UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CRISTIAN LOPES

Monocular depth estimation applied to global localization over 2D floor plans

Dissertation presented in partial fulfillment of the requirements for the degree of Master of Computer Science

Advisor: Profª. Drª. Mariana Luderitz Kolberg
Co-advisor: Prof. Dr. Renan de Queiroz Maffei

Porto Alegre
June 2024

CIP — CATALOGING-IN-PUBLICATION

Lopes, Cristian

Monocular depth estimation applied to global localization over 2D floor plans / Cristian Lopes. – Porto Alegre: PPGC da UFRGS, 2024.

61 f.: il.

Dissertation (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2024. Advisor: Mariana Luderitz Kolberg; Co-advisor: Renan de Queiroz Maffei.

1. Mobile Robots. 2. Indoor Localization. 3. Monocular Depth Estimation. 4. Free Space Density. I. Kolberg, Mariana Luderitz. II. Maffei, Renan de Queiroz. III. Título.

“It’s not that I’m so smart,
it’s just that I stay with problems longer.”
— Albert Einstein

AGRADECIMENTOS

Em primeiro lugar, agradeço à minha família, especialmente à minha mãe, pelo apoio incondicional, palavras de incentivo e carinho não apenas durante a realização deste trabalho, mas ao longo de toda a minha vida.

Agradeço à Profª. Drª. Mariana Luderitz Kolberg e ao Prof. Dr. Renan de Queiroz Maffei, por dedicarem seu tempo e esforço na orientação e supervisão deste trabalho. Seu apoio e orientação foram fundamentais para o desenvolvimento deste estudo e para meu crescimento acadêmico.

Por fim, agradeço aos colegas que me ajudaram e inspiraram na busca por conhecimento durante o período do mestrado. Em especial, agradeço ao Leoni M. Loris por me acompanhar nesta jornada de idas e vindas de um mestrado com dedicação parcial, e por ter enriquecido essa experiência com sua notável capacidade intelectual.

ABSTRACT

Indoor global localization is a critical aspect of autonomous robotic navigation. The increasing demand for service consumer-grade robots that require self-localization calls for research on methods that work with easy setup and low-cost sensors. In this work, we propose a monocular camera-based localization of a motorized wheeled robot using a 2D floor plan as a reference map. The innovation of our method lies in using depth maps estimated from monocular images to compute the free space around the robot to be used as a measurement model in a particle filter strategy. The estimated free space density is compared to the free space density extracted from particles in the 2D floor plan. Due to the inherent imperfections of estimated depth maps, we also propose a new particle weighting approach to account for uncertainties in the depth estimation from the monocular camera. Experiments performed using real-world scenario sequences of images comparing the proposed method with RGB-D camera-based approaches demonstrate the effectiveness of the method, even for imperfect depth maps obtained with the monocular depth estimation model.

Keywords: Mobile Robots. Indoor Localization. Monocular Depth Estimation. Free Space Density.

Monocular depth estimation applied to global localization over 2D floor plans

RESUMO

A localização global em ambientes internos é um aspecto crucial da navegação de robôs autônomos. A crescente demanda por robôs de serviço, que exigem auto-localização, impulsiona a pesquisa de métodos que sejam fáceis de configurar e que utilizem sensores de baixo custo. Neste trabalho, apresentamos uma proposta de localização baseada em câmera monocular para um robô de rodas motorizadas, utilizando um mapa de planta baixa 2D como referência. A inovação de nosso método reside na utilização de mapas de profundidade estimados a partir de imagens monoculares para calcular o espaço livre ao redor do robô, a ser usado como modelo de observação em uma estratégia de filtro de partículas. A densidade de espaço livre estimada é comparada com a densidade de espaço livre extraída das partículas no plano de planta baixa 2D. Devido às imperfeições inerentes dos mapas de profundidade estimados, propomos também uma nova abordagem de ponderação de partículas para considerar as incertezas na estimativa de profundidade da câmera monocular. Experimentos realizados com sequências de imagens do mundo real, comparando o método proposto com abordagens baseadas em câmera RGB-D, demonstram a eficácia do método, mesmo para mapas de profundidade imperfeitos obtidos com o modelo de estimativa de profundidade monocular.

Palavras-chave: Robótica Móvel. Localização em Ambientes Internos. Estimação Monocular de Profundidade. Densidade de Espaço Livre.

LIST OF ABBREVIATIONS AND ACRONYMS

CNN	Convolutional Neural Network
FSD	Free Space Density
GAN	Generative Adversarial Network
HIMM	Histogramic In-Motion Mapping
LIDAR	Light Detection And Ranging
MCL	Monte Carlo Localization
RNN	Recurrent Neural Network
SLAM	Simultaneous Localization and Mapping
ViT	Vision Transformers
WSN	Wireless Sensor Network

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CONTENTS

1 INTRODUCTION

Autonomous mobile robots have the potential to produce huge economic benefits in laboratories, industry, warehousing and logistics, transportation, retail, entertainment, and other fields (HUANG et al., 2023). They can replace humans in manual tasks such as repetitive labor and dangerous operations. Localization is a prerequisite for enabling autonomous robots to operate effectively in various environments. Particularly, service robots require reliable localization because they operate in environments where human interaction and safety are crucial.

