Noname manuscript No.
(will be inserted by the editor)

SeDAR: Reading floorplans like a human

Using Deep Learning to enable human-inspired localisation

Oscar Mendez[1] · **Simon Hadfield**[1] · **Nicolas Pugeault**[2] · **Richard Bowden**[1]

Received: date / Accepted: date

Abstract The use of human-level semantic information to aid robotic tasks has recently become an important area for both Computer Vision and Robotics. This has been enabled by advances in Deep Learning that allow consistent and robust semantic understanding. Leveraging this semantic vision of the world has allowed human-level understanding to naturally emerge from many different
approaches. Particularly, the use of semantic information to aid in localisation and reconstruction has been at the forefront of both fields.

Like robots, humans also require the ability to localise within a structure. To aid this, humans have designed high-level semantic maps of our structures called floorplans. We are extremely good at localising in them, even with limited access to the depth information used by robots. This is because we focus on the distribution of semantic elements, rather than geometric ones. Evidence of this is that humans are normally able to localise in a floorplan that has not been scaled properly. In order to grant this ability to robots, it is necessary to use localisation approaches that leverage the same semantic information humans use.

In this paper, we present a novel method for semantically enabled global localisation. Our approach relies on the semantic labels present in the floorplan. Deep Learning is leveraged to extract semantic labels from RGB images, which are compared to the floorplan for localisation. While our approach is able to use range measurements if available, we demonstrate that they are unnecessary as we can achieve results comparable to state-of-the-art without them.

This work was funded by the EPSRC under grant agreements (EP/R512217/1) and (EP/R03298X/1) and Innovate UK Autonomous Valet Parking Project (Grant No 104273). We would also like to thank NVIDIA Corporation for their GPU grant.

[1]University of Surrey, Guildford, Surrey, UK
[2]University of Exeter, Exeter, Devon, UK

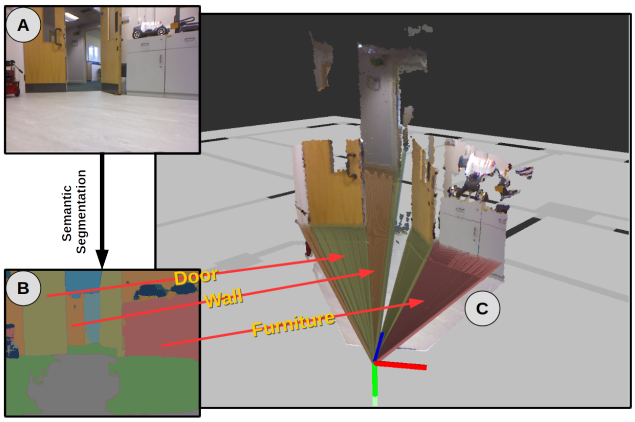


Fig. 1: A) RGB Image, B) CNN-Based Semantic Labelling and C) Sample SeDAR Scan within floorplan.

1 Introduction

Localisation, the process of finding a robot's pose within a pre-existing map, is one of the most important aspects of both Computer Vision and Robotic systems. A globally consistent, well localised sensor can substantially reduce the complexity of problems like Multi-View Stereo (MVS) [1, 31], Autonomous Navigation [44], 3D Reconstruction [8, 32] and even Deep Learning[25]. While all of these problems can estimate their own sensor poses, such as MVS using a Bundle Adjustment (BA) and Autonomous Navigation using Simultaneous Localisation and Mapping (SLAM). Unfortunately, both BA and SLAM suffer from the same limitation: they can only ever guarantee global pose consistency *internally*. This means that while pose estimates are globally consistent, they are only valid within the context of the localisation system. There are no guarantees, at least in vision-

only systems, that the reconstruction can be directly mapped to the real world, or between agents (without explicit alignment). This paper will attempt to address these limitations with a localisation approach that is efficient, accurate and, most importantly, globally consistent with the real-world.

For a robotic system, it should be clear that offline batch approaches are of limited use [16, 17]. This leaves traditional SLAM systems as the only viable approach for localisation. However, SLAM systems are liable to drift in terms of both pose and scale. They can also become globally inconsistent (even internally) in the case of failed loop closures.

This problem is normally addressed by having a localisation system that can relate the pose of the robot to a pre-existing map. Examples of global localisation frameworks include the Global Positioning System (GPS) and traditional Monte-Carlo Localisation (MCL). MCL has the ability to localise within an existing floorplan (which can be safely assumed to be available for most indoor scenarios). This is a highly desirable trait, as it implicitly eliminates drift, is globally consistent and provides a way for the 3D reconstructions to be related to the real world without having to perform expensive post-hoc optimisations. Traditionally, the range-based scans required by MCL have been produced by expensive sensors such as Light Detection And Ranging (LiDAR). These sensors are capable of producing high density measurements at high rates with low noise, making them ideal for range-based MCL. However, as a sensor they are expensive, are physically large and have high power requirements which is an issue for small mobile platforms.

As a response to this, modern low-budget robotic platforms have used RGB-D cameras as a cheap and low-footprint alternative. This has made vision-based floorplan localisation an active topic in the literature. However, while many approaches have been proposed, they normally use heuristics to lift the 2D plan into the 3D coordinate system of SLAM. These heuristics include techniques like assuming the height of doors and walls [28, 52]. Making assumptions about the world allows full 6-Degrees of Freedom (DoF) pose estimations to be computed (by using the assumed geometry). However, this also increases the computational cost and makes algorithms unsuitable for environments that do not conform to these assumptions. Other examples include Liu *et al*. [28], who use visual cues such as Vanishing Points (VPs) or Chu *et al*. [7] who perform piecemeal 3D reconstructions that can then be fitted back to an extruded floorplan. These approaches use innovative ways to extract 3D information from images, however, the data extracted from the image is normally not contained in the floorplan that the sensor is meant to localise in. Fundamentally, this means assumptions must be made about the floorplan. More explicitly, assumptions are made about information *not present* in the floorplan (*e.g*. ceiling and door height). It also does not fully exploit the floorplan, ignoring the semantic information that humans use to localise.

In order to find a robust solution to MCL, inspiration can be drawn from the way humans localise within a floorplan. People do not explicitly measure depths to every visible surface and try to match them against different pose estimates in the floorplan. However, this is exactly how most robotic scan-matching algorithms operate. Similarly, humans do not extrude the 2D geometry present in the floorplan into 3D, as is done in most vision-based approaches. Humans do the exact opposite. Instead of depth, people use high level semantic cues. Instead of extruding the floorplan up into the third dimension, humans collapse the 3D world into a 2D representation. Evidence of this is that many of the floorplans used in everyday life are not strictly accurate or in 3D. Instead, floorplans designed for people opt instead for high levels of discriminative landmarks on a 2D map.

Therefore, this paper proposes a fundamentally different approach that is inspired by how humans perform the task. Instead of discarding valuable semantic information, a Convolutional Neural Network (CNN) based encoder-decoder is used to extract high-level semantic information. All semantic information is then collapsed into 2D, in order to reduce the assumptions about the environment. A state-of-the-art sensing and localisation framework is then introduced, which uses these labels (along with image geometry and, optionally, depth) to localise within a semantically labelled floorplan. It is important to note that this paper explicitly avoids the 3D case because the information necessary for indoor navigation is present in the 2D representation. Therefore, we aim for a fast and efficient localisation approach that does not *require* 3D information.

Semantic Detection and Ranging (SeDAR) is an innovative human-inspired framework that combines new semantic sensing capabilities with a novel semantic Monte-Carlo Localisation (MCL) approach. As an example, Figure 1 shows a sample SeDAR scan localised in the floorplan. SeDAR has the ability to surpass LiDAR-based MCL approaches. SeDAR also has the ability to perform drift-free local, as well as global, localisation. Furthermore, experimental results show that the semantic labels are sufficiently strong visual cues that depth estimates are no longer needed. Not only does this vision-only approach perform comparably to depth-based methods, it is also capable of coping with floorplan inaccuracies more gracefully than strictly depth-based approaches. Furthermore, this approach relies on high-level semantic cues making it robust to repetitive and texture-less regions.

This paper presents several important extensions to our preliminary work [33] presented at the International Conference on Robotics and Automation (ICRA). Firstly, we extend our method to operate on all SUN3D labels (rather than wall, door and window) and add the ability to create

semantic floorplans from a known pose and a SeDAR scan. Secondly, to assist in reproducing the work, we add a significant amount of detail to the methodology, including a complete formalisation of MCL and the SeDAR sensing modality. Thirdly, we create an expansive new dataset for semantic localisation, make it publicly available and add comparison against state-of-the-art SLAM algorithms. Fourthly, we use this extended dataset to explore new properties of the proposed algorithm including results on a hand-drawn map. Finally, an evaluation on the TUM_RGB-D dataset is performed. This evaluation includes the creation of new semantic floorplans and a comparison against state-of-the-art SLAM (2D and 3D) and MCL algorithms

This paper describes the process by which SeDAR is used as a human-inspired sensing and localisation framework. To do this, a generic definition and formalisation of MCL is presented first. Following this, the semantically salient elements are extracted from a floorplan and an RGB image is parsed into a SeDAR scan. The three main novelties of this paper are then presented. In the first, the semantic information present in the floorplan is used to define a new motion model. In the second, the SeDAR scan is used to define a novel sensor model using a combination of range and label information. In the third, an additional sensor model is presented that only depends on label information (an RGB image). Finally, we present localisation results on several datasets and modalities.

2 Literature Review

The field of SLAM is predicated on the simple idea that the pose of a sensor and the reconstructed landmarks are conditioned on each other [11,43]. This idea is not limited to raw features, but can also be done at the level of objects, as shown by McCormac *et al*. [29]. However, if one of them is known *a priori*, it is possible to marginalise the other [36]. In the same way that independent reconstruction algorithms [16,17] can provide more robust representations of the world, independent localisation algorithms can also provide more robust and consistent pose estimates. In fact, recent work by Scheider *et al*. [40] explore the idea that a pre-existing SLAM map is an extremely useful asset for further mapping sessions. However, in each of these cases the environment must be navigated *a-priori*. Instead we propose to use pre-existing, human-readable (and therefore innacurate) 2D floorplans to localise, requiring no initial mapping session.

It is clear that an accurate map will yield an accurate localisation, and scan-matching localisation approaches [15, 10] use this fact successfully. However, independent localisation algorithms can also be extremely useful when only inaccurate maps are available. A clear example of this is the way humans localise within "theme-park"-like maps that encode coarse information using high-level landmarks. While it might not be possible to localise within these maps with millimetre accuracy, these maps (and the techniques that use them) are ideal for solving problems such as loop closure, global localisation, etc. This paper attempts to use this idea by combining pre-existing floorplans with image-based semantic segmentation to provide high-accuracy localisation in 2D.

While it might be desirable to estimate full 3D poses, recent work by Sattler *et al*. [39] demonstrates that large-scale 3D models are not strictly necessary for accurate vision-based localisation. Sattler *et al*. further conclude that 2D-based localisation approaches using coarse maps can be a good first step towards highly accurate localisation. This insight is important to this paper, where the aim is to localise within a 2D floorplan *without* making assumptions about the 3D structure of the building.

2.1 Monte-Carlo Localisation

MCL can be considered the state-of-the-art for mobile robot localisation today. Introduced by Dallaert *et al*. [10], MCL is a form of Particle Filter (PF) where each particle is a pose estimate (and the map is known). It uses a motion model to propagate particles which in turn causes the weights to become the observation likelihood given the pose [48]. Resampling based on the weights then focuses computation in areas with more probable pose estimates.

Monte-Carlo Localisation (MCL) was made possible by the arrival of accurate range-based sensors such as Sound Navigation And Ranging (SoNAR) and LiDAR. These Range-Based Monte-Carlo Localisation (RMCL) approaches are robust, reliable and still considered state-of-the-art in many robotic applications. As such, they will be discussed first below.

Recent advances in computer vision have made vision-based approaches possible. These approaches, called Vision-Based Monte-Carlo Localisation (VMCL), typically use RGB cameras to avoid expensive sensors and will be discussed second.

Finally, the recent rise in Deep Learning has made semantic-based approaches possible. These approaches rely on neural networks to extract semantic information from the world, and use it to localise. Semantic sensing modalities, such as the one presented in this paper, have the ability to revolutionise MCL.

2.1.1 Range-Based Monte-Carlo Localisation (RMCL)

RMCL was first introduced by Fox *et al*. [15] and Dellaert *et al*. [10]. RMCL improved the Kalman Filter based state-of-the-art by allowing multi-modal distributions to be repre-

sented. It also solved the computational complexity of grid-based Markov approaches. More recent approaches, such as those proposed by Kanai *et al*. [23], have moved the focus of RMCL into 3D. Kanai *et al*. focus on a pre-existing 3D reconstruction and simulate 3D depth readings at each particle. In what is probably the closest approach to ours, Bedkowski *et al*. [3] use a 3D LiDAR scanner, extract normals and use them to segment floors, walls, doors and edges between labels. They then use an approach based on Iterative Closest Point (ICP), with added label constraints, to estimate the observation likelihood. While this seems like a very promising approach, Bedowski *et al*. use very simple heuristics to classify their points (surface normals, point height, etc.). This work differs from these approaches by using techniques based on Deep Learning to provide better estimates of semantic labels and more robust observation likelihoods.

2.1.2 Vision-Based Monte-Carlo Localisation (VMCL)

RMCL-based approaches require expensive LiDAR and/or SoNAR sensors to operate reliably. Instead, Dellaert *et al*. [9] extended their approach to operate using vision-based sensor models.

VMCL allowed the use of rich visual features and low-cost sensors, but had limited performance compared to the more robust LiDAR-based systems. However, with the rising popularity of RGB-D sensors, more robust vision-based MCL approaches became possible. Fallon *et al*. [14] presented a robust MCL approach that used a low fidelity *a priori* map to localise, but required the space to be traversed by a depth sensor beforehand. Brubaker *et al*. [5] removed the need to traverse a map with a sensor, and instead used visual odometry, pre-existing roadmaps and a joint MCL/closed-form approach in order to localise a moving car. More recently, visual approaches began to resemble traditional MCL by localising in an extruded floorplan. Winterhalter *et al*. [52] performed MCL using an RGB-D camera, basing the observation likelihood on the normals of an extruded floorplan. Chu *et al*. [7] removed the RGB-D requirement, by creating piecemeal reconstructions and basing the observation likelihood on direct ICP between these reconstructions and the extruded floorplan. Similar work by Neurbert *et al*. [37] also removed the RGB-D requirement, using synthesised depth images from the floorplan and comparing the gradient information against an RGB image, allowing purely monocular localisation. However, these approaches all rely on geometric information to provide an observation likelihood.

