localisation, or determining the pose (position and orientation) of an autonomous device or a person within a given map is a fundamental requirement to deliver a wide range of indoor and outdoor location based services. In the case of a robot, it is an indispensable requirement to support autonomous operations in an environment. In this paper, we use the term “robot” to refer to a device like a smartphone carried by a person, a drone, an actual autonomous device, or a robot. The prevalent use of smartphones equipped with receivers for global positioning systems (GPS) allows a wide range of outdoor location based services in many fields, however, the accuracy of the estimated location is significantly reduced in some urban areas and GPS is ineffective in indoor environments [1].

Accurate localisation in indoor environments is as important as outdoor environments. Typical indoor services that rely on localisation, such as targeted advertising in a shopping mall expects higher precision and the position estimation error is not to exceed a few metres to allow differentiation between floors and nearby rooms [2]. A reliable indoor localisation package would be one of the key elements to stimulate broad adaptation of assistive robots in pursuit of finding acceptable solutions to deliver efficient services to support aging populations [1], [3].

Given the prevalence of low-cost cameras, relating information captured in an image to an associated map, a spatial model that represents the physical environment in which a robot operates, is one of the most cost-effective approaches to localisation in indoor environments. Existing techniques for localisation based on monocular images rely on bearing only simultaneous localisation an mapping (SLAM) [4]. These techniques map the environment by estimating the locations of point features extracted from a camera image while at the same time estimate the camera location. As image features do not persist over long time-scales, mainly as a result of illumination changes, a pre-defined map of such features cannot be used for localisation. Therefore, SLAM is a must in vision based localisation making these approaches computationally heavy. In this paper, we present a novel method for localisation in indoor environments that rely on a map of edges present in the ground plane. Images from a camera are processed to obtain edges that are on the ground plane. These are then used as observations in an extended Kalman filter (EKF) to generate the full six degrees of freedom (DOF) estimate of the camera pose.

We use a Distance Function (DF) based map representation to generate the required observation equations. During the past few years, DF based maps, that use a measure of the distance to the closest occupied space to represent the geometry of the environment, have gained traction in the robotics community. In DF, geometry is not stored explicitly but rather defined as a level set of a function defined over the space in which the geometry is embedded. Therefore, these maps not only encode the occupied regions of the environment but also provide a continuous measure of the distance, a much richer representation of the environment. In [5], [6], [7] authors have demonstrated the use of different formulations of distance functions for localising robots in both indoor and outdoor environments.

In this work, we focus on solving the complete 6-DOF localisation problem to estimate the camera pose given a map of the environment using either a hand-held camera (for example camera of a smart phone) or a body mounted camera. We, however, do not use explicit matching of visual point features in the environment to estimate the state. We segment the ground and ground level edges to extract a binary edge image of the environment using Convolution Neural Network (CNN) based novel approach which in turn is used to generate the observation vector. The ability of the CNN to robustly detect the edges under a broad range of conditions and the availability of sound statistical methods for data association within the EKF leads to an algorithm that makes reliable pose estimates with uncertainty. Furthermore, it facilitates traditional fusion of other sensors such as odometry or inertial measurements where available.

The localisation strategy proposed is as follows. We use

[1] Centre for Autonomous Systems, University of Technology Sydney, Australia.

* Corresponding Author: ravindra.ranasinghe@uts.edu.au

a DF based approach discussed in [5], [6] for representing the map of the environment. This representation is achieved by first obtaining a binary ground plane edge map of the environment and computing its unsigned distance function. The mapping phase uses an RGB-D camera with ORB-SLAM2 [4] to obtain the camera poses while traversing the target environment. In order to generate the binary edge map, camera images are processed through a CNN to generate the corresponding ground plane edge images which are subsequently assembled into a map using the known camera poses. We approximate the DF and their first and second derivatives using cubic splines to obtain an efficient and compact form of the representation. While computing DF and its derivatives are processing intensive, this is a one-off operation that is carried out prior to localisation.