Global localization in indoor environments using low-cost sensors is still an open challenge for various applications requiring autonomous robots (HUANG et al., 2023). The global localization problem consists of estimating the pose of a robot in a reference map with no prior knowledge of the initial pose. Solutions to such problem have to handle sensor uncertainties, accurately represent scenarios, and effectively differentiate ambiguities in the environment. The ability to accurately and autonomously determine one's position is crucial for enabling intelligent machines to interact effectively with their surroundings, make informed decisions, and navigate through complex environments.

Some of the keys to democratizing consumer-grade robotics applications are the use of inexpensive sensors and ease of setup (BONIARDI et al., 2019b). Thus, the appropriate choice of the sensor and representation of the map is very important to a broad diffusion of consumer robots. Typically, SLAM techniques do not fit these requirements due to their hard setup with the necessity of a previous collection of sensor measurements to build a coherent representation of the environment. Also, it usually requires highly accurate maps built with the same sensor method employed for robot localization (BONIARDI et al., 2019b). An alternative is using already available map representations, for instance, floor plans, which are generally available upfront for indoor structures and consist of a robust representation of the environment (ITO et al., 2014).

Monocular cameras are widely available and commonly found in various consumer devices such as smartphones, laptops, tablets, and even wearable devices. Hence, they are a practical sensor option for addressing the challenge of global localization for consumers. Moreover, monocular depth estimation has progressed considerably recently due to advancements in deep learning techniques and the availability of larger datasets (KHAN; SALAHUDDIN; JAVIDNIA, 2020). Although advancements were achieved, limitations still exist in the depth maps obtained from monocular depth estimation, which

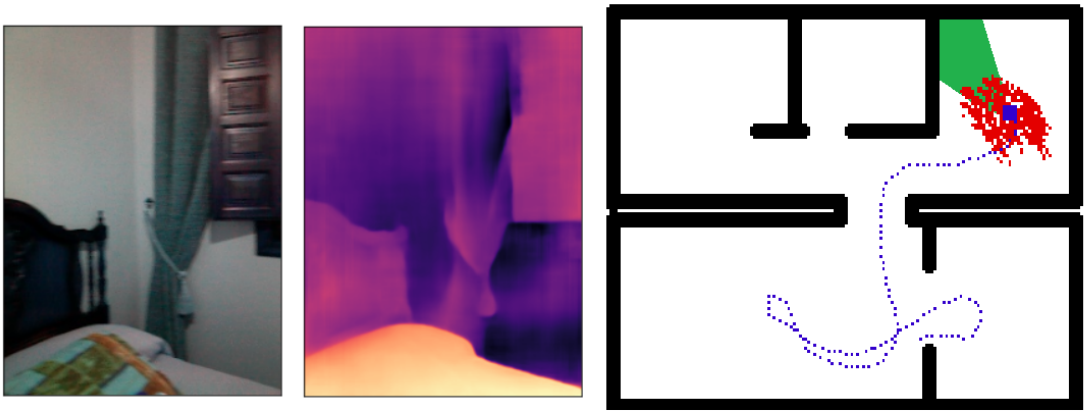
include scale ambiguity and handling occlusions.

1.1 Goal and Contributions

Due to the limitations that still exist in monocular depth estimation, when using depth maps produced with monocular cameras for robot localization, the representation of the sensor measurements has to account for uncertainties. In (MAFFEI et al., 2015), authors proposed using the Free Space Density (FSD) as the observation model, which measures the free space around the robot. The FSD can be computed consistently across different sensor types like 2D laser rangefinders (MAFFEI et al., 2015), omnidirectional cameras (RIBACKI et al., 2018), or RGB-D cameras (MAFFEI et al., 2020). Its versatility allows an efficient incorporation of the depth maps uncertainty in the observation model. Given the consistent computation of FSD across various sensor types and advancements in monocular depth estimation techniques, we conjecture that the integration of sufficiently accurate monocular depth maps with FSD as an observation model can enable effective robot localization in diverse environments.

In this work, we address the need for practical solutions utilizing low-cost sensors in the challenge of global localization in indoor environments for autonomous robots. We propose to use monocular depth maps to compute the observation model in a Monte Carlo Localization (MCL) approach. Estimated depth maps obtained from monocular images usually present high uncertainty influenced by multiple factors related to the algorithm, scene complexity, camera properties, and environmental conditions. We selected Free-Space Density (FSD) as the observation model (MAFFEI et al., 2020) due to its simplicity and robustness against noisy sensor measurements. Figure 1.1 presents a visualization of the method, where a monocular camera obtains the input image which is then used to predict a depth map with a monocular depth estimation model. The depth map is used to estimate the free space around the robot which is applied in an MCL approach for robot localization.

We evaluate the monocular depth estimation model performance against RGB-D ground truth data for the experiments dataset. Additionally, we evaluate how MCL behaves for noisy FSD measurements to assess the observation model choice. We also estimate the uncertainty in the FSD computation due to monocular depth maps scale ambiguity. Given the preliminary results, we propose a new approach for particle weighting using FSD to account for monocular depth estimation uncertainty. The main contributions



(a) (b) (c)

Figure 1.1 – Monocular depth estimation used to compute the observation model in an MCL approach. (a) Image obtained by the robot using the monocular camera (b) Monocular depth estimation model prediction from image (c) Floor plan presenting robot trajectory and position in blue, particles in red, and robot field of view in green.

of this work can be summarized as follows:

- A global localization method over 2D floor plans based on monocular depth estimation. This novel approach leverages the use of low-cost sensors and readily available floor plans. This combination makes localization systems accessible and affordable, offering a cost-effective solution for global localization tasks.
- New particle weighting strategy using FSD for noisy sensor measurements. By incorporating sensor uncertainty into the particle weighting process, our method achieves improved convergence towards the accurate robot position.