MCL-based approaches tend to be robust, but they operate entirely on the *geometric* information present in the floorplan. Therefore, they require depth images directly from sensors and/or local SLAM-based reconstructions. By contrast, our approach aims to use *non-geometric* semantic information present in the floorplan in order to perform the localisation.

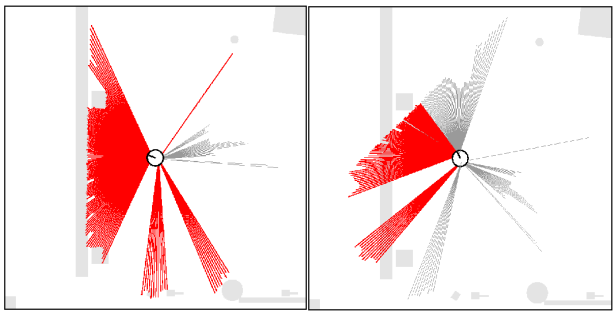
The use of semantic information for indoor localisation has been enabled by advances in Deep Learning, such as the approaches of Badrinarayanan *et al*. [2], Kendal *et al*. [24] and Long *et al*. [41]. More importantly, approaches like that of Holder *et al*. [20] have begun to take these approaches outdoors. Poschmann *et al*. [38], and the work presented in this paper, attempt to use semantic information in an MCL context. Poschmann *et al*. follow a very similar approach to Neurbert *et al*. but synthesise semantic images (rather than depth ones) and base the observation likelihood on photometric consistency with a CNN-based segmentation method (on an RGB image). However, the work presented in this paper does not synthesise semantic images but rather uses the semantic segmentation of the real observation to augment traditional LiDAR-like sensors. Furthermore, we make no assumptions about the 3D environment, and instead rely on RGB observations and a 2D floorplan.

2.2 Closed-Form Localisation Approaches

While the field of MCL evolved in the robotics community, non-MCL-based approaches became more popular in the vision community. Shotton *et al*. [42] used regression forests to predict the correspondences of every pixel in the image to a known 3D scene, they then combined this in a RANdom SAmple and Consensus (RANSAC) approach in order to solve the camera pose. Melbouci *et al*. [30] used extruded floorplans, but performed local bundle adjustments instead of MCL. Caselitz *et al*. [6] use a local SLAM system to create reconstructions that are then aligned using ICP to a LiDAR-built 3D map. However, instead of MCL they optimise the correspondences with a non-linear least squares approach.

More recent approaches have begun to also look at semantic information. Wang *et al*. [51] use text detection from shop fronts as semantic cues to localise in the floorplan of a shopping centre. Liu *et al*. [28] who use floorplans as a source of geometric and semantic information, combined with vanishing points, to localise monocular cameras. These vision-based approaches tend to use more of the non-geometric information present in the floorplan. However, a common trend is that assumptions must be made about geometry not present in the floorplan (*e.g*. ceiling height). The floorplan is then extruded out into the 3$^{\text{rd}}$ dimension to allow approaches to use the information present in the image.

The proposed approach differs from the approach of Poschmann *et al*. [38], Wang *et al*. [51] and Liu *et al*. [28] in two important ways. Firstly, it does not require an extruded floorplan, opting instead to project the sensory information down to 2D and localise there. This makes our approach be able to run in real time. Secondly, it has the capability



(a) Correct (b) Incorrect

Fig. 2: Laser scan matching, the robot is correctly localised when the observations match the geometry of the map [47].

of augmenting traditional LiDAR sensors making it a more generic solution.

We use a CNN-based semantic segmentation (that is understandable to humans) in order to extract labels that are inherently present in human-readable floorplans. This allows us to take all that information and collapse it into a 3-DoF problem, making our approach more
tractable than competing 6-DoF approaches while avoiding additional assumptions.

3 Problem Definition

While there exist many approaches to perform MCL, Range-Based Monte-Carlo Localisation (RMCL) [52,7] is widely considered to be the state-of-the-art localisation method for pre-existing maps. RMCL is a scan-matching algorithm, it assumes the presence of a sensor that provides range and bearing tuples across a scanline. The problem then becomes one of finding the pose of the robot that makes the sensor observations match the floorplan. Figure 2a shows a case of the scan being correctly matched for a correctly localised robot. Conversely, Figure 2b shows an incorrectly matched scan for an incorrect pose.

State-of-the-art localisation performs this matching in a Sequential Monte-Carlo (SMC) [10] framework, which can be broadly summarised as follows. Firstly, there is a prediction stage where particles are propagated using a motion-model, which is normally odometry from the robot (with Gaussian noise). Secondly, an update phase where each particle is weighted according to how accurately the observations align to the map. Finally, a re-sampling step is performed proportional to the weight of each particle and the process is then repeated.

More formally, the current pose $\mathbf{x}_t \in \mathbb{X}_t \subset \mathrm{SE}(2)$ can be estimated as a set of possible pose samples $\mathbb{S}_t = \left\{ s_t^i; i = 1..N \right\}$ given odometry measurements $\mathbb{U}_t = \left\{ u_j; j = 1..t \right\}$, sensor measurements $\mathbb{Z}_t = \left\{ z_j; j = 1..t \right\}$ and a 2D map $\mathbb{V}$. Under the assumption that all odometry measurements are equally likely, the posterior is calculated as

$$\Pr\left(s_t^i \middle| \mathbb{Z}_t, \mathbb{U}_t\right) \propto \Pr\left(z_t \middle| s_t^{i\prime}, \mathbb{V}\right) \Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right) \Pr\left(s_{t-1}^i \middle| \mathbb{Z}_{t-1}, \mathbb{U}_{t-1}\right), \tag{1}$$

which implies that only the most recent odometry and observations are used [10]. This means that at each iteration the particles from $\Pr\left(s_{t-1}^i \middle| \mathbb{Z}_{t-1}, \mathbb{U}_{t-1}\right)$ are: propagated using a motion model $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right)$, weighted using a sensor model $\Pr\left(z_t \middle| s_t^{i\prime}, \mathbb{V}\right)$ and resampled according to the posterior $\Pr\left(s_t^i \middle| \mathbb{Z}_t, \mathbb{U}_t\right)$. Algorithm 1 describes this process in more detail.

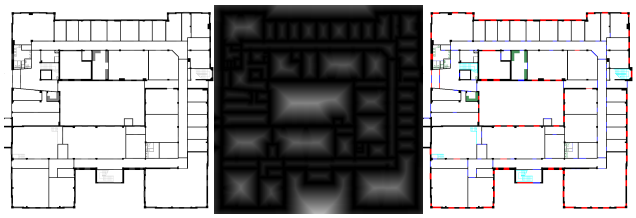
function MCL($\mathbb{S}_{t-1}, u_t, z_t$)
  $\mathbb{S}_t = \mathbb{S}'_t = \emptyset$
  for $i = 1 \rightarrow N$ **do**
    $s_t^{i\prime} \leftarrow$ MOTION_MODEL(u_t, s_{t-1}^i)
    $w_t^i \leftarrow$ SENSOR_UPDATE(z_t, $s_t^{i\prime}$, $\mathbb{V}$)
    $\mathbb{S}'_t \leftarrow \mathbb{S}'_t + \left\langle s_t^{i\prime}, w_t^i \right\rangle$
  end for
  for $i = 1 \rightarrow N$ **do**
    $s_t \leftarrow$ WEIGHTED_SAMPLE($\mathbb{S}'_t$)
    $\mathbb{S}_t \leftarrow \mathbb{S}_t + s_t$
  end for
  $\bar{\mathbb{S}}_t \leftarrow$ MEAN($\mathbb{S}_t$)
  return $\bar{\mathbb{S}}_t$
end function

Algorithm 1: Sequential Monte-Carlo Localisation in a known floorplan.

As stated previously, in an MCL context the prediction stage is performed using a motion model. The motion model is defined by the odometry received from the robot (u_t). This odometry can be used to "shift" the particles, assigning a likelihood based on the probability of the final position given the measured odometry. More formally, particles are propagated according to u_t with Gaussian noise applied such that

$$\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right) \sim \mathcal{N}\left(u_t + s_{t-1}^i, \boldsymbol{\Upsilon}_t\right) \tag{2}$$

where $\boldsymbol{\Upsilon}_t$ is the covariance of the odometry, and the symbol $\sim$ implies $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^i\right)$ is distributed as $\mathcal{N}\left(u_t + s_{t-1}^i, \boldsymbol{\Upsilon}_t\right)$ meaning Gaussian noise is applied to the linear and angular components of the odometry. This means the motion model allows MCL-based approaches to reason about the noise characteristics of their odometry. While it would be impossible to fully account for noise in the odometry (due to wheel slippage, changing model parameters, etc.), a well tuned motion



(a) Original (b) Likelihood Field (c) Semantic

Fig. 3: Original floorplan compared to the likelihood field and the labelled floorplan.

model allows for a robust estimate. In Section 6.1, the traditional definition of a motion-model is augmented to include a "ghost factor" that uses semantic information to influence how particles move through occupied space.

The sensor model is defined by each range-scanner observation. The probability of each full range-scan (z_t) can be estimated under the assumption that each measurement in the scan is independent of each other. That is,

$$\Pr\left(z_t \middle| s_t^{i\prime}, \mathbb{V}\right) = \prod_{k=1}^{K} \Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) \tag{3}$$

is the likelihood of the putative particle $s_t^{i\prime}$, where

$$z_t = \left\{\left\langle \theta_t^k, r_t^k \right\rangle; k = 1..K\right\} \tag{4}$$

is the set of range and bearing tuples that make up each scan. Calculating the likelihood can be done two ways, using a beam model [49] or a likelihood field model [46].

In the beam model, a raycasting operation is performed. Starting from the pose of the current particle, a ray is cast along the bearing angle θ_t^k. The raycasting operation terminates when an occupied cell is reached and the likelihood is estimated as

$$\Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = e^{\frac{-\left(r_t^k - r_t^{k*}\right)^2}{2\sigma_o^2}} \tag{5}$$

where r_t^k is the range obtained from the sensor and r_t^{k*} is the distance travelled by the ray.

In the likelihood field model, a distance map is used in order to avoid the expensive raycasting operation. The distance map is a Lookup Table (LUT) of the same size as the floorplan, where each cell contains the distance to the nearest geometry. This map is estimated similar to a Chamfer distance [4], where a search is performed in a window around each cell and the distance to the closest occupied cell in the floorplan is stored. When queried, this distance is converted into a likelihood using equation 5. Figure 3 shows the estimated distance map for a floorplan, the creation of which will be explored further in Section 6.2. This distance map is only estimated once during initialisation. During runtime, the endpoint of each measurement can be estimated directly from the pose, bearing and range. The probability is then simply related to the distance reported by the LUT.

The raycasting method is (strictly speaking) more closely related to the sensing modality, as the closest geometry may not lie along the ray. However, in practice, most robotics systems use the likelihood field model as it is both faster and tends to provide better results. This is because the raycasting operation can report incorrect measurements due to small pose errors. An example of this is when looking through an open door, an error of a few centimetres can make the rays miss the door. This makes the distribution inherently less smooth.

4 Methodology

The problem with state-of-the-art approaches is that they only use the range information from the sensor, fundamentally limiting how discriminative each reading can be.

Instead, this paper presents a semantic sensing and localisation framework called SeDAR. SeDAR introduces a likelihood field model that incorporates semantically salient information into the traditional range-enabled approach. In an alternative approach, SeDAR combines the raycasting and likelihood field approaches in a novel formulation which allows localisation without range measurements. Experimental evaluation shows that SeDAR outperforms traditional RMCL when using both semantic and depth measurements. When using semantic-only measurements, it is shown that SeDAR can perform comparably to depth-enabled approaches.

5 Semantic Labelling and Sensing

Before using the semantic labels to aid in floorplan localisation, it is necessary to extract them. To do this, a floorplan is labelled in order to identify semantically salient elements. These salient elements are then identified in the camera of the robot by using a state-of-the-art CNN-based semantic segmentation algorithm [24].

5.1 Floorplan

RMCL requires a floorplan and/or previously created range-scan map that is accurate in scale and globally consistent, this presents a number of challenges. A previously created range-scan map requires a robust SLAM algorithm such as GMapping [19] to be run. This is not ideal as it forces the robot to perform an initial exploration to construct a map before localisation can be performed. Moreover, the SLAM algorithm is also sensitive to noise and the resulting map is

difficult to interpret by humans. Instead of using a metric-accurate reconstruction, a more flexible and feasible alternative is using a human-readable floorplan.

RMCL is not robust to differences between the floorplan and what the robot can observe (*e.g.* inaccuracies, scale variation and furniture). To overcome these issues, the localisation is augmented with semantic labels extracted from an existing floorplan. For the remainder of this section, and without loss of generality, the labels will be limited to walls, doors and windows. The reason for this limitation is two-fold. Firstly, they are salient pieces of information that humans naturally use to localise and are therefore easy to discuss. Secondly, they are simple to automatically extract from a floorplan using image processing. In practice, we use simple image processing techniques along with manual labeling to create a labeled floorplan. As can be seen in Figure 3c, these semantically salient elements have been colour coded to highlight the different labels. It should be noted, that this limitation will be lifted in section 7.4, where all the labels in the CNN-based semantic segmentation algorithm [24] are used to both construct the floorplan and localise within it.

In order to make a labelled floorplan readable by the robot, it must first be converted into an occupancy grid. An occupancy grid is a 2D representation of the world, in which each cell in the grid has an occupancy probability attached to it. Any cell that is above a threshold is then considered as being occupied. Estimating the occupancy of an existing floorplan is done by taking the normalised greyscale value from the floorplan image.