During the run-time localisation operation, we extract ground plane edges from the image captured by a monocular camera using a CNN. We use the condition that the sum of squared distance function values at each edge pixel when superimposed on the DF map is zero when there is no misalignment between the predicted and actual camera pose in an EKF framework to compute a 6-DOF pose estimate. Camera pose can in turn be transformed into the robot pose when the mounting location is known.

The paper is organised as follows. Sec. II reviews the related literature. Sec. III provides the novel CNN based edge extraction approach proposed in this paper. Sec. IV details the DF based EKF framework for localisation. Experimental results presented in Sec. V demonstrate the merits of the proposed scheme using multiple data sets. Sec. VI provides a summary of the contributions of this paper and presents concluding remarks.

II. Related Work

This section provides a summary of indoor localisation methods proposed in the literature for use with a hand-held device such as a smart phone or wearable camera.

There have been many indoor localisation methods that exploits various attributes of Wi-Fi signals. The work presented in [8], [9], [10], [11], [3] are typical examples that use signal propagation characteristics or Angle of Arrival of the Wi-Fi signal or Wi-Fi signal strength (RSSI) based fingerprinting techniques for indoor localisation. Both signal propagation and Angle of Arrival based methods require the prior knowledge of Wi-Fi access points that are deployed in indoor environment. For fingerprinting approaches, a comprehensive model of RSSI distribution for the indoor environment must be built.

Ultra-wideband (UWB) based radio systems that use time of flight based localisation has recently gained attention due to its relatively high accurate estimates [12]. However, UWB based systems require the installation of specialized hardware modules in the environment. In [13], Rimminen et al. used RFID tags attached to subjects to localise them in an indoor environment.

Machine vision based methods are popular alternative for indoor localisation. Vision is one of the most important types of sensing which is extensively used in indoor navigation[14]. In [14], an appearance-based information was used to achieve indoor localisation. Many existing methods have utilized Monte Carlo localaisation with cameras [15], [16]. There are few other vision based localisation methods that use the direct image matching approach for localisation [17]. In [18] authors have formulated the problem of finding the location of a robot as a regression problem where a continuous pose is predicted from an input image.

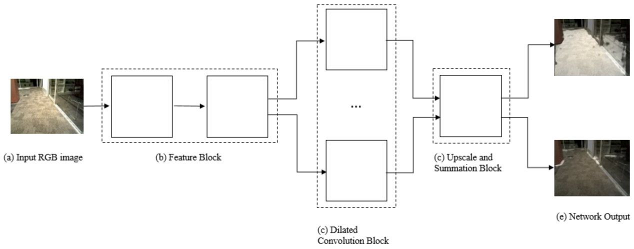


Fig. 1: CNN model to extract coarse edge image

III. CNN based edge extraction

CNNs are deep learning models that take advantage of the spatial structure of 2D images [19] to learn rich representation which has been used for image based localisation.

In this work, we aim to use RGB images from a hand-held monocular camera to segment ground edges by using the segmentation of the ground plane as an auxiliary problem and hence to obtain coarse binary edge images. The premise here is that the geometric information CNN uses in predicting the floor helps the network to learn the floor edges.

Once a coarse edge image is available from the network, an edge image with fine scaled edges can be extracted by taking the intersection of the network generated coarse edge image and the Canny edge detector generated image using the original image. The resulting image with fine scale edges that belong to the ground plane is the input to the DF based EKF framework (Sec. IV) to accomplish localisation.

A. *Network Architecture*

Adapted from the recent work of Google DeepLabv3 [20], our model of network structure, as shown in Fig. 1, has dilated convolutions after a feature block to extract the different spatial features.

1) Feature Extraction: Two convolutions with rectifier non-linearities and $3 \times 3 \times 32$ filters are used to construct a feature map from the image. Batch normalisation is applied before each non-linearity.