1.2 Organization

This work is organized as follows. In Section 2, we present a background of mobile robotics focusing on global localization. We also present a detailed explanation of FSD and its use to solve the global localization problem. Section 3 presents the challenges and deep learning research in monocular depth estimation. The model used to compute the depth maps is also presented. Concluding the background, Section 4 presents the related work in global localization and FSD.

The proposed method using the monocular depth estimation model to compute FSD is presented in Section 5. A detailed explanation of the new particle weighting strategy is also presented. Section 6 presents the experiments performed to evaluate the use of the monocular depth estimation model in solving the global localization problem.

Initially, preliminary experiments evaluating the robustness of FSD against noisy measurements are presented. Then, we present an evaluation of the proposed method against methods proposed by other authors. Finally, our conclusions and future work are presented in Section 7.

2 MOBILE ROBOTICS BACKGROUND

This section presents the global localization problem in the field of robotics and explores some proposed solutions. Among these solutions, particle filters stand out as a promising approach, leveraging probabilistic techniques to estimate a robot's location within its environment. Additionally, the role of map representation is presented, focusing on how it contributes to enhancing a robot's ability to navigate and comprehend complex environments. Moreover, this section discusses the importance of observation models, crucial in enhancing the robot's understanding of its surroundings.

2.1 Global localization

Robot localization in real-world scenarios aims to accurately determine a robot's position and orientation within its operational environment based on the sequence of observations and robot motion. Most methods model the pose estimate as a probability distribution over the space of solutions (THRUN; BURGARD; FOX, 2005). The estimated pose is computed through a Bayesian approach. The localization problem is described in Equation (2.1) as a posterior probability $bel(x_t)$ over the state x_t at time t, given all previous measurements $z_{1:t}$ and all prior control inputs $u_{1:t}$.

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{2.1}$$

The Bayes Filter is the most general algorithm for calculating the belief $bel(x_t)$ from measurement and control data (THRUN; BURGARD; FOX, 2005). The solution is computed recursively, which means, the belief $bel(x_t)$ at time t is calculated from the belief $bel(x_{t-1})$ at time $t-1$. It is computed in 2-steps: control update (*prediction*) and measurement update. The control update step, also known as the prediction step, is presented in Equation (2.2). It computes the prediction $\overline{bel}(x_t)$ based on the prior belief $bel(x_{t-1})$ and the control u_t utilizing the motion model $p(x_t|u_t, x_{t-1})$, representing the probability of the robot being at pose x_t given the control input u_t and the previous pose x_{t-1}.

$$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) \cdot bel(x_{t-1})\, dx \tag{2.2}$$

The measurement update, also known as the correction step, refines the predicted

belief about the robot's pose $\overline{bel}(x_t)$ by incorporating sensor measurements z_t. Equation (2.3) presents the corrected belief computed utilizing the measurement model $p(z_t|x_t)$, representing the probability of observing sensor measurements z_t given the robot's pose x_t. The result is normalized with η to ensure that the result is a valid probability distribution (sum up to 1).

$$bel(x_t) = \eta \cdot p(z_t|x_t) \cdot \overline{bel}(x_t) \tag{2.3}$$

Combining the prediction step (to estimate the new belief without sensor measurements) with the measurement update step (to refine the belief using sensor measurements) forms the basis of the Bayesian filtering approach for robot localization in unknown environments. The filter convergence success relies on an appropriate motion model and measurement model for the given problem. However, achieving success in this convergence faces hurdles due to the inherent uncertainties and variations encountered in real-world environments. Consequently, selecting a suitable map representation, motion model, and measurement model becomes critical for achieving accurate and reliable robot localization.

A concrete implementation of the Bayes Filter for localization requires the definition of three probability distributions: the initial belief $p(x_0)$, the motion model $p(x_t|u_t, x_{t-1})$ and the measurement model $p(z_t|x_t)$. Different methods implement Bayesian filtering among which Kalman Filter is the most popular one (LEONARD; DURRANT-WHYTE, 1991a), which assumes Gaussian noise and linear motion model. On the other hand, Monte Carlo Localization (MCL) (DELLAERT et al., 1999), also known as particle filter, has emerged as a robust implementation for robot localization, proving to be highly effective in various applications, given its simplicity, robustness, and ability to model arbitrary distributions.

2.2 Monte Carlo localization

Particle filter maintains a collection of M samples known as particles to depict the posterior distribution $p(x_t|z_{1:t}, u_{1:t})$ as presented in Equation (2.4).

$$\chi_t = \{p_t^{[1]}, p_t^{[2]}, ..., p_t^{[M]}\} \tag{2.4}$$

where each particle $p_t^{[m]} = \langle x, \omega \rangle$ has a pose x at time t and a weight ω associated

to it.