The map can then be defined as a set of voxels

$$\mathbb{V} = \left\{ v_{\mathbf{m}}; \mathbf{m} \in \mathbb{M} \right\} \tag{6}$$

where $\mathbb{M}$ is a set of integer 2D positions. Assuming $\mathcal{L} = \{a, d, w\}$ is the set of possible cell labels (wall, door, window), each cell is defined as

$$v_{\mathbf{m}} = \left\langle v^{o}_{\mathbf{m}}, v^{w}_{\mathbf{m}}, v^{d}_{\mathbf{m}}, v^{a}_{\mathbf{m}} \right\rangle \tag{7}$$

where $v^{o}_{\mathbf{m}}$ is the occupancy likelihood and $v^{\ell}_{\mathbf{m}}$, where $\ell \in \mathcal{L}$, denotes the label likelihood. The semantic floorplans presented in this work maintain occupancy and label likelihoods, which can then be either thresholded (as in equation 14) or used directly.

Having incorporated the semantic labels into the standard occupancy grid, it is now necessary to use them in sensing.

5.2 SeDAR Sensor

Extracting semantic labels from a robot-mounted sensor is one of the most important parts of SeDAR. It is theoretically possible to directly label range-scans from a LiDAR-based scanner. In fact, there is a wide range of landmark-based SLAM systems that use range sensors [12]. However, there are limitations on the amount of information that can be extracted from a range-scan.

Beyond the structure of the environment, the additional information contained in floorplans pertains to important architectural features (such as doors and windows). These architectural features are well defined in terms of their appearance. Therefore, they are ideally suited to semantic segmentation of the image.

In SeDAR, labels are extracted from the RGB image only. This is by design, as it allows the use of cameras that cannot sense depth. In the following sections, this sensing modality will be used in a novel MCL framework that does not require range-based measurements. However, it should be noted that SeDAR is capable of using range measurements, should they be available.

If they are used, SeDAR is completely agnostic to the source of the depth measurements. They can come from a deep learning-based depth estimation [27] or a dense Structure from Motion (SfM) system [13]. However, for the purposes of this paper, a simple RGB-D sensor is used. Either way, the method for parsing an RGB-D image into a SeDAR scan is the same.

5.2.1 RGB-D to SeDAR

For a low-cost robotic system that uses an RGB-D image as a proxy for a more expensive LiDAR scanner, a horizontal depth scanline is typically extracted from the depth image as

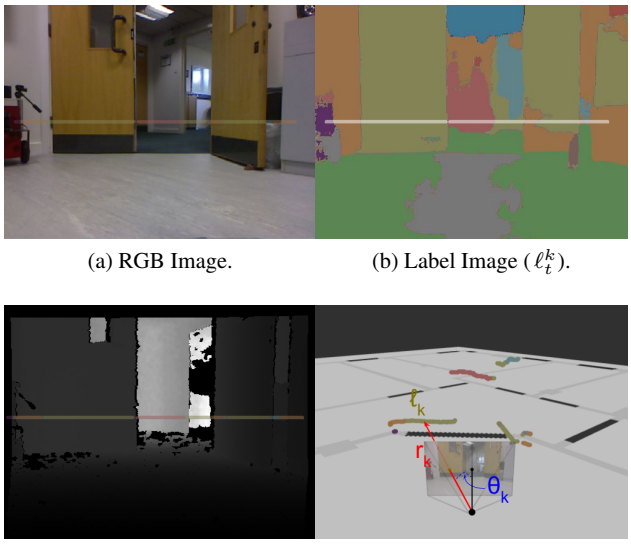
$$z_t = \left\{ \left\langle \theta^k_t, r^k_t \right\rangle; k = 1..K \right\}, \tag{8}$$

where θ^k_t is the angle around the vertical axis and r^k_t is the corresponding range. This can be accomplished by looking exclusively at the depth image.

The angle around the vertical axis, θ^k_t, can be calculated by

$$\theta^k_t = \operatorname{atan2}\left(\frac{u - c_x}{f_x} \right) \tag{9}$$

where (u, v), (c_x, c_y), (f_x, f_y) are the pixel coordinates, principal point and focal length, respectively, of the camera. While it is possible to estimate a second angle along the vertical axis, this is unnecessary in the case of floorplan localisation. More importantly, incorporating this information into the localisation framework requires assumptions to be made about the floorplan (*e.g.* ceiling height). The underlying assumption is that the centre scanline corresponds to casting rays *parallel* to the floorplan. This implies the camera must be parallel to the ground plane. However, cameras mounted at an arbitrary Special Orthogonal Space (SO(3)) orientation can still be used assuming that an appropriate scanline is



(a) RGB Image. (b) Label Image (ℓ_t^k).

(c) Depth Image (r_t^k). (d) SeDAR Scan.

Fig. 4: Visualisation: sensor input, semantic segmentation and the resulting SeDAR scan.

used. In practice, small errors in the orientation of the camera are negligible.

The range measurement r_t^k can be calculated as

$$r_t^k = \sqrt{\left(\frac{d_t^k\,(u - c_x)}{f_x}\right)^2 + \left(\frac{d_t^k\,(v - c_y)}{f_y}\right)^2 + \left(d_t^k\right)^2} \tag{10}$$

where d_t^k is the current depth measurement at pixel k. At this point, a traditional range-scan can be emulated. Notice that in a standard range-scan, all the visible information present in the RGB image is being discarded. On the other hand, a SeDAR-scan consists of a set of bearing, range and label tuples,

$$z_t = \left\{\left\langle \theta_t^k,\, r_t^k,\, \ell_t^k \right\rangle; k = 1..K\right\}, \tag{11}$$

where ℓ_t^k is the semantic label. While the scanline still discards a large amount of information in the RGB image, it is important to note that the methods used to estimate the label have already used the context the image provides. It would also possible to look at wider scanlines and provide likelihoods for each label (rather than a single label). In our experiments, this has been unnecessary.

In order to estimate the labels, a CNN-based encoder-decoder network is used, trained on the SUN3D [53] dataset, that can reliably detect doors, walls, floors, ceilings, furniture and windows. This state-of-the-art semantic segmentation runs at frame-rate on an NVIDIA Titan Xp, which allows images to be parsed into a SeDAR-scan with negligible latency. The label ℓ_t^k is then simply the label at pixel k.

It is important to note that the CNN has not been fine-tuned to any specific task. In fact, this is an important limitation of the approach presented in this paper. When the semantic segmentation fails, the observations become unreliable. This means that correct particles can given low scores and removed from the filter. However, in practice, the CNN appears to generalise well to most indoor environments.

Figure 4 shows the input images and the resulting SeDAR scan. Figure 4a shows the RGB image from which the label image in Figure 4b is extracted. Figure 4c shows the depth image. In all of these, the scanline shown in the middle of the image denotes specific pixel locations where ℓ_t^k and r_t^k are extracted from the label and depth image, respectively. Finally, Figure 4d shows the resulting SeDAR scan, where the scanline can be seen localised within a floorplan. A localised range-less SeDAR scanline would look similar to this, as every $(\langle \theta_t^k,\, \ell_t^k \rangle)$ tuple would perform ray-tracing until it hit an obstacle. Without ray-tracing, the scanline would simply have no depth. Now that the semantic labels are added into the map and the sensor, they can be used in a novel MCL algorithm.

6 Semantic Monte-Carlo Localisation

It has been shown that there is a large amount of easily-attainable semantic information present in both the floorplan and the image. This information has been largely ignored in the MCL literature in favour of range-based approaches.

In this Section, this semantic information is combined into a novel semantic MCL approach. In the motion model, the semantic information is used to inform collision models. In the sensor model two approaches are presented. The first introduces a likelihood field model that incorporates semantically salient information into the traditional approach. The second approach combines the raycasting and likelihood field approaches into a method which allows localisation without range measurements.

6.1 Motion Model

Equation 2 formalised the motion model as $\Pr\left(s_t^{i\prime} \middle| u_t,\, s_{t-1}^i\right)$. However, it is well understood in the literature that the actual distribution being approximated is $\Pr\left(s_t^{i\prime} \middle| u_t,\, s_{t-1}^i,\, \mathbb{V}\right)$. This encodes the idea that certain motions are more or less likely depending on the map (*e.g.* through walls).

Under the assumption that the motion of the robot is small, it can be shown that

$$\Pr\left(s_t^{i\prime} \middle| u_t,\, s_{t-1}^i,\, \mathbb{V}\right) = \kappa \Pr\left(s_t^{i\prime} \middle| u_t,\, s_{t-1}^i\right) \Pr\left(s_t^{i\prime} \middle| \mathbb{V}\right) \tag{12}$$

(see *e.g.* [47]) where κ is a normalising factor and $\mathbb{V}$ is the set containing every cell in the map. This allows the two likelihoods to be treated independently.

In an occupancy map, the motion $\Pr\left(s_t^{i\prime} \middle| u_t, s_{t-1}^{i}\right)$ is defined in the same way as equation 2. The prior $\Pr\left(s_t^{i\prime} \middle| \mathbb{V}\right)$ is simply the occupancy likelihood of the cell that contains $s_t^{i\prime}$, that is

$$\Pr\left(s_t^{i\prime} \middle| \mathbb{V}\right) = 1 - \Pr\left(v^o_{s_t^{i\prime}}\right) \tag{13}$$

which is an elegant solution in the case where the "floorplan" was previously built by the robot.

However, this approach becomes problematic when using human-made floorplans. Human-made floorplans typically have binary edges (when they are made on a computer) or edges with image artefacts (when they are scanned into a computer). This does not reflect what the robot can observe and can cause issues with localisation. Therefore, most approaches tend to assume a binary interpretation of the occupancy. This is done by setting the probability to

$$\Pr\left(v^o_{s_t^{i\prime}}\right) = \begin{cases} 1 & \text{if } v^o_{s_t^{i\prime}} \geq \tau_o \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where τ_o is a user defined threshold. This thresholding operation is necessary when the floorplan is not created by the robot (*e.g.* using scan-matching). While this makes depth-based methods perform reliably, it is a crude estimate of reality. For instance, most humans would not even notice if a door is a few centimetres away from where it should be. Issues like this present real problems when particles propagate though doors, as it is possible that the filter will discard particles as they collide with the edge of the door frame.

Instead, the motion model presented here uses semantic information to augment this with a *ghost factor* that allows particles more leeway in these scenarios. Therefore the proposed prior is

$$\Pr\left(s_t^{i\prime} \middle| \mathbb{V}\right) = \left(1 - \Pr\left(v^o_{s_t^{i\prime}}\right)\right) e^{-\epsilon_{\text{G}}\delta_d} \tag{15}$$

where δ_d is the distance to the nearest door. While other labels such as windows can be used, in the case of a ground-based robot doors are sufficient. The distance, δ_d, can be efficiently estimated using a lookup table as defined in Section 6.2.

More importantly, ϵ_{G} is a user defined factor that determines how harshly this penalty is applied. Setting $\epsilon_{\text{G}} = 0$ allows particles to navigate through walls with no penalty, while very high values approximate equation 14. The effects of ϵ_{G} will be explored in Section 7.1.1. This motion model is more probabilistically accurate than the occupancy model used in most RMCL approaches, and has the added advantage of leveraging the high-level semantic information present in the map.

Having presented a semantically enabled motion model, it is now necessary to give the sensor model the same treatment.

6.2 Sensor Model

The naïve way of incorporating semantic measurements into the sensor model would be to use the beam model. In this modality, the raycasting operation would provide not only the distance travelled by the ray, but also the label of the cell the ray hit. If the label of the cell and the observation match, the likelihood of that particle being correct is increased. However, this approach suffers from the same limitations as the traditional beam model: it has a distinct lack of smoothness. On the other hand, the likelihood field model is significantly smoother, as it provides a gradient between each of the cells. By contrast, the approach presented here uses a joint method that can use likelihood fields to incorporate semantic information in the presence of semantic labels. More importantly, it can also use raycasting within a likelihood field in order to operate without range measurements.

As described in Section 3, the likelihood field model calculates a distance map. For each cell $v_{\mathbf{m}}$, the distance to the nearest occupied cell

$$\delta_o(\mathbf{m}) = \min_{\mathbf{m}'} \|\mathbf{m} - \mathbf{m}'\|, \quad v^o_{\mathbf{m}'} > \tau_o \tag{16}$$

is calculated and stored. When a measurement $z_t^k = \left\langle \theta_t^k, r_t^k \right\rangle$ is received, the endpoint is estimated and used as an index to the distance map. Assuming a Gaussian error distribution, the weight of each particle $s_t^{i\prime}$ can then be estimated as

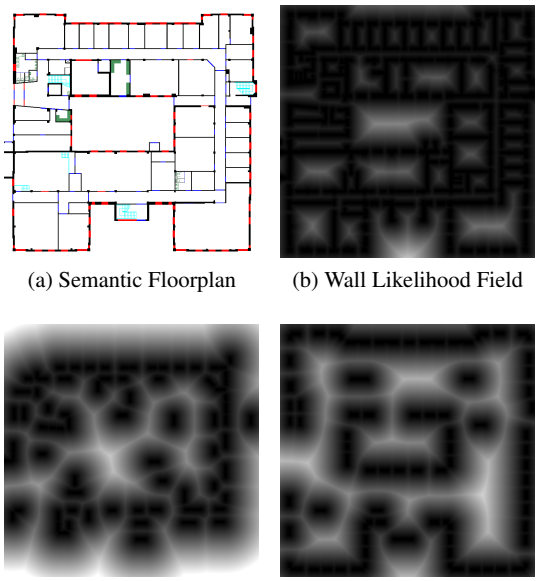
$$\Pr_{\text{RNG}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = e^{\frac{-\delta_o^2}{2\sigma_o^2}} \tag{17}$$

where δ_o is the value obtained from the distance map and σ_o is dictated by the noise characteristics of the sensor. However, this model has three main limitations. Firstly, it makes no use of the semantic information present in the map. Secondly, the parameter σ_o must be estimated by the user and assumes all measurements within a scan have the same noise parameters. Thirdly, it is incapable of operating in the absence of range measurements.

Instead, as mentioned in Section 5.1, this work uses the semantic labels present in the map to create multiple likelihood fields. For each label present in the floorplan, a distance map is calculated. This distance map stores the shortest distance to a cell with the same label.

Formally, for each map cell $v_{\mathbf{m}}$ the distance to the nearest cell of each label is estimated as

$$\delta_\ell(\mathbf{m}) = \min_{\mathbf{m}'} \|\mathbf{m} - \mathbf{m}'\|, \quad v^\ell_{\mathbf{m}'} > \tau_o \tag{18}$$



(a) Semantic Floorplan (b) Wall Likelihood Field

(c) Door Likelihood Field (d) Window Likelihood Field

Fig. 5: Original floorplan compared to the likelihood field for each label.

where $\delta_{\ell} \in \{\delta_a, \delta_d, \delta_w\}$ are distances to the nearest wall, door and window, respectively. Figure 5 shows the distance maps for each label. For clarity, the argument $(\mathbf{m})$, is omitted for the remainder of the paper.