2) Dilated Convolutions: Dilated convolutions (also known as Atrous convolution in DeepLab [20]) take into account a greater field of view by adding spaces between each filter. A standard convolution represents a dilated convolution of rate $= 1$. Using the feature map, multiple scales of the feature map can be learned.

In our network, for each dilated convolution, there are 32 filters of size 3×3. The rates of each convolution are $(1, 2, 4, 8)$ with strides $(1, 1, 2, 4)$.

3) Upsampling and Summation: Due to increasing strides, repeated upscaling is applied to match the dimensions when passing into the summation block.

B. Generation of Floor and Edge Images for Network Training

An annotated data-set with ground truth was obtained using an RGB-D camera. With each RGB image, we take the corresponding depth image to segment planes and take the floor using a pass-through xyz-filter and the approximate robot pose (hence the pose of the camera). As this method is prone to errors such as larger vertical planes being accepted rather than the floor, the dataset was manually corrected by deleting the images corresponding to false floor measurements.

Once the floor is segmented, we look at the edge regions, and take the local mean depth of the floor, if the local mean depth of the edge region corresponds to the local mean of the floor then the edge region is assumed to belong to the ground.

C. Input to the network

We have an RGB image, $\mathsf{X} \in [0,1]^{M\times N\times 3}$ which can be mapped with a CNN to produce two single channelled images being the segmented floor and floor edges, represented as $\mathsf{Y}, \mathsf{Z} \in [0,1]^{M\times N}$ respectively.

The input is down-sampled from 480×640 to 120×160. The output dimensions are set to 120×160.

D. Output of the network

After summing the dilated and up-sampled convolutions, a final set of convolutions are applied to the summation block with a sigmoid output assigning the probability of the segmented floor and edge regions.

The output is clipped with a small constant, $\epsilon = 10^{-4}$, to avoid saturation and NaN values during training.

E. Network Loss

We use mean pixel-wise binary cross entropy (BCE) on the sigmoid output of the network and the labelled ground-truth.

$$L(\hat{\mathsf{Y}}, \mathsf{Y}; \mathsf{X}) = \frac{1}{MN}\sum_{i}^{M}\sum_{j}^{N} -y_{ij}\ln(\hat{y}_{ij}) - (1-y_{ij})\ln(1-\hat{y}_{ij})$$

Therefore, the objective is to minimize the additive loss w.r.t. the network parameters:

$$L(\hat{\mathsf{Y}}, \mathsf{Y}; \mathsf{X}) + L(\hat{\mathsf{Z}}, \mathsf{Z}; \mathsf{X})$$

F. Evaluation Metric

As a qualitative comparison for the network, we use intersection over union (IoU):

$$\frac{|\hat{\mathsf{Y}} \cap \mathsf{Y}|}{|\hat{\mathsf{Y}} \cup \mathsf{Y}|} \quad \text{and} \quad \frac{|\hat{\mathsf{Z}} \cap \mathsf{Z}|}{|\hat{\mathsf{Z}} \cup \mathsf{Z}|}$$

The IoU metric quantifies the number of false-positives and true-negatives when comparing the network output to ground truth.

G. Training

Adam optimization [21] is used with a poly-decaying learning rate ranging from 0.01 to 0.0005. The first and second momentum values were 0.9 and 0.999 respectively.

H. Regularization

Drop-out is applied after each feature block so that the test error matches training error at an appropriate level to prevent overfitting.

Fig. 2 shows five sample inputs RGB images, corresponding network results on edge region detection, network segmented floor images and the outputs of Canny edge detector on RGB images augmented with the results on edge region detection. The red pixels marked on the Fig. 2(d) belong to the ground plane, i.e. the set of common pixels points from the intersection operation between Fig. 2(b) and the output from the Canny edge detector.

Furthermore, the CNN can be shown to be robust to lighting changes as depicted in Fig. 3.