The MCL algorithm is presented in Algorithm 1, which iteratively estimates the particle set χ_t from the set χ_{t-1} using a *Sampling-Importance-Resampling* (SIR) process. In the first step (*Sampling*), each particle is propagated according to the control u_t by applying the motion model $p(x_t|u_t, x_{t-1})$ over the previous particle pose, i.e. $x(p_{t-1}^{[m]})$, as presented in lines 3-4.

Algorithm 1 Monte Carlo Localization

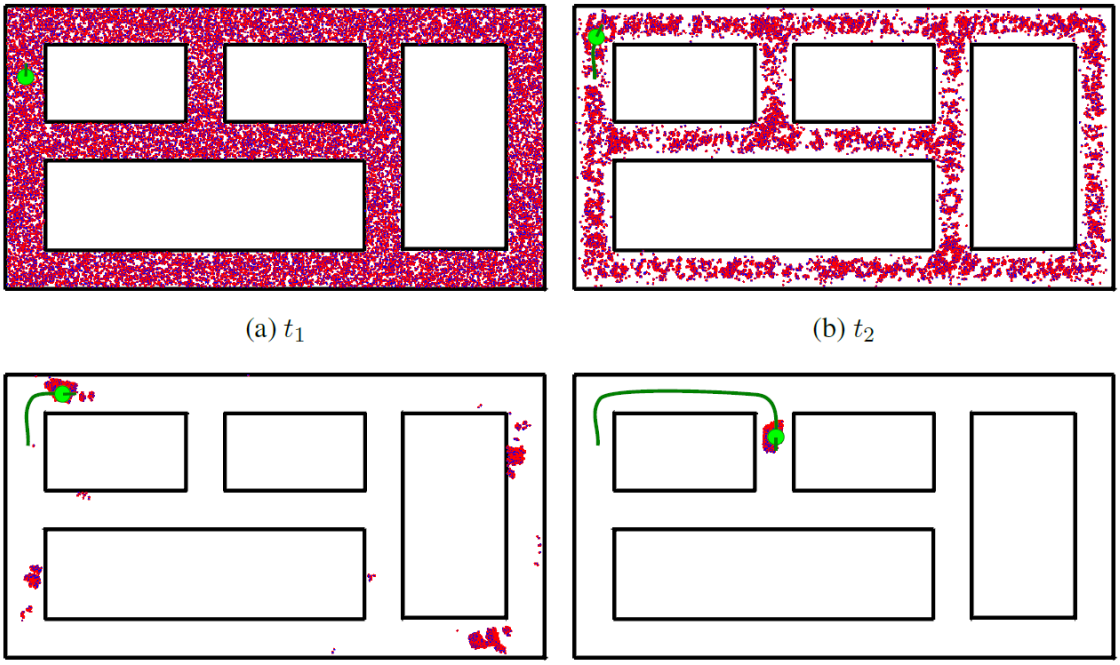
Input: $\chi_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{m}$
Output: χ_t

1: $\overline{\chi}_t = \chi_t = \emptyset$
2: **for** m in $1...M$ **do**
3: $\quad x_{t-1} = x(\mathbf{p}_{t-1}^{[m]})$
4: $\quad$ sample $x \sim p(x_t|\mathbf{u}_t, x_{t-1})$
5: $\quad \omega = p(\mathbf{z}_t|x, \mathbf{m})$
6: $\quad \mathbf{p}_t^{[m]} = \langle x, \omega \rangle$
7: $\quad \overline{\chi}_t = \overline{\chi}_t \cup \{\mathbf{p}_t^{[m]}\}$
8: **end for**
9: **for** m in $1...M$ **do**
10: $\quad$ draw $\mathbf{p}_t^{[}i]$ from $\overline{\chi}_t$ with probability $\propto \omega_t^{[i]}$
11: $\quad \chi_t = \chi_t \cup \{\langle x_t^{[i]}, 1/M \rangle\}$
12: **end for**

The second step of the algorithm (*Importance weighting*) consists of assigning an individual weight ω to each particle. The weight is determined using the observation model, comparing the actual sensor readings with the estimated sensor readings of the particle. The idea is to compute a similarity between the target distribution $p(x_t|z_{1:t}, u_{1:t})$ and the proposal distribution $p(x_t|z_{1:t-1}, u_{1:t})$, generated after the sampling process. The new particles compose a temporary particles' set $\overline{\chi}_t$ with their corresponding pose and weight as presented in lines 5-7.

The third and final stage of the MCL algorithm (*Resampling)* randomly selects, with replacement, the same quantity M of particles from $\overline{\chi}_t$ and incorporates them into a new set χ_t as presented in lines 9-10. One common method for sample selection is the wheel-roulette algorithm, which assigns the probability of selecting each particle $p_t^{[m]}$ in proportion to its weight $\omega(p_t^{[m]})$. Resampling is a critical step in a particle filter as it plays a vital role in aligning the distribution of particles with the true posterior. Typically, during resampling, particles with higher weights are more likely to be duplicated than those with lower weights. This approach enhances the representation of high-weight particles and improves the approximation of the particle distribution to the actual posterior distribution.