This approach overcomes the three limitations of the state-of-the-art. Firstly, the use of semantic information [7,10,23, 28,52]. Secondly, adapting the sensor noise parameters to the map [10,23,52]. Thirdly, operation in the absence of range measurements [3,10,23,52]. These points will now be discussed.

6.2.1 Semantic Information

Most localisation approaches [7,10,23,28,52] do not use any semantic information present in the map. While approaches such as that of Bedkowski *et al.* [3] and Poschmann [38] have begun to use this information, they either rely on geometric primitives for their semantic segmentation approach ([3]) or rely on synthetic 3D reconstructions of the map ([38]). Contrary to this, SeDAR uses the semantic information present in the map. When an observation $z_t^k = \langle \theta_t^k, r_t^k, \ell_t^k \rangle$ is received, the bearing θ_t^k and range r_t^k information are used to estimate the endpoint of the scan. The label ℓ_t^k is then used to decide which semantic likelihood field to use. Using the endpoint from the previous step, the label-likelihood can be estimated similarly to equation 17,

$$\Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = e^{\frac{-\delta_{\ell}^2}{2\sigma_{\ell}^2}} \tag{19}$$

where δ_{ℓ} is the distance to the nearest cell of the relevant label and σ_{ℓ} is the standard deviation (which will be defined using the label prior). The probability of an observation given the map and pose can then be estimated as

$$\Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = \epsilon_o \Pr_{\text{RNG}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) + \epsilon_{\ell} \Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) \tag{20}$$

where ϵ_o and ϵ_{ℓ} are user defined weights. When $\epsilon_{\ell} = 0$ the likelihood is the same as standard RMCL. On the other hand, when $\epsilon_o = 0$ the approach uses only the semantic information present in the floorplan. These weights are explored and defined in Section 7.1.1. Unlike range scanners, σ_{ℓ} cannot be related to the physical properties of the sensor. Instead, this standard deviation is estimated directly from the prior of each label on the map. Defining σ_{ℓ} this way has the benefit of not requiring tuning. However, there is a much more important effect that must be discussed.

6.2.2 Semantically Adaptive Standard Deviation

Most approaches [10,23,52] rely on hand-tuned parameters for the standard deviation of the observation likelihood σ_o. However, when a human reads a floorplan, unique landmarks are the most discriminative features. The more unique a landmark, the easier it is to localise using it (because there are not many areas in the map that contain it). It then follows that the more rare a landmark, the more discriminative it is for the purpose of localisation. Indeed, it is easier for a person to localise in a floorplan by the configuration of doors and windows than it is by the configuration of walls. This translates into the simple insight: *lower priors are more discriminative*. Therefore, σ_{ℓ} is tied to the prior of each label not only because it is one less parameter to tune, but because it implicitly makes observing rare landmarks more beneficial than common landmarks.

Relating σ_{ℓ} to the label prior $\Pr(\ell)$ controls how smoothly the distribution decays w.r.t. distance from the cell. We make the likelihoods more spatially *lenient* on sparser labels: the smaller $\Pr(\ell)$ is, the smoother the decay. In essence, this allows more discriminative landmarks to contribute towards the localisation from further away.

6.2.3 Range-less Semantic Scan-Matching

The final, and most important, strength of this approach is the ability to perform all of the previously described methodology in the complete absence of range measurements. Most

approaches [3,10,23,52] are incapable of operating without the use of range measurements. Those that are capable of range-less performance [7,28], rely on strong assumptions about the geometry ([28]) and/or estimate a proxy for depth measurements ([7]). Both these cases have important limitations that are avoided by our semantic scan-matching.

The approach has so far been formalised on the assumption of either $\langle \theta_t^k, r_t^k \rangle$ tuples (existing approaches) or $\langle \theta_t^k, r_t^k, \ell_t^k \rangle$ tuples (SeDAR-based approach). However, our approach is capable of operating directly on $\langle \theta_t^k, \ell_t^k \rangle$ tuples. In other words, depth measurements are *explicitly* not added or used.

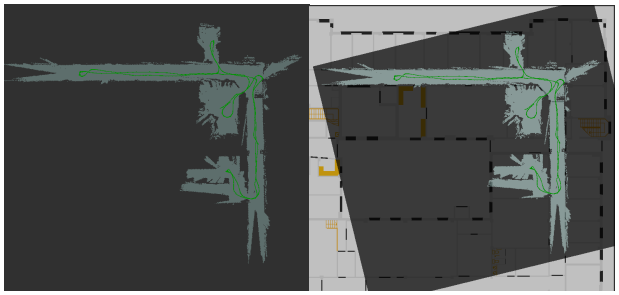
Incorporating range-less measurements is simple. The beam and likelihood field models are combined in a novel approach that avoids the degeneracies that would happen in traditional RMCL approaches. In equation 5, the likelihood of a ray is estimated using the difference between the range (r_t^k) obtained from the sensor and the range (r_t^{k*}) obtained from the raycasting operation. Unfortunately, in the absence of a range-based measurement (r_t^k) this is impossible. Using the standard distance map is also impossible, since the endpoint of the ray cannot be estimated. Using raycasting in the distance map also fails similarly: the raycasting terminates on an occupied cell, implying $\delta_o = 0$ for every ray cast.

On the other hand, the semantic likelihood fields can still be used as δ_ℓ will still have a meaningful and discriminative value. This operation is called semantic raycasting. For every $z_t^k = \langle \theta_t^k, \ell_t^k \rangle$, the raycasting is performed as described in Section 3. However, instead of comparing r_t^k and r_t^{k*} or using δ_o, the label ℓ_t^k is used to decide what likelihood field to use. The cost can then be estimated as

$$\Pr\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) = \Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right) \tag{21}$$

where $\Pr_{\text{LBL}}\left(z_t^k \middle| s_t^{i\prime}, \mathbb{V}\right)$ is defined in equation 19. This method is essentially a combination of the beam-model and the likelihood field model. More explicitly, a ray is cast along the bearing of every observation $z_t^k = \langle \theta_t^k, \ell_t^k \rangle$ tuple. Once the raycasting hits an occupied cell, we use the occupied cell's location to perform a lookup into the likelihood field corresponding to the observation's label. This gives us a distance to the nearest cell with the same label. If the sensor is correctly localised, every distance should be zero. If it isn't, the likelihood fields provide a smooth cost-function towards the correct pose.

It would be possible to assign binary values (*i.e.* label matches or not) to equation 21. This approach would make the observation likelihood directly proportional to the series of labels along the scanline (*i.e.* how closely the bearing/label tuples match what is observable from each particle's pose). However, this would be a naïve solution that provides no smooth gradient to the correct solution. Instead, this approach uses the *angular distribution* of labels, combined with distances from the likelihood field, to provide a smooth cost-function that converges reliably.



(a) Ground Truth (b) Overlay to Floorplan

Fig. 6: Sample Trajectory used for evaluation.

The previous Sections have presented a series of methods to localise a ground-based robot on a pre-existing floorplan. In the following Section, it will be shown that these methods are capable of outperforming standard RMCL approaches when using range-measurements. Moreover, it will be shown that they provide comparable performance when operating exclusively on bearing/label tuples from RGB images without range information.

7 Evaluation

This section will evaluate the strengths of the approach. As an initial step, an evaluation and parameter exploration will be performed on a dataset consisting of a robot driving around a building. As a second step, our approach will be benchmarked on the popular TUM-RGBD dataset [45].

7.1 Human-Readable Floorplans

Evaluation on our the first dataset, which will be relased along with the paper, will focus on three main experiments. Firstly, a thorough evaluation of the performance of the system for a single trajectory is performed along with a parameter exploration of SeDAR. This is done in order to give an insight into the intrinsic characteristics of SeDAR mentioned in section 4. Secondly, a repeatability experiment is undertaken, where the performance of multiple similar trajectories is evaluated. This is done in order to demonstrate the robustness and performance of SeDAR. Finally, an evaluation on a more challenging hand-drawn map is performed. This experiment allows us to demonstrate that SeDAR can localise in geometrically inaccurate maps.

In order to evaluate the approach, a dataset will ideally have several important characteristics. The dataset should

consist of a robot navigating within a human-readable floorplan. Human-readability is required to ensure semantic information is present. The trajectory should be captured with an RGB-D camera. This is in order to easily extract all the possible tuple combinations (range, bearing and label). Finally, the trajectory of the robot should be on the same plane as the floorplan.

To satisfy these constraints, we created a new dataset to evaluate our approach on. The floorplan in Figure 3a is used because it is large enough for meaningful tests and has human-readable floorplans available. The dataset was collected using the popular TurtleBot platform [18], as it has a front-facing Kinect that can be used for emulating both LiDAR and SeDAR. The dataset, will be released along with the publication of this paper.

Normally, the ground-truth trajectory for floorplan localisation is either manually estimated (as in [52]) or estimated using external Motion Capture (MoCap) systems (as in [45]). However, both of these approaches are limited in scope. Manual ground-truth estimation is time-consuming and impractical. MoCap is expensive and difficult to calibrate, especially over the large public areas required for floorplan localisation. In order to overcome these limitations, a well established RGB-D SLAM system [26] is used to provide an initial estimate. This estimate is then manually refined, using both a computationally expensive global optimisation and judicious manual intervention. While it does not localise within a floorplan, it does provide an accurate reconstruction and trajectory for the robot, which can then be registered with the floorplan. Figure 6a shows a sample trajectory and map estimated by [26], while Figure 6b shows them overlaid on the floorplan.

To quantitatively evaluate SeDAR against ground truth, the Absolute Trajectory Error (ATE) metric presented by Sturm *et al*. [45] is used. ATE is estimated by first registering the two trajectories using the closed form solution of Horn [21], who find a rigid transformation ${}^{\mathbb{G}}\mathbf{T}_{\mathbb{X}}$ that registers the trajectory $\mathbb{X}_t$ to the ground truth $\mathbb{G}_t$. At every time step t, the ATE can then be estimated as

$$e_{\mathbf{g}} = \mathbf{g}_t^{-1}\, {}^{\mathbb{G}}\mathbf{T}_{\mathbb{X}}\, \mathbf{x}_t \tag{22}$$

where $\mathbf{g}_t \in \mathbb{G}_t$ and $\mathbf{x}_t \in \mathbb{X}_t$ are the current time-aligned poses of the ground truth and estimated trajectory, respectively. The Root Mean Square Error (RMSE), mean and median values of this error metric are reported, as these are indicative of performance over coarse room-level initialisation. In order to visualise the global localisation process, the error of each successive pose is shown (error as it varies with time). These metrics are sufficient to objectively demonstrate the systems ability to globally localise in a floorplan, while also being able to measure room-level initialisation performance.

The work presented here is compared against the extremely popular MCL approach present in the Robot Operating System (ROS), called Adaptive Monte Carlo Localisation (AMCL) [10]. AMCL is the standard MCL approach used in the robotics community. Any improvements over this approach are therefore extremely valuable. Furthermore, Adaptive Monte Carlo Localisation (AMCL) [10] is considered to be the state-of-the-art and is representative of the expected performance of the RMCL approaches detailed in Section 2.1, such as Kanai *et al*. [23], Bedkowski *et al*. [3], Winterhalter *et al*. [52] and Chu *et al*. [7]. To demonstrate that our algorithm can outperform SLAM, we also compare against 2D scan-matching [19], monocular [34] and RGB-D [35] approaches in the coarse (room-level) scenario.

In all experiments, any common parameters (such as σ_o) are kept the same. The only parameters varied are ϵ_ℓ, ϵ_o and $\epsilon_{\mathbb{G}}$.

7.1.1 Detailed Analysis of a Single Trajectory

In order to establish a baseline of performance, and to explore the characteristics of SeDAR discussed in this paper, we first present a thorough evaluation of a single trajectory on a clean floorplan. This trajectory, seen in figure 6, covers multiple rooms and corridors and is therefore a representative sample to evaluate on.

As a first experiment, a room-level initialisation is given to both AMCL and the proposed approach. This means that the uncertainty of the pose estimate, roughly corresponds to telling the robot what room in the floorplan it is in. More explicitly, the standard deviations on the pose estimate are of $2.0m$ in (x, y) and $2.0rad$ in θ. The systems then ran with a maximum of 1000 particles (minimum 250) placed around the covariance ellipse. The error is recorded as each new image in the dataset is added. For the SLAM approaches, it is not necessary to define an initialisation. However, it was necessary to increase the number of features on ORB-SLAM2 so that the tracking could be perfomed successfully. Apart from this, all SLAM algorithms ran with their default parameters.

Figure 7 compares four distinct scenarios against AMCL. Of these four scenarios, two use the range measurements from the Microsoft Kinect (blue lines) and two use only the RGB image (red lines).

The first range-enabled scenario uses the range measurements to estimate the endpoint of the measurement (and therefore the index in the distance map) and sets the range and label weights to ($\epsilon_o = 0.0$ and $\epsilon_\ell = 1.0$), respectively. This means that while the range information is used to inform the lookup in the distance map, the costs are only computed using the labels. The second range-enabled scenario performs a weighted combination ($\epsilon_o = 0.25, \epsilon_\ell = 0.75$) of both the semantic and traditional approaches. It is interesting

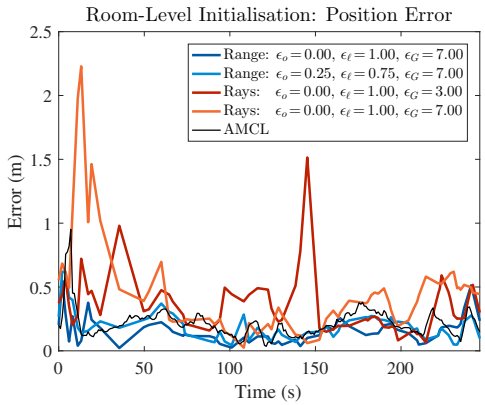


Fig. 7: Semantic Floorplan Localisation, room-level initialisation.

to note that this performs slightly worse than the label-only approach, likely because the geometric cues in a hallway environment are relatively weak compared to semantic cues.

In terms of the ray-based version of this approach, equation 21 is used. This means there are no parameters to set. Instead, a mild ghost factor ($\epsilon_G = 3.0$) and a harsh one ($\epsilon_G = 7.0$) are shown.