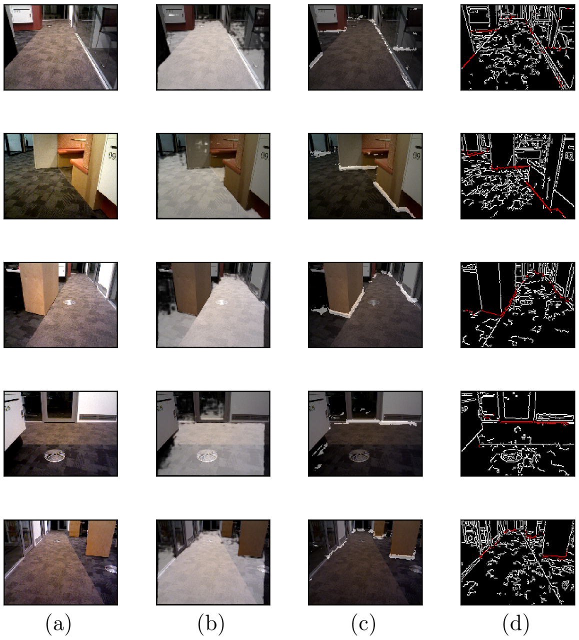


Fig. 2: (a) RGB image used as input into the network, (b) network result on floor segmentation, (c) network result on edge region detection, (d) canny edge detection on RGB image with edge region to find edges, taking the intersection of the sets returns edge pixels (red) belonging to the ground plane.

IV. Localisation Framework

Finding the location in space that best describes the observations from a sensor mounted on a robot is the objective of robot localisation algorithm. This typically requires an observation equation that computes the expected sensor observations given as a function of the robot location and the map. Then the objective is to find the robot location that

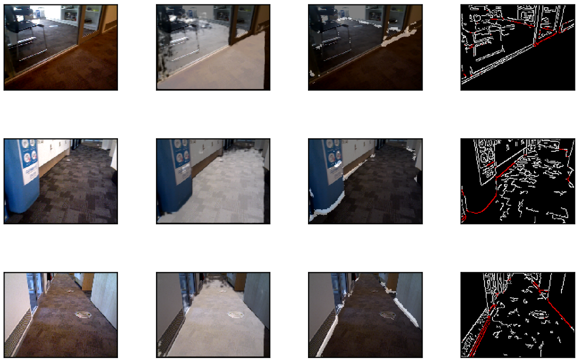

(a) (b) (c) (d)

Fig. 3: Extraction of ground features by the CNN under different lighting conditions. (a) RGB image, (b) network result on floor segmentation, (c) network result on edge region detection, (d) canny edge detection result on RGB image.

minimises the difference between the actual and expected sensor observations.

The following subsections illustrate the use of an EKF for robot localisation in an environment represented using DF.

A. Distance Function Maps

Robot's operating environment is represented using a DF map which is generated using a function that captures the distance from a given location to the closest occupied space. When the width of occupied regions of the environment are similar in size to the resolution of the sensors, for example, in the case of an indoor environment with thin walls, use of the unsigned distance function is more convenient. In this form of DF, the absolute value of the distance from a point to the nearest boundary is assigned to the DF [6]. When V is the set of occupied grid cells in an occupancy grid map, the DF can be expressed by (1) at any given point $\boldsymbol{x}_{\boldsymbol{z}_i}$.

$$d_{DF_i} = DF(\boldsymbol{x}_{\boldsymbol{z}_i}) = \min_{\boldsymbol{v}_j \in V} \|\boldsymbol{x}_{\boldsymbol{z}_i} - \boldsymbol{v}_j\| \tag{1}$$

Mullen et al. [22] have demonstrated that unsigned distance function variant is more robust to outliers and noise.

B. Observation Model

Observation model relates sensor readings and robot location to the map. In this work, we use a camera (either a hand-held or a physically mounted camera on the robot) as our sensor. Using the camera images and the CNN method described in Sec. III, we extract the corresponding binary edge images.