It is important to notice that Algorithm 1 presents only a single step of the MCL algorithm. An illustration of multiple steps of the MCL algorithm is presented in Figure 2.1. At the beginning of the process *(a)*, the particles are distributed across all available space since the robot's initial position can be anywhere. In *(b)*, as the robot initiates movement, the level of uncertainty gradually decreases. However, even with a minor displacement, as depicted in (b), the robot's precise location remains unclear. The uncertainty of the position is decreased in *(c)* when the robot performs a right turn, given the limited number of corners of the environment. Finally, when the robot executes another right turn, the remaining ambiguities are resolved, leading to a clearer understanding of the robot's exact location.



(a) t_1 (b) t_2

(c) t_3 (d) t_4

Figure 2.1 – Particle filter convergence in Monte Carlo Localization, where particles are displayed in pink, while the robot and its trajectory are depicted in green. The particles start to converge as the robot undergoes sufficient movement, aiding in the resolution of ambiguities within the environment. Figure adapted from (MAFFEI, 2017).

2.3 Map representation

The construction and representation of the map are some of the challenges in global localization. Many approaches use simultaneous localization and mapping (SLAM) to build the map based on previous sensor measurements. SLAM is often used in conjunction with navigation systems to enable robots to move autonomously within their

environment. For instance, (ALHMIEDAT et al., 2023) proposes a SLAM-based localization and navigation system for service robots. However, since the mapping depends on the robot, this process increases the cost and time of implementation and may also need specialists to guarantee map consistency in complex environments (BONIARDI et al., 2019a). Alternative methods involve leveraging a Wireless Sensor Network (WSN) for robot localization using techniques like triangulation or fingerprinting based on Received Signal Strength (RSS) (ALHMIEDAT, 2023). These approaches also require the setup of the WSN which is a challenge for service robots used in domestic scenarios.

Some approaches use semantic information for localization (ALQOBALI et al., 2024). Semantic information localization relies on recognizing specific objects or features in the environment to determine the robot's position. While this approach can provide rich contextual information, it may struggle in environments where recognizable features are sparse or subject to change. On the other hand, indoor man-made structures usually present a floor plan which may be used as a reference map. Although they do not present objects, like furniture, floor plans describe well the environment structure (MAFFEI et al., 2020). Thus, they are a robust representation even when there are momentary changes in the distribution of objects. Utilizing floor plans as a reference map simplifies the mapping process and reduces the dependency on sensor measurements, making it more cost-effective and less time-consuming.

A commonly used map representation is the occupancy grid map. Occupancy maps are location-based, where a binary occupancy value is assigned to each *x*-*y* coordinate specifying whether or not a given location is occupied with an object (THRUN; BURGARD; FOX, 2005). Cells corresponding to pixels representing walls are set as obstacles, whereas the remaining cells are designated as free space. They serve as an effective and flexible representation for floor plans, offering a balance between granularity and computational efficiency.

Histogramic In-Motion Mapping (HIMM) is an effective mapping technique that employs a two-dimensional Cartesian histogram grid to represent obstacles detected by range finder sensors (BORENSTEIN; KOREN, 1991a). HIMM divides the occupancy probability space into a limited set of integer values and updates the cells' occupancy value along the sensor's acoustic axis through straightforward incrementation and decrementation processes. They can be used to generate local occupancy grid maps from sensor measurements for map representation.

Figure 2.2 presents the HIMM model. In the HIMM model, cells within a grid

have occupancy values ranging from 0 (minimum) to 15 (maximum), totaling 16 possible values. Initially, all cells are set to the minimum value (0) assuming an empty map, or to an intermediate value (8) indicating the initial lack of knowledge about obstacles. During robot navigation, the algorithm translates the current robot pose to the corresponding grid cell. Then, it checks cells within the sensor's perceptual field, considering a maximum sensor range translated into a cell distance. For each cell, it calculates the angular difference with the robot's orientation and identifies the sensor beam used for cell updates. Comparisons are made between cell and sensor distances, determining whether a cell is likely to be occupied or free. Occupied cells are incremented by 3, while free cells are decremented by 1, ensuring values between 0 and 15.

An example of a HIMM-produced map is presented in Figure 2.3. We can observe the algorithm's capability to enhance certainty regarding free spaces and obstacles as the robot moves in the environment. Areas barely observed, like the edges of the sensor's scan, exhibit lower certainty in the generated map.

2.4 Free space density

The selection of the measurement model in MCL is crucial as it directly influences the accuracy and reliability of the robot's pose estimation. Choosing a suitable measurement model in MCL involves considering the sensor characteristics, environmen-