Since coarse room-level initialisation is an easier problem than global initialisation, the advantages of the range-enabled version of this approach are harder to see compared to state-of-the-art. However, it is important to note how closely the ray-based version of the approach performs to the rest of the scenarios despite using no depth data. Apart from a couple of peaks, the ray-based method essentially performs at the same level as AMCL. This becomes even more noticeable in Table 1, where it is clear that range-based semantic MCL (using only the labels) outperforms state of the art, while the ray-based $\epsilon_G = 3.0$ version lags closely behind. The reason $\epsilon_G = 3.0$ performs better than $\epsilon_G = 7.0$ is because small errors in the pose can cause the robot to "clip" a wall as it goes through the door. Since $\epsilon_G = 3.0$ is more lenient on these scenarios, it is able to outperform the harsher ghost factors.

Table 1 also shows comparison against three SLAM algorithms. It is clear that monocular SLAM does not perform well in this scenario. This is because there are large areas of plain textureless regions where tracking is lost. RGB-D SLAM performs better in this scenario, as it can rely on depth cues to maintain tracking. Finally, 2D SLAM also performs well (although slightly worse than RGB-D). However, in all cases, SeDAR outperforms the performance of SLAM algorithms. Not only does the range-enabled SeDAR significantly outperform SLAM, but the ray-based approach also manages to outperform both 2D and 3D depth-enabled SLAM. These results present a clear indication that SeDAR-based localisation approaches are capable of outperforming SLAM methods.

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	0.24	0.21	0.20	0.11	0.04	0.95
GMapping [19]	0.71	0.63	0.57	0.31	0.32	1.43
ORB_SLAM2 (Mono) [34]	8.67	7.90	6.48	3.58	2.48	15.47
ORB_SLAM2 (RGB-D) [35]	0.43	0.40	0.38	0.16	0.09	0.73
Range (Label Only)	**0.19**	**0.16**	**0.14**	**0.10**	**0.02**	**0.55**
Range (Combined)	0.22	0.19	0.17	0.11	0.04	0.62
Rays ($\epsilon_o = 3.0$)	0.40	0.34	0.27	0.22	0.07	1.51
Rays ($\epsilon_o = 7.0$)	0.58	0.45	0.38	0.37	0.02	2.23

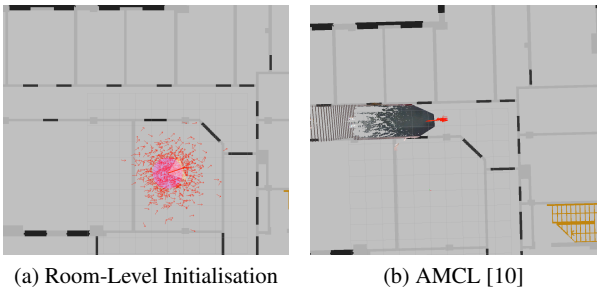
Table 1: Room-Level Initialisation

In order to give further context to these results, the results of state-of-the-art approaches by Winterhalter *et al.* [52] and Chu *et al.* [7] are mentioned here. These approaches are chosen as they present the most comparable methods in the literature. Although direct comparison is not possible (due to differences in the approach, and the availability of code and datasets) an effort has been made to present meaningful metrics. Winterhalter *et al.* [52] report (in their paper) an error of $0.2 - 0.5\text{m}$. Winterhalter *et al.* are estimating a 6-DoF pose, which might make this seem like an unfair comparison. However, they do this on a much smaller room-sized dataset meaning the error is relatively large. While they perform experiments on larger floorplan-level datasets, the errors reported are much noisier ranging between $0.2 - 2\text{m}$ on the coarse initialisation and $0.2 - 8\text{m}$ on the global initialisation. Chu *et al.* [7] report (in their paper) a mean error of 0.53m on the TUMindoor dataset [22], which is similar to the one presented here. These results present further evidence that the SeDAR-based localisation approach can outperform the state-of-the-art localisation approaches.

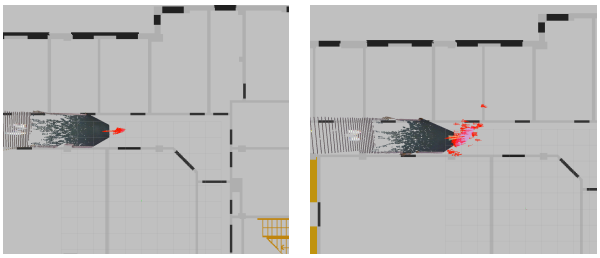
In terms of qualitative evaluation, both the convergence behaviour and the estimated path of MCL-based approaches is shown.

The convergence behaviour can be seen in Figure 8. Here, Figure 8a shows how the filter is initialised to roughly correspond to the room the robot is in. As the robot starts moving, it can be seen that AMCL (8b), the range-based version of SeDAR (8c) and the ray-based version (8d) converge. Notice that while the ray-based approach has a predictably larger variance on the particles, the filter has successfully localised. This can be seen from the fact that the Kinect point cloud is properly aligned with the floorplan. It is important to note that although the Kinect point cloud is present for visualisation in the ray-based method, the depth is *not* used.

The estimated paths can be seen in Figure 9, where the red path is the estimated path and green is the ground truth. Figure 9a shows the state-of-the-art, which struggles to converge at the beginning of the sequence (marked by a blue circle). It can be seen that the range-based approach in Figure 9b (combined label and range), converges more quickly and maintains a similar performance to AMCL. It only slightly deviates from the path at the end of the ambiguous corridor on the left, which also happens to AMCL. It can also be

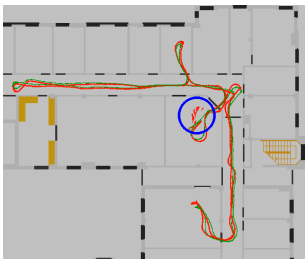


(a) Room-Level Initialisation (b) AMCL [10]

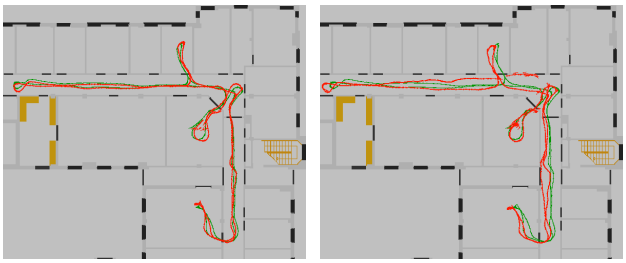


(c) SeDAR (Range-Based) (d) SeDAR (Ray-Based)

Fig. 8: Qualitative view of Localisation in different modalities.



(a) AMCL [10]



(b) SeDAR (Range-Based) Path (c) SeDAR (Ray-Based) Path

Fig. 9: Estimated path from coarse room-level initialisations.

seen that the ray-based approach performs very well. While it takes longer to converge, as can be seen by the estimated trajectory in Figure 9c, it corrects itself and only deviates from the path in areas of large uncertainty (like long corridors).

These experiments show that SeDAR-based MCL is capable of operating when initialised at the coarse room-level.

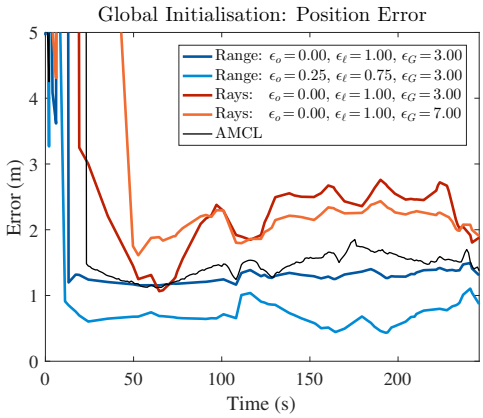


Fig. 10: Semantic Floorplan Localisation, global initialisation.

It is now important to discuss how discriminative SeDAR is when there is no initial pose estimate provided to the system.

Having evaluated against SLAM-based approaches in a local initialisation scenario, the focus will now be on the ability of SeDAR-based MCL to perform global localisation. In these experiments, the system is given no indication of where in the map the robot is. Instead, a maximum of $50,000$ particles (minimum $15,000$) are placed over the entire floorplan. Figure 10 shows the same four scenarios as in the previous section. For the range-based scenarios (blue lines) it can be seen that using only the label information ($\epsilon_o = 0.0, \epsilon_\ell = 1.00$) consistently outperforms the state of the art, both in terms of how quickly the values converge to a final result and the actual error on convergence. This shows that SeDAR used in an MCL context is more discriminative than the standard occupancy maps in RMCL. The second range-based measurement ($\epsilon_o = 0.25, \epsilon_\ell = 0.75$) significantly outperforms all other approaches. In this case, the strong geometric cues present in the dataset are helping the particle filter converge faster (therefore skewing the result in favour of the combined method).

In terms of the ray-based version of the approach (red lines), two scenarios are compared. A mild ghost factor ($\epsilon_G = 3.0$) and a harsh one ($\epsilon_G = 7.0$). These versions of the approach both provide comparable performance to the state-of-the-art. It is important to emphasise that this approach uses absolutely no range and/or depth measurements. As such, comparing against depth-based systems is inherently unfair. Still, SeDAR ray-based approaches compare favourably to AMCL. In terms of convergence, the mild ghost factor $\epsilon_G = 3.0$ gets to within several meters accuracy even quicker than AMCL, at which point the convergence rate slows down and is overtaken by AMCL. The steady state performance is also comparable. While the performance temporarily degrades, it manages to recover and keep a steady error rate throughout the whole run. On the other hand, the harsher

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	7.31	2.26	**0.20**	6.95	**0.028**	35.45
Range (Label Only)	6.71	2.59	1.31	6.20	1.15	38.60
Range (Combined)	**4.78**	**1.69**	0.69	**4.47**	0.43	**31.19**
Rays ($\epsilon_{\text{G}} = 3.0$)	7.74	4.36	2.46	6.40	1.07	27.55
Rays ($\epsilon_{\text{G}} = 7.0$)	8.09	4.49	2.22	6.73	1.61	28.47

Table 2: Global Initialisation

Average Trajectory Error (RMSE)			
Ghost Factor (ϵ_{G})	**Range (Labels)**	**Range (Weighted)**	**Rays**
0.0	10.88	10.13	11.71
3.0	**6.71**	**4.78**	**7.74**
5.0	6.97	6.30	9.54
7.0	7.19	6.10	8.09

Table 3: Global ATE for Different Ghost Factors

Average Trajectory Error (RMSE)			
Ghost Factor (ϵ_{G})	**Range (Labels)**	**Range (Weighted)**	**Rays**
0.0	0.25	0.27	1.20
3.0	0.24	0.25	**0.40**
5.0	0.22	0.24	0.70
7.0	**0.19**	**0.22**	0.58

Table 4: Room-Level ATE for Different Ghost Factors

ghost factor $\epsilon_{\text{G}} = 7.0$ takes longer to converge, but remains steady and eventually outperforms the milder ghost factor. Table 2 shows the RMSE, error along with other statistics. In this case, the combined range and label method performs best.

As before, qualitative analysis can be provided by looking at the convergence behaviour and the estimated paths.

In order to visualise the convergence behaviour, Figure 11a shows a series of time steps during the initialisation of the filters. On the first image, the particles have been spread over the ground floor of a $(49\text{m} \times 49\text{m})$ office area. In this dataset, the robot is looking directly at a door during the beginning of the sequence. Therefore, in Figure 11b the filter converges with particles looking at doors that are a similar distance away. The robot then proceeds to move through the doors. Going through the door means the filter converges significantly faster as it implicitly uses the ghost factor in the motion model. It also gives the robot a more unique distribution of doors (on a corner), which makes the filter converge quickly. This is shown in Figure 11c.

The estimated paths can be seen in Figure 12, where the blue circle denotes the point of convergence. It can be seen that AMCL takes longer to converge (further away from the corner room) than the range-based approach. More importantly, it can be seen that the range-based approach suffers no noticeable degradation in the estimated trajectory over the room-level initialisation. On the other hand, the performance of the ray-based method degrades more noticeably. This is because the filter converges in a long corridor with ambiguous label distributions (doors left and right are similarly spaced). However, once the robot turns around the system recovers and performs comparably to the range-based approach.

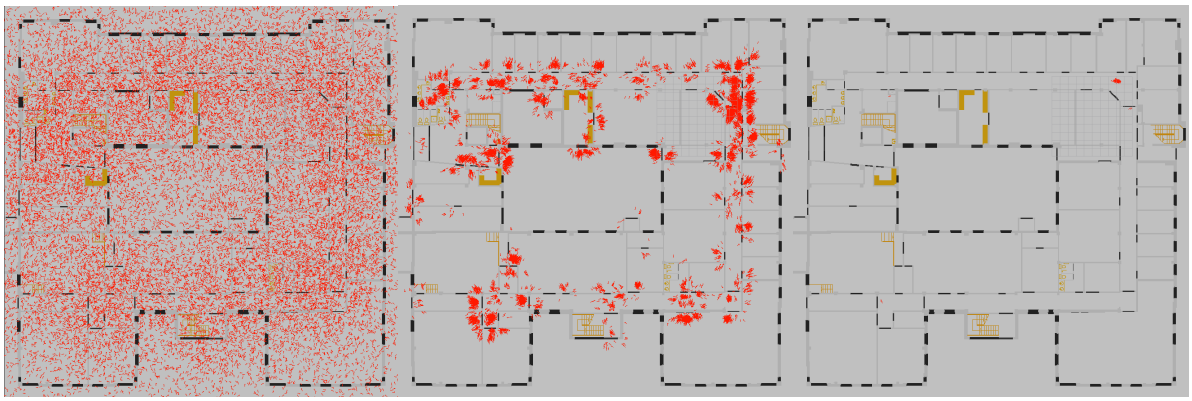
As mentioned previously, entering or exiting rooms helps the filter converge because it can use the ghost factor in the motion model. The following experiments, evaluate how the ghost factor affects the performance of the approach.

The effect of the ghost factor can be measured in a similar way to the overall filter performance. Results show that the ghost factor provides more discriminative information when it is *not* defined in a binary fashion. This is shown in the label-only scenario for both the range-based and ray-based approaches, in both the global and coarse room-level initialisation. Figure 13 shows the effect of varying the ghost factor during global initialisation. It can be seen that not penalising particles going through walls, $(\epsilon_{\text{G}} = 0)$, is not a good choice. This makes sense, as there is very little to be gained from allowing particles to traverse occupied cells without any consequence. It follows that the ghost factor should be set as high as possible. However, setting the ghost factor to a large value $(\epsilon_{\text{G}} = 7.0)$, which corresponds to reducing the probability by 95% at 0.43m, does not provide the best results.