The observation vector $\boldsymbol{z}$, of a single binary edge image, consisting of n edge points with coordinates $(\lambda_i,\ \mu_i) = \boldsymbol{z}_i \in \boldsymbol{z}$ on the image plane can be projected from the current estimate of the 6-DOF robot pose $\boldsymbol{x} = (x,\ y,\ z(\text{altitude}),\ \psi(\text{roll}),\ \theta(\text{pitch}),\ \phi(\text{yaw}))^\top$, using (2) to obtain the observation vector in 2D space $\boldsymbol{x}_{\boldsymbol{z}_i}$ on the ground plane.

$$\begin{aligned} m\left(\boldsymbol{x}_{k+1|k}, \boldsymbol{z}_i\right) &= \boldsymbol{x}_{\boldsymbol{z}_i} \\ &= \begin{bmatrix} x_{k+1|k} \\ y_{k+1|k} \end{bmatrix} - \frac{z_{k+1|k}}{R_{3,1}\lambda_i + R_{3,2}\mu_i + R_{3,3}f} \\ &\quad \cdot \begin{bmatrix} R_{1,1}\lambda_i + R_{1,2}\mu_i + R_{1,3}f \\ R_{2,1}\lambda_i + R_{2,2}\mu_i + R_{2,3}f \end{bmatrix} \end{aligned} \tag{2}$$

where, R is the 3D rotation matrix representing the robot pose, $R(\psi,\ \theta,\ \phi)$, and f is the focal length of the camera.

Given a DF based map of the environment, it is now possible to obtain a measure for the "disparity" between expected and observed sensor observations by extracting the value of the DF at locations $\boldsymbol{x}_{\boldsymbol{z}_i}$ [5].

$$\mathbf{d}_{DF} = DF(\boldsymbol{z} \mid \boldsymbol{x}) = \begin{bmatrix} DF(\boldsymbol{x}_{\boldsymbol{z}_1}) \\ \vdots \\ DF(\boldsymbol{x}_{\boldsymbol{z}_i}) \\ \vdots \\ DF(\boldsymbol{x}_{\boldsymbol{z}_n}) \end{bmatrix} \tag{3}$$

The observation model detailed above is written as a function of the robot state $\boldsymbol{x}$ and expected sensor observations $\boldsymbol{z}$ as in (4). Enforcing the condition that the sum of squared distance function values at these points is expected to be zero when there is no misalignment results in the observation equation that is suitable for robot localisation in an EKF framework.

$$h(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{n} DF(\boldsymbol{x}_{\boldsymbol{z}_i})^2 \tag{4}$$

The formulation of the EKF is done in a similar manner to our previous work proposed in [5]. The state vector $\boldsymbol{x}$ consists of 12 dimensions, namely, the 6-DOF robot pose, linear velocities ν, and angular velocities ω; $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} & \boldsymbol{\theta} & \nu & \omega \end{bmatrix}^\top$.

We use a constant velocity model similar to the model proposed by Davison et al. [23] for motion prediction. With each image, the CNN in combination with Canny edge detector produces is a set of n pixels as the observation vector $\boldsymbol{z} = \{\boldsymbol{z}_i\}_{i=1,\cdots,n}$.

Assuming that each pixel measurement is corrupted in each direction with noise $\mathcal{N}(0, \sigma^2)$ which leads to $\Sigma_{\boldsymbol{z}} = \text{diag}(\sigma^2, \sigma^2)$.

We use *an innovation gate* at runtime to reject outliers by placing an upper bound on (5).

$$\frac{DF(\boldsymbol{x}_{\boldsymbol{z}_i})}{\sqrt{\nabla DF_{(x,y)} \left(\nabla m_{\boldsymbol{x}} P_{k+1|k} \nabla m_{\boldsymbol{x}}^\top + \nabla m_{\boldsymbol{z}} \Sigma_{\boldsymbol{z}} \nabla m_{\boldsymbol{z}}^\top\right) \nabla DF_{(x,y)}^\top}} \tag{5}$$

where $\nabla m_{\boldsymbol{x}}$ and $\nabla m_{\boldsymbol{z}}$ are the Jacobians of the measurement equation (2).