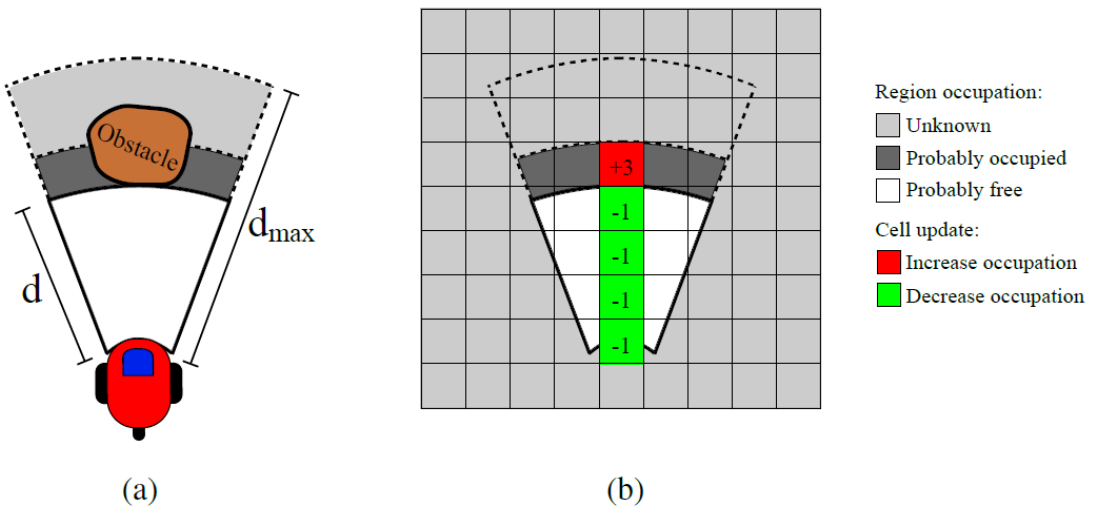
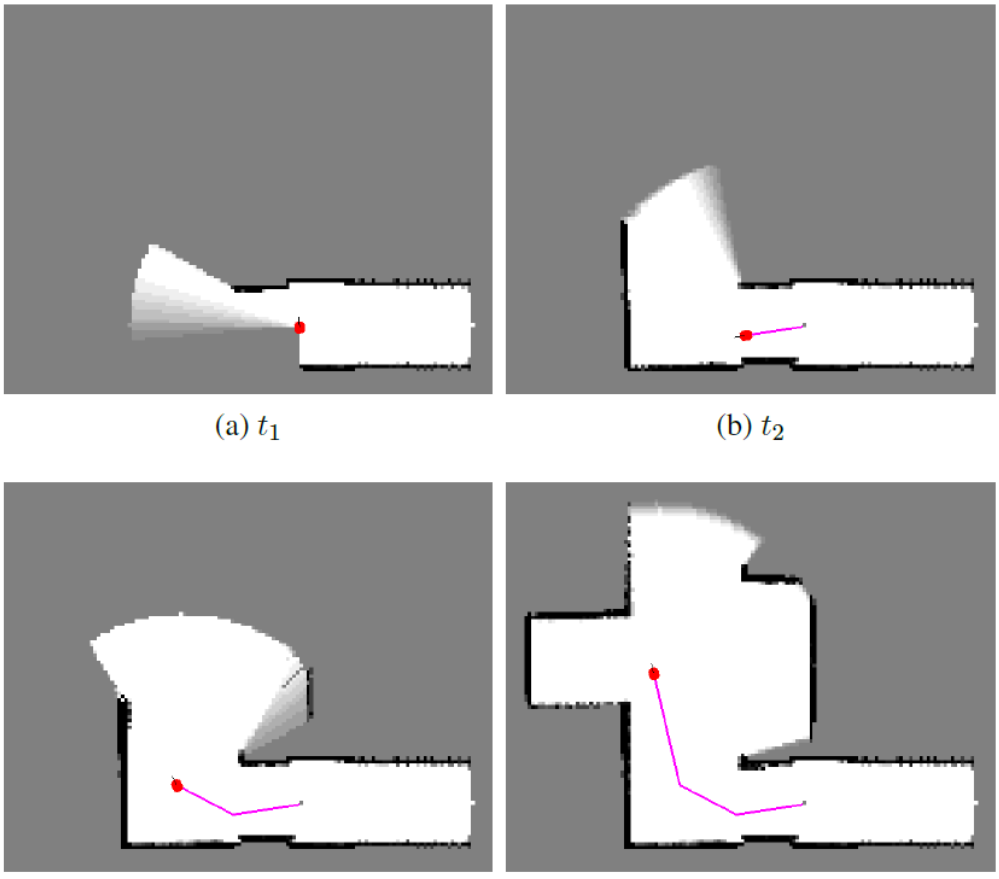


Figure 2.2 – HIMM model: (a) Robot detects an obstacle at distance d. (b) Grid cell update occurs solely along the sensor's axis. In the free region (depicted as white), the occupation of green cells is reduced, while in the occupied region (represented as dark gray), the occupation of red cells is increased. Figure adapted from (MAFFEI, 2017).



(a) t_1 (b) t_2

(c) t_3 (d) t_4

Figure 2.3 – Example of HIMM map generated for a robot moving in the environment. As the robot moves, it collects more evidence about the free-space and obstacles. Figure adapted from (MAFFEI, 2017).

tal conditions, and the specific requirements of the robot's localization task. When using cameras as the sensor, the chosen measurement model should be able to encode data at a lower level concisely and effectively.