While it might seem intuitive to assume that a higher ϵ_{G} will always be better, this is not the case. High values of the ghost factor correspond to a binary interpretation of occupancy which makes MCL systems unstable in the presence of discrepancies between the map and the environment. This happens because otherwise correct particles can clip door edges and be completely eliminated from the system. A harsh ghost factor also exacerbates problems with a limited number of particles. In fact, $\epsilon_{\text{G}} = 3.0$, corresponding to a 95% reduction at 1.0m, consistently showed the best results in all of the global initialisation experiments, as can be seen in Table 3.

In terms of room-level initialisation, having an aggressive ghost factor is more in line with the initial intuition. Table 4 shows that for both of the range-based scenarios, $\epsilon_{\text{G}} = 7.0$ provides the best results. This is because coarse room-level initialisation in the presence of range-based measurements is a much easier problem to solve. As such, the problem of particles "clipping" edges of doors is less of an issue.

On the other hand, the ray-based scenario still prefers a milder ghost factor of $\epsilon_{\text{G}} = 3.0$. In this scenario, inaccuracies in both the map and the sensing modalities allow for otherwise correct particles to be heavily penalised by an aggressive ghost factor. Both of these results are reflected in Figures 14a and 14b.

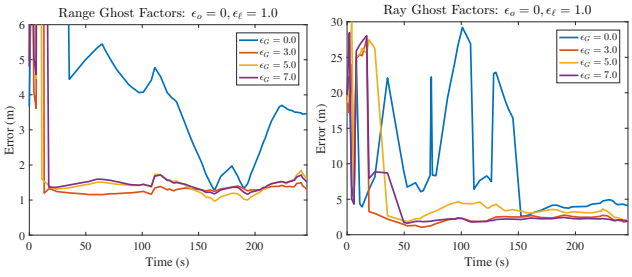


(a) Global Initialisation (b) Looking at Doors (c) Converged

Fig. 11: Qualitative view of Localisation in different modalities.

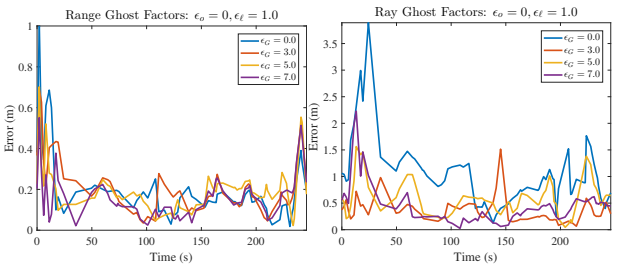
(a) AMCL Path (b) SeDAR (Range-Based) Path (c) SeDAR (Ray-Based) Path

Fig. 12: Estimated path from global initialisations.



(a) Range-Based (b) Ray-Based

Fig. 13: Different ghost factors (ϵ_{G}), global initialisation.



(a) Range-Based (b) Ray-Based

Fig. 14: Different ghost factors (ϵ_{G}), coarse room-level initialisation.

These results allow a single conclusion. The ghost factor must be tuned to the expected amount of noise in the map and sensing modality. Aggressive ghost factors can be used in cases where the pre-existing map is accurate and densely sampled, such as the case where the map was collected by the same sensor being used to localise (*i.e.* SLAM). On the other hand, in the case where there are expected differences between what the robot is able to observe (*e.g.* furniture, scale errors, etc.), it is more beneficial to provide a milder ghost factor in order to be more lenient on small pose errors.

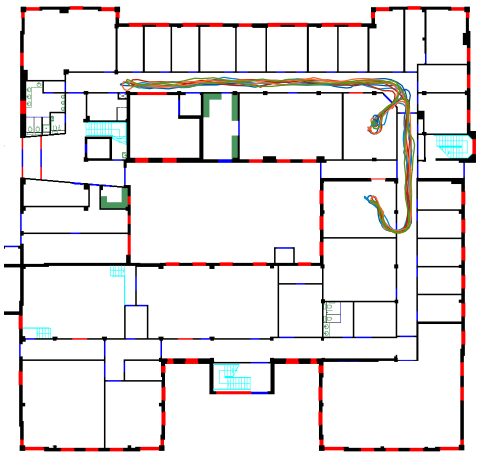


Fig. 15: Same trajectory repeated five times.

7.2 Cross-Trajectory Performance

In the previous section, we used a single trajectory to showcase the performance characteristics of SeDAR. It allowed us to gain important insights into the way it operates. In this section, we will aim to demonstrate that the performance is not limited to a single trajectory.

We do this by evaluating our performance on five different trajectories. In a global initialisation scenario, the stochastic nature of MCL approaches creates large variability in the ATE during initialisation. Therefore, using the room-level initialisation allows us to more meaningfully assess the performance on multiple trajectories.

The trajectories, shown in figure 15, are captured on a similar route to allow for direct comparison. However, they are captured at very different times, meaning they contain large variability in the visual domain. These trajectories include static and dynamic obstacles, people, changing geometry and other difficulties. They therefore present a challenge for state-of-the-art MCL and SLAM approaches, which normally assume a static map.

In table 5 we show the localisation performance of several methods accumulated over all five trajectories. The average value for each metric is presented, with its standard deviation shown in parenthesis. It can be seen that range-based SeDAR outperforms all competing approaches by a significant margin. More importantly, the ray-based version of SeDAR also significantly outperforms AMCL and monocular SLAM while performing comparably to RGB-D and scan-matching SLAM. Fundamentally, this means that there exist scenarios where geometric measurements are inferior to semantic understanding without depth.

We also compare against a several learning-based approach (PoseNet [25] and PoseLSTM [50]). To enable us to perform a meaningful comparison, we evaluate the ATE of PoseNet and PoseLSTM trained on the original trajectory consisting of 1128 images with ground truth pose. The algorithms are then tested on each of the 5 unseen pose trajectories. The results shown in table 5 demonstrate that SeDAR (Range and Ray) can outperform PoseNet and PoseLSTM. This can be explained by the fact that PoseNet and PoseLSTM maintain a single estimate of the pose, while SeDAR-based approaches can maintain multi-nomial distributions. However, it would be interesting to explore hybrid approaches where PoseNet-like localisation approaches can be used to initialise an MCL-like approach.

As seen in figure 16, AMCL does not perform well in this scenario because it struggles to initialise properly in several of the trajectories. This happens due to AMCL's naïve use of the floorplan. In SeDAR, the semantic information is inherently leveraged, as shown in figure 12b, in order to aid initialisation. By contrast, AMCL can only reason about the geometry of the scene which causes it to fall into local minima (as in figure 16b).

It is clear that correctly initialised AMCL should outperform ray-based SeDAR (but not range-based). However, ray-based SeDAR can offer more consistently correct initialisations despite the lack of depth information. Qualitatively, both range (figure 17) and ray-based (figure 18) SeDAR have much more consistent trajectories. While ray-based SeDAR is liable to noisier paths, it is still capable of accurately finding the correct path in all five trajectories. This implies that the semantic cues present in the floorplan are inherently more discriminative than the traditional geometric cues used by AMCL.

SLAM approaches do not suffer with incorrect localisation problems in the same way that AMCL does. However, monocular SLAM again struggles to maintain tracking in difficult indoor trajectories that include texture-less regions. On the the hand, RGB-D and scan-matching SLAM algorithms suffer with problems due to loop closure which means their trajectory estimates drift and introduce errors. By comparison both range and ray based SeDAR do not suffer with problems due to tracking or loop closures. This highlights an important strength of 2D localisation algorithms: they can leverage pre-existing maps. Our approach does not need to traverse the environment once before it can localise reliably within it.

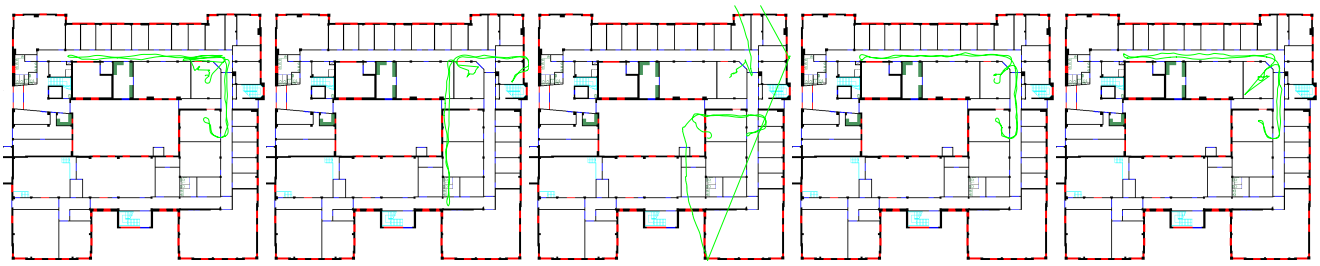
7.3 Inaccurate Hand-Drawn Map

The semantic cues in the floorplan are so discriminative, that it is possible to use them for localisation even when the geometric characteristics are severely compromised. To demonstrate this, we use a crudely hand-drawn version of the floorplan in the same multi-trajectory benchmark.

This inaccurate version of the world, seen in figure 19, presents a scenario where the geometry of the scene is com-

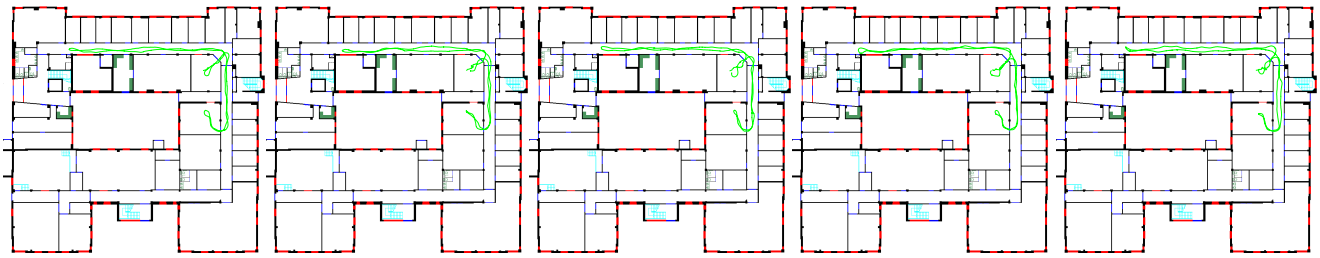
Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	3.66 (6.38)	2.86 (5.12)	1.93 (3.38)	2.25 (3.83)	0.77 (1.50)	11.53 (17.55)
GMapping [19]	0.57 (0.38)	0.51 (0.35)	0.50 (0.40)	0.22 (0.17)	0.25 (0.23)	0.84 (0.54)
ORB_SLAM2 (Mono) [34]	10.41 (0.69)	9.35 (0.54)	8.14 (1.07)	4.55 (0.63)	2.64 (0.68)	20.85 (1.97)
ORB_SLAM2 (RGB-D) [35]	0.57 (0.27)	0.49 (0.20)	0.44 (0.09)	0.29 (0.20)	0.13 (0.03)	1.22 (0.78)
PoseNet [25]	6.78 (0.54)	5.04 (0.53)	3.45 (0.44)	4.53 (0.36)	0.55 (0.23)	23.00 (1.66)
PoseLSTM [50]	6.76 (0.46)	4.85 (0.58)	3.36 (0.61)	4.68 (0.42)	0.53 (0.25)	24.72 (1.49)
Range (Label Only)	**0.33** (0.04)	**0.29** (0.04)	**0.27** (0.04)	**0.15** (0.01)	**0.04** (0.03)	**0.96** (0.11)
Range (Combined)	0.38 (0.05)	0.33 (0.06)	0.29 (0.08)	0.18 (0.03)	0.06 (0.07)	1.09 (0.19)
Rays ($\epsilon_G = 3.0$)	0.85 (0.21)	0.72 (0.18)	0.65 (0.14)	0.44 (0.11)	0.09 (0.05)	2.27 (0.55)
Rays ($\epsilon_G = 7.0$)	1.65 (0.83)	1.42 (0.74)	1.26 (0.62)	0.83 (0.41)	0.26 (0.33)	3.92 (2.05)

Table 5: Coarse Room-Level Initialisation (Multiple Trajectories)



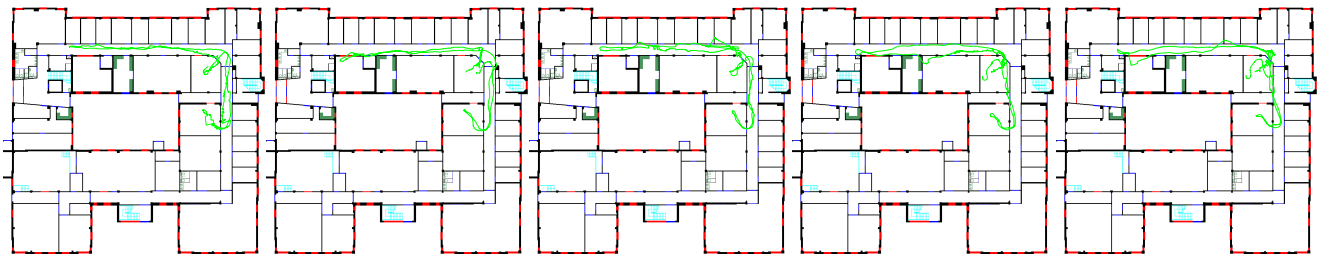
(a) AMCL Path 1 (b) AMCL Path 2 (c) AMCL Path 3 (d) AMCL Path 4 (e) AMCL Path 5

Fig. 16: Estimated paths from coarse room-level initialisations.



(a) SeDAR Range Path 1 (b) SeDAR Range Path 2 (c) SeDAR Range Path 3 (d) SeDAR Range Path 4 (e) SeDAR Range Path 5

Fig. 17: Estimated path from coarse room-level initialisations.



(a) SeDAR Ray Path 1 (b) SeDAR Ray Path 2 (c) SeDAR Ray Path 3 (d) SeDAR Ray Path 4 (e) SeDAR Ray Path 5

Fig. 18: Estimated path from coarse room-level initialisations.