The the measurement covariance $\nabla m_{\boldsymbol{x}} P_{k+1|k} \nabla m_{\boldsymbol{x}}^\top + \nabla m_{\boldsymbol{z}} \Sigma_{\boldsymbol{z}} \nabla m_{\boldsymbol{z}}^\top$, can be obtained by projecting the expected covariance through the measurement equation. Then the variance in the direction of the closest edge point is found by multiplying the covariance matrix by the Jacobian of the DF w.r.t. the vector $(x,\ y)^\top$ to calculate (5). We then take the Z-score and reject outliers based on some level of uncertainty.

TABLE I: Mean squared error values of the estimates for each dataset.

Dataset	$x(m^2)$	$y(m^2)$	$z(m^2)$	roll(rad^2)	pitch(rad^2)	yaw(rad^2)
(i)	0.151	0.133	0.156	0.047	0.061	0.145
(ii)	0.179	0.149	0.130	0.040	0.050	0.121
(iii)	0.195	0.218	-	-	-	0.067

V. Experimental Results

We present experiments conducted using three datasets to evaluate the proposed algorithm. These datasets have been collected in three different environments: i) domestic environment, ii) office environment, and iii) laboratory environment. The authors collected the first two datasets. For the third dataset, we use the publicly available PUT RGB-D dataset [24].

In dataset (i) and (ii), a hand-held Asus Xtion pro$^{\text{TM}}$ camera is walked along in multiple loops in the respective environments to collect RGB-D data. The ground truth is obtained via ORB-SLAM2.

In dataset (iii), the authors in [24] used a small rover equipped with an RGB-D camera to survey a laboratory environment. This dataset contains multiple loops recorded in the said environment. The ground truth is provided through a motion capture system.

For the first two datasets, the initial run is used to generate the poses using ORB-SLAM2, which has been used to calculate the unsigned distance function map of the environment.

In these experiments, we use a 1σ innovation gate to reject outliers. A manual observation shows that most of such outliers are objects in the scene that was not present while the map was created. Furthermore, shadows and patches of light that appear in the scene due to the time of day and different artificial lighting also gets filtered during this stage.

Fig. 4 shows the 6-DOF trajectories of the camera in the three datasets and the ground truth is projected to the ground plane for ease of illustration. Fig. 5 shows the error plots for the three datasets, compared to the $\pm 2\sigma$ uncertainty bounds. For the dataset (iii), ground-truth in z, pitch, and roll directions are not available in the dataset. The robot travels approximately 30m, 60m, and 50m respectively in the three datasets.

It can be seen that the estimated trajectory closely follows the ground truth. Furthermore, the error values are generally within the $\pm 2\sigma$ uncertainty. Table I shows the mean squared error values for each dataset.

The experiments were conducted on a computer equipped with an Intel Core® i7-4578U 3.0Ghz processor. Average time taken to estimate a single pose is $98.5ms$ while $5.1ms$ is used for CNN based edge extraction. The code is written using Python v2.7, and we use Theano v1.0.0 [25] and Lasagne v0.2 [26] for implementation of the CNN.

VI. Conclusion

In this paper, we presented an algorithm to estimate the 6-DOF location of a monocular camera with respect to the ground plane map of an indoor environment. When an image is captured, we use a CNN to extract only the edges that belong to the ground plane from the image, which is used for pose estimation within an EKF framework against an unsigned distance function of the environment map, with the initial location approximately known.

This formulation relaxes the assumption in our previous work on outdoor localisation over flat terrain [5] that relied on the ground plane to not contain any 3D objects, which is violated in indoor environments.