Previous works (MAFFEI et al., 2015; RIBACKI et al., 2018; MAFFEI et al., 2020) have been using Free-Space Density (FSD) to represent the observation model to solve the global localization problem in floor plans using MCL. The FSD of a given position is computed for a circular region centered at the position and is defined as the free-space inside such region multiplied by a circular kernel. Given a grid map, the FSD (Ψ) for the region centered at the cell m_0 is a value between 0 and 1 defined by Equation 2.5.

$$\Psi(m_0) = \sum_{m_i} s(m_i, m_o) K(\|(m_i - m_0)\|) \tag{2.5}$$

where $K(.)$ is the circular kernel[1], m_i is a cell inside the kernel region and $s(m_i, m_0)$ is

[1] In this work, a uniform kernel with radius of $1.5m$ is used to compute FSD.

defined by Equation 2.6.

$$s(m_i, m_o) = \begin{cases} 1, & \text{if } m_i \text{ is a } \textit{free-space} \text{ cell and } \textit{visible} \text{ from } m_0 \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

Since the FSD is determined by a scalar value, one of its advantages is the efficient storage of the FSD field in the whole map, which allows a fast query in particle weighting (MAFFEI et al., 2020). The faster the query, the more particles can be used, which covers better the space of solutions and thus increases the probability of the filter converging to the right pose. Figure 2.4 presents the FSD value computed for different robot positions in a given map.

The process of particle weighting is performed by comparing the measured FSD Ψ_{robot} to the FSD value of each particle $\Psi_{particle}$. In previous works (MAFFEI et al., 2015; RIBACKI et al., 2018; MAFFEI et al., 2020), the weight assigned to a particle is given by Equation 2.7, where Δ_Ψ is the difference between the maximum and minimum FSD found in the reference map.

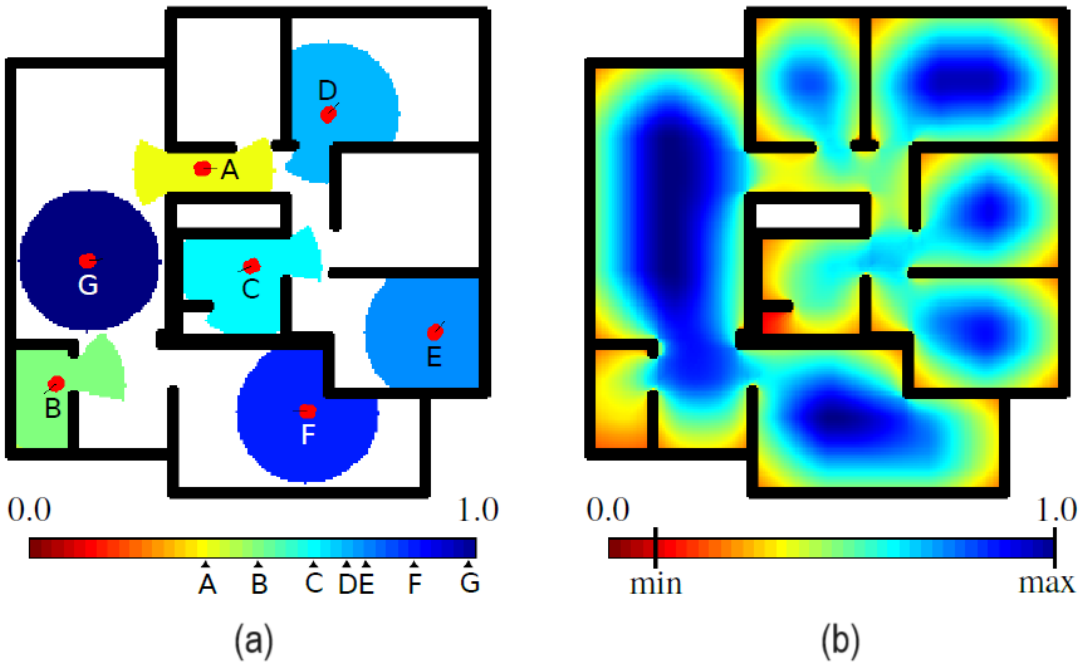


Figure 2.4 – Free-space density computed over a grid map (a) The value of FSD computed for visible cells in a circular kernel centered at the robot placed at different positions (b) The FSD field computed for all cells of the grid map illustrated by different colors from dark red (Ψ=0) and dark blue (Ψ=1). Figure adapted from (MAFFEI et al., 2020).

$$f_{\Psi}(\Psi_{robot}, \Psi_{particle}) = 1.0 - \frac{min(|\Psi_{robot} - \Psi_{particle}|, \Delta_{\Psi})}{\Delta_{\Psi}} \tag{2.7}$$

One of the issues encountered in FSD computation for local maps was the cells inside the kernel which had not been visited by the robot. In (MAFFEI et al., 2020), they proposed to use an Interval FSD $[\Psi(m_0)]$ to account for *unknown* cells in particles' weighting. The Interval FSD of a region centered at m_0 proposed by Maffei *et al.* (MAFFEI et al., 2020) is presented in Equation 2.8.

$$[\Psi(m_0)] = [\underline{\Psi(m_0)}, \overline{\Psi(m_0)}] \tag{2.8}$$

where the infimum $\underline{\Psi(m_0)}$ of the interval corresponds to the definition of the FSD in Equation 2.5 and the supremum $\overline{\Psi(m_0)}$ of the interval is defined by Equation 2.9.

$$\Psi(m_0) = \sum_{m_i} s_{unk}(m_i, m_o) K(\|(m_i - m_0)\|) \tag{2.9}$$

where $s_{unk}(m_i, m_0)$ is defined by Equation 2.10.

$$s_{unk}(m_i, m_o) = \begin{cases} 1, & \text{if } m_i \text{ is a } \textit{free-space} \text{ or } \textit{unknown} \text{ cell} \\ & \text{and } \textit{visible} \text{ from } m_0 \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