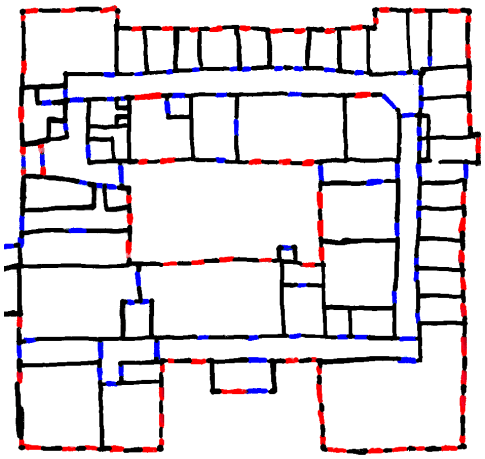


Fig. 19: Crude, hand-drawn floorplan.

promised *but the semantic elements are not*. To us, the image might look similar to the original floorplan because we focus on the unaltered semantic elements. To a robot, this image presents an important deviation from the geometrically accurate floorplan used in previous experiments. This effect is reflected in the performance of the strictly-geometric AMCL.

In the previous section, we used a coarse initialisation in order to benchmark the trajectories. In this case, we are interested in benchmarking the localisation performance within the hand-drawn map. As such, a global localisation will allow us to explore how discriminative semantic and geometric cues are. Furthermore, since the map is inherently inaccurate, any detail in the ground truth will be lost (due to ambiguity with the map).

In table 6, it is again shown that range-based SeDAR outperforms the state-of-the-art by a significant margin. On the other hand, the ray-based version performs comparably (although with much tighter margins) which implies a more consistent behaviour. The reason range-based approaches perform better, and ray-based approaches are more consistent is the same. As mentioned before, the geometric elements in this map are simply not enough for accurate localisation. The semantic elements must be used in order to achieve reasonable performance.

Qualitatively, the performance of these methods can be more closely evaluated. Figures 20, 21 and 22 show the global localisation trajectory once it has converged. In figure, 20, it can be seen that AMCL never actually converges to the right location. This means that AMCL has correctly localised in zero of the five trajectories. In fact, the reason its accuracy is even comparable to ray-based SeDAR is due to the registration step in ATE. More explicitly, AMCL finds a local minima early and relies on the odometry to provide an "accurate" trajectory in the wrong location. By contrast, SeDAR takes longer to converge but generally finds the correct location. Figure 21 shows that range-based SeDAR correctly localises in three out of five trajectories. Finally, Figure 22 correctly localises in four out of five trajectories (and is extremely close in the fifth). It is important to stress that ray-based SeDAR actually finds the correct position of the robot more accurately and consistently than both range-enabled approaches.

This type of behaviour implies that SeDAR has a higher level of understanding than AMCL. Our approach is capable of ignoring geometrically local minima because the semantic elements do not support it. This higher-level reasoning is an important step towards localisation on a human level.

7.4 Benchmark Evaluation

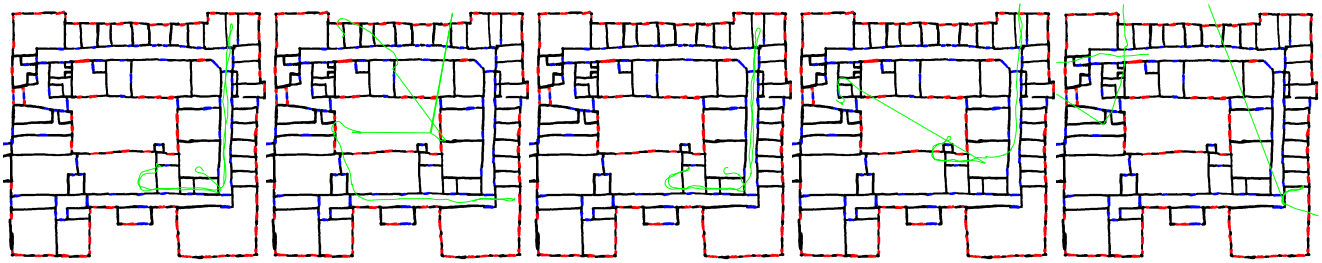
So far, SeDAR-based localisation has been demonstrated to outperform both MCL and SLAM state-of-the-art algorithms on a custom dataset. In this section, we will aim to evaluate SeDAR on a well-known SLAM dataset [45]. The evaluation will be performed against the same algorithms [10, 19, 34, 35] as used in the previous dataset.

The TUM-RGBD [45] dataset is a well established benchmark that contains many different trajectories. As part of this dataset, a Robot SLAM category captured on a Pioneer2 is included. Since our algorithm requires planar trajectories and a horizontal camera, we use this subset of the benchmark in our evaluation. The trajectories used are known as Pioneer_360, Pioneer_Slam, Pioneer_Slam2 and Pioneer_Slam3. An occupancy and semantic floorplan of the area the robot navigates were also created in order to enable evaluation. These maps can be seen in figure 23. It should be noted that SeDAR, AMCL [10] and GMapping [19] all exploit the planar constraint present in the dataset. ORB_SLAM2 (Mono and RGBD) [34, 35] does not have this constraint. It should also be noted that SLAM approaches ([19, 34, 35]) do not have prior knowledge of the environment.

Tables 7, 8, 9 and 10 show the results on Pioneer_360, Pioneer_Slam, Pioneer_Slam2 and Pioneer_Slam3, respectively. In Pioneer_360, Pioneer_Slam, Pioneer_Slam3, it is clear that range-based SeDAR outperforms all other approaches. It is also important to notice that ray-based SeDAR also outperforms monocular SLAM and performs similar to the depth-based approaches such as AMCL. The only exception to this is during Pioneer_Slam2, where monocular SLAM outperforms ray-based SeDAR. However, this is an extremely challenging sequence for monocular SLAM the system constantly looses tracking due to motion blur and low textures. In order to get these numbers, only partial trajectories were used (as it was impossible to obtain full sequences on monocular slam). Even then, the system required constant monitoring to ensure tracking was not lost. On the other hand, SeDAR reports a position for every pose in the trajectory.

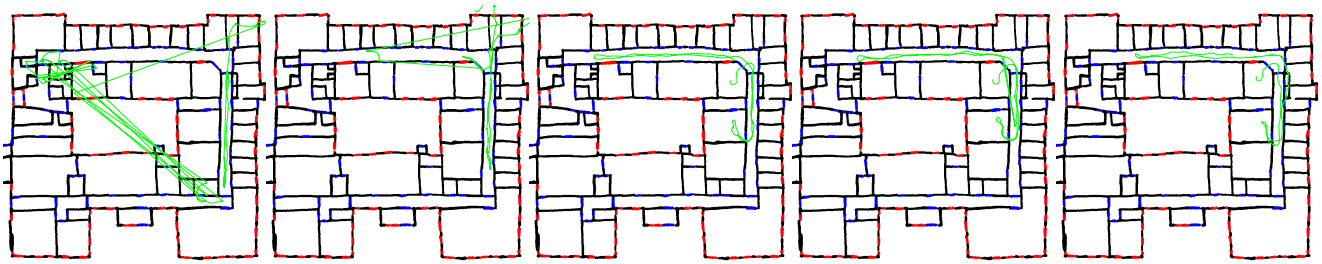
Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	11.18 (4.29)	8.57 (4.86)	7.31 (5.12)	6.69 (1.23)	0.98 (0.71)	30.39 (3.64)
Range (Label Only)	7.26 (4.06)	4.63 (4.30)	3.76 (4.76)	5.18 (1.57)	0.78 (0.80)	30.00 (2.55)
Range (Combined)	**4.64** (1.95)	**2.48** (1.43)	**1.39** (0.65)	**3.86** (1.49)	**0.33** (0.17)	27.08 (5.96)
Rays ($\epsilon_{\text{G}} = 3.0$)	11.39 (1.65)	9.72 (2.30)	8.30 (3.21)	5.67 (0.77)	2.09 (0.32)	**25.83** (4.38)
Rays ($\epsilon_{\text{G}} = 7.0$)	11.24 (2.22)	8.97 (2.74)	7.40 (3.75)	6.57 (0.35)	1.32 (0.95)	28.91 (2.08)

Table 6: Global Initialisation (Drawn Map)



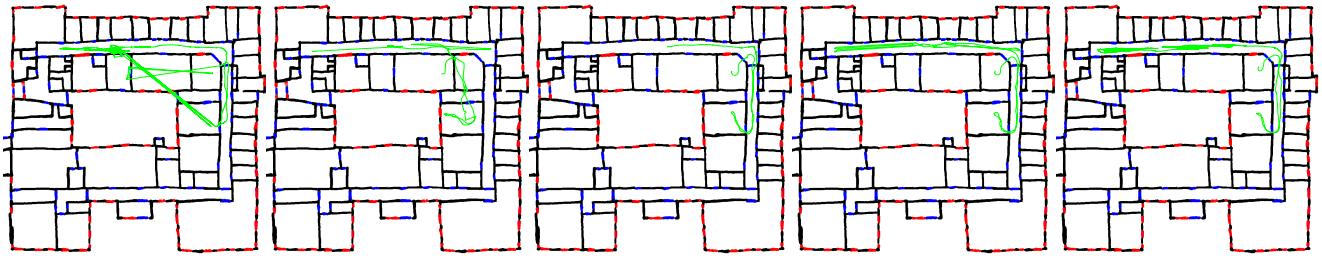
(a) AMCL Path 1 Incorrect Localisation (b) AMCL Path 2: Initialisation Fail (c) AMCL Path 3: Incorrect Localisation (d) AMCL Path 4: Initialisation Fail (e) AMCL Path 5: Initialisation Fail

Fig. 20: Estimated AMCL path from global initialisations.



(a) SeDAR Range Path 1: Initialisation Fail (b) SeDAR Range Path 2: Initialisation Fail (c) SeDAR Range Path 3: Correct (d) SeDAR Range Path 4: Correct (e) SeDAR Range Path 5: Correct

Fig. 21: Estimated SeDAR Range path from global initialisations.



(a) SeDAR Ray Path 1: Correct (b) SeDAR Ray Path 2: Noisy Path (c) SeDAR Ray Path 3: Correct (d) SeDAR Ray Path 4: Correct (e) SeDAR Ray Path 5: Correct

Fig. 22: Estimated SeDAR Ray path from global initialisations.

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	0.313	0.207	0.108	0.235	0.020	1.179
GMapping [19]	0.917	0.817	0.738	0.418	0.121	1.960
ORB_SLAM2 (Mono) [34]	0.756	0.673	0.566	0.343	0.200	1.579
ORB_SLAM2 (RGB-D) [35]	0.161	0.135	0.123	0.088	**0.011**	0.472
Range (Depth Only)	0.384	0.170	0.078	0.345	0.014	2.113
Range (Label Only)	0.154	0.133	0.120	0.079	0.014	0.414
Range (Combined)	**0.115**	**0.088**	**0.060**	**0.074**	0.016	**0.349**
Rays ($(\epsilon_a = 3.0)$)	0.436	0.340	0.274	0.272	0.045	1.522
Rays ($(\epsilon_a = 5.0)$)	0.256	0.191	0.148	0.170	0.012	1.007

Table 7: TUM-RGBD Pioneer_360

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	0.162	0.145	0.140	**0.073**	0.020	0.383
GMapping [19]	0.900	0.717	0.581	0.544	0.024	2.433
ORB_SLAM2 (Mono) [34]	2.076	1.948	1.969	0.717	0.168	3.645
ORB_SLAM2 (RGB-D) [35]	0.287	0.269	0.262	0.100	0.018	0.718
Range (Depth Only)	0.157	0.137	0.122	0.076	0.017	0.424
Range (Label Only)	0.181	0.161	0.147	0.084	0.016	0.388
Range (Combined)	**0.150**	**0.129**	**0.118**	0.077	**0.002**	**0.344**
Rays ($(\epsilon_a = 3.0)$)	0.783	0.595	0.423	0.508	0.109	2.222
Rays ($(\epsilon_a = 5.0)$)	0.911	0.782	0.651	0.468	0.021	1.977

Table 8: TUM-RGBD Pioneer_Slam

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	**0.147**	**0.123**	**0.110**	0.080	**0.007**	**0.373**
GMapping [19]	1.063	0.857	0.635	0.629	0.118	2.703
ORB_SLAM2 (Mono) [34]	1.442	1.317	1.157	0.586	0.326	2.194
ORB_SLAM2 (RGB-D) [35]	0.166	0.150	0.125	0.071	0.070	0.345
Range (Depth Only)	0.175	0.166	0.165	0.055	0.036	0.357
Range (Label Only)	0.287	0.244	0.228	0.151	0.061	1.651
Range (Combined)	0.160	0.145	0.130	**0.068**	0.008	0.338
Rays ($(\epsilon_a = 3.0)$)	1.707	1.465	1.184	0.877	0.040	3.817
Rays ($(\epsilon_a = 5.0)$)	1.571	1.318	1.004	0.855	0.024	3.643

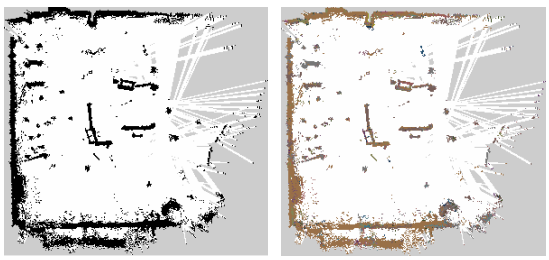
Table 9: TUM-RGBD Pioneer_Slam2

Average Trajectory Error (m)						
Approach	**RMSE**	**Mean**	**Median**	**Std. Dev.**	**Min**	**Max**
AMCL [10]	0.163	0.132	0.105	0.095	0.003	0.440
GMapping [19]	0.967	0.851	0.758	0.460	0.019	1.762
ORB_SLAM2 (Mono) [34]	2.036	1.886	1.978	0.767	0.240	3.379
ORB_SLAM2 (RGB-D) [35]	0.164	0.134	**0.101**	0.094	0.033	0.404
Range (Depth Only)	0.118	0.096	0.082	0.068	0.005	0.346
Range (Label Only)	0.163	0.145	0.127	**0.073**	0.027	0.419
Range (Combined)	**0.143**	**0.121**	0.122	0.077	**0.010**	**0.397**
Rays ($(\epsilon_a = 3.0)$)	1.394	1.236	1.359	0.645	0.109	2.980
Rays ($(\epsilon_a = 5.0)$)	3.717	3.581	3.754	0.998	1.699	5.131

Table 10: TUM-RGBD Pioneer_Slam3

This means that SeDAR performs much better than monocular SLAM.