The results from multiple datasets show that the estimated trajectory closely follows the ground truth.

Making the algorithm available as a readily usable package and integrating a method to solve the “kidnapped robot problem” which exempts the requirement of knowing the approximate initial location is planned for future work.

References

[1] M. Gillham, G. Howells, S. Spurgeon, and B. McElroy, “Floor covering and surface identification for assistive mobile robotic real-time room localization application,” *Sensors*, vol. 13, no. 12, pp. 17 501–17 515, 2013.

[2] M. Werner, M. Kessel, and C. Marouane, “Indoor positioning using smartphone camera,” in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*. IEEE, 2011, pp. 1–6.

[3] A. Perera, J. Arukgoda, R. Ranasinghe, and G. Dissanayake, “Localization system for carers to track elderly people in visits to a crowded shopping mall,” in *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on*. IEEE, 2017, pp. 1–8.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] J. Unicomb, L. Dantanarayana, J. Arukgoda, R. Ranasinghe, G. Dissanayake, and T. Furukawa, “Distance Function based 6DOF Localization for Unmanned Aerial Vehicles in GPS Denied Environments,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 09 2017.

[6] R. Ranasinghe, G. Dissanayake, T. Furukawa, J. Arukgoda, and L. Dantanarayana, “Environment representation for mobile robot localisation ,” in *2017 IEEE 12th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 12 2017.

[7] J. Arukgoda, R. Ranasinghe, L. Dantanarayana, G. Dissanayake, and T. Furukawa, “Vector Distance Function Based Map Representation for Robot Localisation ,” in *The Australian Conference on Robotics and Automation (ACRA) 2017*, 12 2017.

[8] P. Bahl and V. N. Padmanabhan, “Radar: an in-building rf-based user location and tracking system,” in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, 2000, pp. 775–784 vol.2.

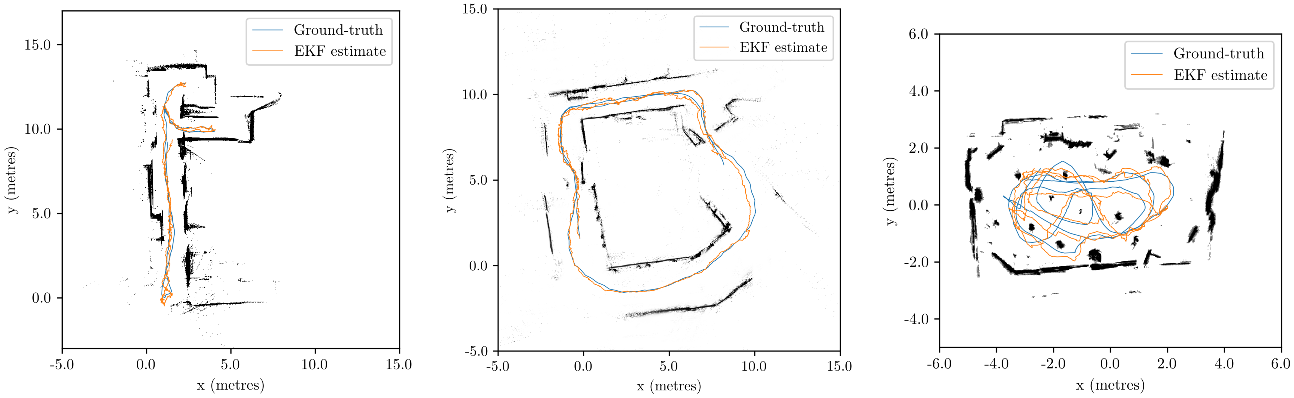
[9] P. Tarrío Alonso, H. Martín Rodríguez, and A. M. Bernardos Barbolla, “Enhancing the Performance of Propagation Model-Based Positioning Algorithms,” 1 2009. [Online]. Available: http://oa.upm.es/5601/1/INVE_MEM_2009_68030.pdf

[10] Cisco, “Location Tracking Approaches,” in *Wi-Fi Location-Based Services 4.1 Design Guide*, 2008.