Similarly, in Pioneer_Slam2 AMCL outperforms overall. This is due to the fact that errors in the semantic segmentation network make the label estimates in SeDAR noisy.



(a) Occupancy Map (b) Semantic Map

Fig. 23: Maps created using ground truth poses and scan data.

Comparably, the depth estimates from an RGB-D camera are less noisy. This is a known limitation of this approach, as we do not fine-tune the network to any scenario. This is further evidenced by the much-higher error on the ray-based approach in this sequence, where the labels are the only cue available for localisation. However, semantic segmentation is a fast moving field, and improvements to the segmentation would quickly translate to increased performance for SeDAR.

7.5 Timing

The approach presented here makes the conscious decision to collapse the 3D world into a 2D representation. This has very noticeable effects to the computational complexity, and therefore speed, of the approach.

The speed of this approach was evaluated on a machine equipped with an Intel Xeon X5550 (2.67GHz) and an NVidia Titan X (Maxwell). OpenMP was used for threading expensive for-loops (such as the raycasting). During room-level initialisation, or once the system has converged, the approach can run with 250 particles in 10ms, leaving more than enough time to process the images from the Kinect into a SeDAR scan. Transforming the RGB images into semantic labels is the most extensive operation, taking on average 120ms. This means that a converged filter can run at $8-10$ fps. When performing global localisation, the approach can integrate a new sensor update, using $50,000$ particles, in 2.25 seconds. This delay does not impact the ability of the system to converge, as most MCL approaches require motion between each sensor integration, meaning the effective rate is much lower than the sensor output.

8 Conclusion

In conclusion, this work has presented a novel approach that is capable of localising a robotic platform within a known floorplan using human-inspired techniques. First, the semantic information that is naturally present and salient in a floorplan was extracted. The first novelty was using the semantic information present in a standard RGB image to extract labels and present them as a new sensing modality called SeDAR. The semantic information present in the floorplan and the SeDAR scan were then used in a SeDAR-based MCL approach. This approach then presented three main novelties. In the first, the semantic information present in the floorplan was used to define a novel motion model for MCL. In the second, the SeDAR scan was used to localise in a floorplan using a combination of range and label information. In the third, SeDAR was used in the absence of range data to localise in the floorplan using only an RGB image.

These novelties present an important step forward for the state-of-the-art of MCL, and therefore localisation in general. Not only is this work capable of removing the requirement of expensive depth sensors [10, 15], it also has the ability to improve the performance of localisation approaches that use depth sensors [52]. When compared against the state-of-the-art monocular approaches [7, 37], leveraging the semantic information present in an RGB image allows less accurate maps to be used by utilising other information present in the map. Taken together, these contributions open the door for the usage of maps designed for human use. This implies that localisation as a discrete process to reconstruction becomes a viable alternative, as pre-existing floorplans can be used to localise while the 3D structure is reconstructed. The advances presented in this paper make it clear that the use of semantic information to aid localisation is the next step for the field.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building Rome in a Day. Communications of the ACM **54**(10), 105–112 (2011). DOI 10.1145/2001269.2001293
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv pp. 1–14 (2015). DOI 10.1109/TPAMI.2016.2644615. URL http://arxiv.org/abs/1511.00561
3. Bedkowski, J.M., Röhling, T.: Online 3D LIDAR Monte Carlo localization with GPU acceleration. Industrial Robot: An International Journal **44**(4), 442–456 (2017). DOI 10.1108/IR-11-2016-0309. URL http://www.emeraldinsight.com/doi/10.1108/IR-11-2016-0309
4. Borgefors, G.: Distance Transformations in Digital Images. Computer Vision, Graphics, and Image Processing **34**(3), 334–371 (1986)
5. Brubaker, M.A., Geiger, A., Urtasun, R.: Lost! leveraging the crowd for probabilistic visual self-localization. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3057–3064. IEEE (2013). DOI 10.1109/CVPR.2013.393
6. Caselitz, T., Steder, B., Ruhnke, M., Burgard, W.: Monocular camera localization in 3D LiDAR maps. In: International Conference on Intelligent Robots and Systems (IROS), pp. 1926–1931. IEEE/RSJ (2016). DOI 10.1109/IROS.2016.7759304
7. Chu, H., Kim, D.K., Chen, T.: You are here: Mimicking the Human Thinking Process in Reading Floor-Plans. In: International Conference on Computer Vision (ICCV), pp. 2210–2218 (2015). DOI 10.1109/ICCV.2015.255
8. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. Transactions on Graphics **36**(3) (2017). DOI 10.1145/nnnnnnn.nnnnnnn. URL http://arxiv.org/abs/1604.01093
9. Dellaert, F., Burgard, W., Fox, D., Thrun, S.: Using the Condensation algorithm for robust, vision-based mobile robot localization. In: Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, p. 594 (1999). DOI 10.1109/CVPR.1999.784976
10. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte Carlo localization for mobile robots. In: International Conference on Robotics and Automation (ICRA), May, pp. 1322–1328. IEEE, Detroit (1999). URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=772544
11. Durrant-Whyte, H.: Uncertain geometry in robotics. Robotics and Automation **4**(I), 23–31 (1988). URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=768 http://ieeexplore.ieee.org/xpls/abs%7B_%7Dall.jsp?arnumber=768
12. Durrant-Whyte, H.F., Dissanayake, M.W.M.G., Gibbens, P.W.: Towards deployment of large scale simultaneous localisation and map building (SLAM) systems. In: International Symposium of Robotics Research, February, pp. 121–127 (1999)
13. Engel, J., Sturm, J., Cremers, D.: Semi-Dense Visual Odometry for a Monocular Camera. In: International Conference on Computer Vision (ICCV), pp. 1449–1456. IEEE (2013). DOI 10.1109/ICCV.2013.183. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6751290
14. Fallon, M.F., Johannsson, H., Leonard, J.J.: Efficient scene simulation for robust monte carlo localization using an RGB-D camera. In: International Conference on Robotics and Automation (ICRA), pp. 1663–1670. IEEE (2012). DOI 10.1109/ICRA.2012.6224951
15. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In: Association for the Advancement of Artificial Intelligence (AAAI), pp. 343–349 (1999). DOI 10.1.1.2.342. URL http://dl.acm.org/citation.cfm?id=315322
16. Furukawa, Y., Ponce, J.: Accurate, Dense, and Robust Multi-View Stereopsis. Pattern Analysis and Machine Intelligence (PAMI) **32**(8), 1362–1376 (2010)
17. Galliani, S., Schindler, K.: Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In: International Conference on Computer Vision (ICCV) (2015)
18. Garage, W.: TurtleBot. Open Source Robotics Foundation. http://www.turtlebot.com/. URL http://www.turtlebot.com/
19. Grisetti, G., Stachniss, C., Burgard, W.: Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. IEEE Transactions on Robotics **23**(1), 34–46 (2007). DOI 10.1109/TRO.2006.889486
20. Holder, C.J., Breckon, T.P., Wei, X.: From On-Road to Off: Transfer Learning Within a Deep Convolutional Neural Network for Segmentation and Classification of Off-Road Scenes pp. 149–162 (2016). DOI 10.1007/978-3-319-46604-0_11. URL http://link.springer.com/10.1007/978-3-319-46604-0_11
21. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A **4**(4) (1987). DOI 10.1364/JOSAA.4.000629. URL https://www.osapublishing.org/abstract.cfm?URI=josaa-4-4-629
22. Huitl, R., Schroth, G., Hilsenbeck, S., Schweiger, F., Steinbach, E.: TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In: Proc. of the International Conference on Image Processing. Orlando, FL, USA (2012). URL http://navvis.de/dataset. Dataset available at http://navvis.de/dataset
23. Kanai, S., Hatakeyama, R., Date, H.: Improvement of 3D Monte Carlo Localization Using a Depth Camera and Terrestrial Laser Scanner. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XL-4/W5**(May), 61–66 (2015). DOI 10.5194/isprsarchives-XL-4-W5-61-2015. URL http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-4-W5/61/2015/
24. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. arXiv (2015). URL http://arxiv.org/abs/1511.02680
25. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE international conference on computer vision, pp. 2938–2946 (2015)

26. Labbe, M., Michaud, F.: Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM. In: International Conference on Intelligent Robots and Systems (IROS), pp. 2661–2666. IEEE/RSJ, IEEE/RSJ, Chicago (2014)
27. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: International Conference on 3D Vision (3DV), pp. 239–248 (2016). DOI 10.1109/3DV.2016.32
28. Liu, C., Schwing, A.G., Kundu, K., Urtasun, R., Fidler, S.: Rent3D: Floor-plan priors for monocular layout estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3413–3421. IEEE (2015). DOI 10.1109/CVPR.2015.7298963
29. McCormac, J., Clark, R., Bloesch, M., Davison, A., Leutenegger, S.: Fusion++: Volumetric object-level slam. In: 2018 International Conference on 3D Vision (3DV), pp. 32–41. IEEE (2018)
30. Melbouci, K., Collette, S.N., Gay-Bellile, V., Ait-Aider, O., Dhome, M.: Model based RGBD SLAM. In: International Conference on Image Processing (ICIP), vol. 2016-Augus, pp. 2618–2622 (2016). DOI 10.1109/ICIP.2016.7532833
31. Mendez, O., Hadfield, S., Pugeault, N., Bowden, R.: Next-Best Stereo: Extending Next-Best View Optimisation For Collaborative Sensors. In: British Machine Vision Conference (BMVC). BMVA Press. (Oral), York, UK (2016). URL http://personal.ee.surrey.ac.uk/Personal/S.Hadfield/papers/Mendez16.pdf
32. Mendez, O., Hadfield, S., Pugeault, N., Bowden, R.: Taking the Scenic Route to 3D : Optimising Reconstruction from Moving Cameras. In: International Conference on Computer Vision (ICCV). IEEE, Venice, Italy (2017). URL http://cvssp.org/Personal/OscarMendez/papers/pdf/Mendez17.pdf
33. Mendez, O., Hadfield, S., Pugeault, N., Bowden, R.: SeDAR - Semantic Detection and Ranging: Humans can localise without LiDAR, can robots? In: International Conference on Robotics and Automation (ICRA). IEEE., Brisbane, Australia (2018)
34. Mur-Artal, R., Montiel, J., Tardós, J.D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics **31**(5), 1147–1163 (2015). DOI 10.1109/TRO.2015.2463671
35. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017)
36. Murphy, K.: Bayesian Map Learning in Dynamic Environments. In: Conference on Neural Information Processing Systems (NIPS), vol. 12, pp. 1015–1021 (2000). URL http://cs.krisbeevers.com/research/research%7B_%7Dqual/10-murphy00.pdf http://cs.krisbeevers.com/research/research_qual/10-murphy00.pdf
37. Neubert, P., Schubert, S., Protzel, P.: Sampling-based Methods for Visual Navigation in 3D Maps by Synthesizing Depth Images. In: International Conference on Intelligent Robots and Systems (IROS) (2017)
38. Poschmann, J., Neubert, P., Schubert, S., Protzel, P.: Synthesized Semantic Views for Mobile Robot Localization. In: European Conference on Mobile Robotics (ECMR), pp. 403–408. IEEE, Paris (2017)
39. Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., Pajdla, T.: Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
40. Schneider, T., Dymczyk, M., Fehr, M., Egger, K., Lynen, S., Gilitschenski, I., Siegwart, R.: maplab: An open framework for research in visual-inertial mapping and localization. IEEE Robotics and Automation Letters **3**(3), 1418–1425 (2018)
41. Shelhamer, E., Long, J., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. Pattern Analysis and Machine Intelligence (PAMI) **39**(4), 640–651 (2017). DOI 10.1109/TPAMI.2016.2572683
42. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2930–2937. IEEE (2013). DOI 10.1109/CVPR.2013.377
43. Smith, R., Cheeseman, P.: On the representation and estimation of spatial uncertainty. International Journal of Robotics Research **5**(4), 56–68 (1987). URL http://ijr.sagepub.com/content/5/4/56.short
44. von Stumberg, L., Usenko, V., Engel, J., Stückler, J., Cremers, D.: From Monocular SLAM to Autonomous Drone Exploration (2016). URL http://arxiv.org/abs/1609.07835
45. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: International Conference on Intelligent Robots and Systems (IROS), pp. 573–580. IEEE/RSJ (2012). DOI 10.1109/IROS.2012.6385773
46. Thrun, S.: A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots. International Journal of Robotics Research **20**(5), 335–363 (2001). DOI 10.1177/02783640122067435. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0035327477&partnerID=tZOtx3y1
47. Thrun, S.: Probabilistic robotics. Communications of the ACM **45**(3) (2002). DOI 10.1145/504729.504754. URL http://portal.acm.org/citation.cfm?doid=504729.504754
48. Thrun, S., Burgard, W., Fox, D.: Monte Carlo Localisation. In: Probabilistic Robotics, chap. 8.3, pp. 238–267. MIT Press, Cambridge, Massachussets (2006)
49. Thrun, S., Fox, D., Burgard, W., Dellaert, F.: Robust Monte Carlo Localization for Mobile Robots. Artificial Intelligence **128**(1-2), 99–141 (2001). DOI http://dx.doi.org/10.1016/S0004-3702(01)00069-8
50. Walch, F., Hazirbas, C., Leal-Taix, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using lstms for structured feature correlation. In: ICCV (2017). URL https://github.com/NavVisResearch/NavVis-Indoor-Dataset
51. Wang, S., Fidler, S., Urtasun, R.: Lost Shopping! Monocular Localization in Large Indoor Spaces. In: International Conference on Computer Vision (ICCV), pp. 2695–2703 (2015). DOI 10.1109/ICCV.2015.309
52. Winterhalter, W., Fleckenstein, F., Steder, B., Spinello, L., Burgard, W.: Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans. In: International Conference on Intelligent Robots and Systems (IROS), vol. 2015-Decem, pp. 3138–3143. IEEE/RSJ (2015). DOI 10.1109/IROS.2015.7353811
53. Xiao, J., Owens, A., Torralba, A.: SUN3D: A database of big spaces reconstructed using SfM and object labels. In: International Conference on Computer Vision (ICCV), pp. 1625–1632 (2013). DOI 10.1109/ICCV.2013.458