[11] J. Torres-Sospedra, R. Montoliu, S. Trilles, . Belmonte, and J. Huerta, “Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems,” *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 12 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415005527

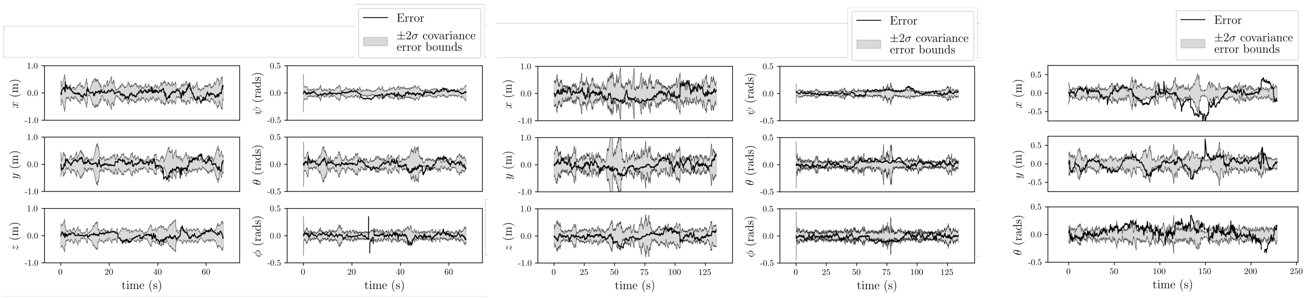
[12] J. Kolakowski, A. Consoli, V. Djaja-Josko, J. Ayadi, L. Morrigia, and F. Piazza, “UWB localization in EIGER indoor/outdoor positioning system,” in *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. IEEE, 9 2015, pp. 845–849. [Online]. Available: http://ieeexplore.ieee.org/document/7341422/

[13] H. Rimminen, M. Linnavuo, and R. Sepponen, “Human identification and localization using active capacitive rfid tags and an electric field floor sensor,” *Int. Rev. Electr. Eng*, vol. 5, pp. 1061–1068, 2010.



(a) (b) (c)

Fig. 4: The trajectories from datasets collected in (a) the domestic environment, (b) the office environment, and (c) the lab environment.



(a) (b) (c)

Fig. 5: Pose errors and covariances for (a) apartment, (b) CAS office, and (c) lab environment datasets [24].

[14] J. Rivera-Rubio, I. Alexiou, and A. A. Bharath, "Appearance-based indoor localization: A comparison of patch descriptor performance," *Pattern Recognition Letters*, vol. 66, pp. 109–117, 2015.

[15] D. Schulz and D. Fox, "Bayesian color estimation for adaptive vision-based robot localization," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2004, pp. 1884–1889.

[16] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image-retrieval system with monte carlo localization," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 208–216, 2005.

[17] N. Yazawa, H. Uchiyama, H. Saito, M. Servieres, and G. Moreau, "Image based view localization system retrieving from a panorama database by surf," in *Proc. IAPR Conf. on Machine Vision Applications (MVA)*, 2009, pp. 118–121.

[18] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4762–4769.

[19] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[20] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[22] P. Mullen, F. De Goes, M. Desbrun, D. Cohen-Steiner, and P. Alliez, "Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets," *Computer Graphics Forum*, vol. 29, no. 5, pp. 1733–1741, 9 2010.

[23] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[24] A. Schmidt, M. Fularz, M. Kraft, A. Kasiński, and M. Nowicki, "An indoor rgb-d dataset for the evaluation of robot navigation algorithms," in *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, A. Kasinski, W. Philips, D. Popescu, and P. Scheunders, Eds. Cham: Springer International Publishing, 2013, pp. 321–329.

[25] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[26] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri *et al.*, "Lasagne: First release." Aug. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.27878