When using Interval FSD, the weight ω of particle p, given the particle cell FSD $\Psi(m_p)$ and the robot cell Interval FSD $[\Psi(m_r)]$, is defined in Equation 2.11.

$$\omega(p) = \begin{cases} 1, & \text{if } \Psi(m_p) \in [\Psi(m_r)] \\ f_{\Psi}(\Psi(m_p), \overline{\Psi(m_r)}), & \text{if } \Psi(m_p) > \overline{\Psi(m_r)} \\ f_{\Psi}(\Psi(m_p), \underline{\Psi(m_r)}), & \text{if } \Psi(m_p) < \underline{\Psi(m_r)} \end{cases} \tag{2.11}$$

Considering the suitability of FSD for particle weighting and the possibility of computing it with different sensor modalities, in this work, we propose computing FSD using depth maps predicted from camera images with a pre-trained monocular depth estimation model.

3 MONOCULAR DEPTH ESTIMATION BACKGROUND

Depth estimation is a task in computer vision with many applications in the robotics field, such as SLAM, navigation, object detection and semantic segmentation. Traditional methods of depth estimation, such as, stereo vision and structure from motion are mainly focused on multi-view geometry. These methods have various limitations, including high computational complexity and energy requirements (KHAN; SALAHUDDIN; JAVIDNIA, 2020). Additionally, multi-view geometry methods suffer from the challenge of capturing enough features in the image to match when the scene has less or no texture (MING et al., 2021). Methods that use active sensors, such as RGB-D cameras and LIDAR can get the depth map from a single image. However, RGB-D cameras have limited measurement range and suffer from outdoor sunlight sensitivity, while LIDAR presents high power consumption and high cost (ZHAO et al., 2020). On the other hand, monocular cameras are vastly available to consumers and usually require lower computational and energy demands for depth estimation. Recently, following the advancements in deep-learning techniques and publicly available datasets, the performance of monocular depth estimation methods has improved significantly (KHAN; SALAHUDDIN; JAVIDNIA, 2020).

One of the biggest challenges of applying deep learning to monocular depth estimation is the lack of high-quality labeled data, which is expensive to acquire (ZHAO et al., 2020). Methods that use ground truth depth maps can directly output depth information and learn in a supervised manner, penalizing errors between predictions and ground truth. Due to the challenges in obtaining labeled data, many methods have proposed using semi-supervised and unsupervised frameworks (MING et al., 2021). Semi-supervised approaches require a smaller amount of labeled data combined with a large amount of unlabeled data. Unsupervised methods use geometric constraints from multi-view images obtained from stereo vision or frame sequences (KHAN; SALAHUDDIN; JAVIDNIA, 2020). Usually, supervised methods achieve better performance than unsupervised ones, but since they are trained in a limited amount of data, they do not generalize well to other datasets. Both supervised and unsupervised methods can suffer from scale ambiguity and scale inconsistency, but these issues are particularly prevalent in unsupervised methods, as they usually output disparity, which is only proportional to depth information (ZHAO et al., 2020).

Recent works have tackled different aspects of monocular depth estimation, in-

cluding model architecture, dataset creation, and scale ambiguity. A new model architecture was introduced in (KIM et al., 2018), which proposed a deep variational model for single-image depth estimation by integrating predictions from global and local convolutional neural networks (CNNs), capturing complementary depth attributes. (HAJI-ESMAEILI; MONTAZER, 2024) introduced a novel approach to monocular depth estimation by leveraging depth and surface normal datasets collected from video games, addressing the challenge of acquiring large-scale depth datasets. Scale ambiguity was a subject of study from (GUIZILINI et al., 2023), which introduced ZeroDepth, a novel monocular depth estimation framework capable of predicting metric scale for arbitrary test images across different domains and camera parameters.

To create a model that can capture the diversity of the visual world and work robustly in real scenarios, (RANFTL et al., 2020) have proposed mixing different datasets for training. They have developed a loss function that is invariant to depth range and scales, allowing them to combine data from different sources with diverse sense modalities. In their work, they also evaluated the model in datasets never seen during training, in a *zero-shot cross-dataset transfer* approach. The experiments confirmed the model performs well in a variety of environments even when they were not seen during training. However, since the model is trained in a scale and shift invariant loss, its output is the relative depth information requiring additional steps to obtain metric depth (BHAT et al., 2023).

Besides the datasets and loss functions, the design of dense prediction architectures often follows a pattern that divides the network into an encoder and a decoder. The encoder extracts high-level features from the input image and downsamples the feature maps, while the decoder upsamples the feature maps and produces the final output, such as a depth map. Additionally, to preserve the features of each scale effectively, the corresponding layers of the encoder and decoder are concatenated using skip-connections (MING et al., 2021). Figure 3.1 presents the encoder-decoder pipeline of monocular depth estimation models.

The architecture is usually based on convolutional networks, however, convolutional layers reduce resolution and granularity in deeper feature maps. The loss of resolution is critical for dense prediction where features should be resolved at the input image resolution level for high performance (RANFTL; BOCHKOVSKIY; KOLTUN, 2021). Some authors have also introduced the use of Recurrent Neural Networks (RNNs) in monocular depth estimation to learn temporal features from video sequences. Generative

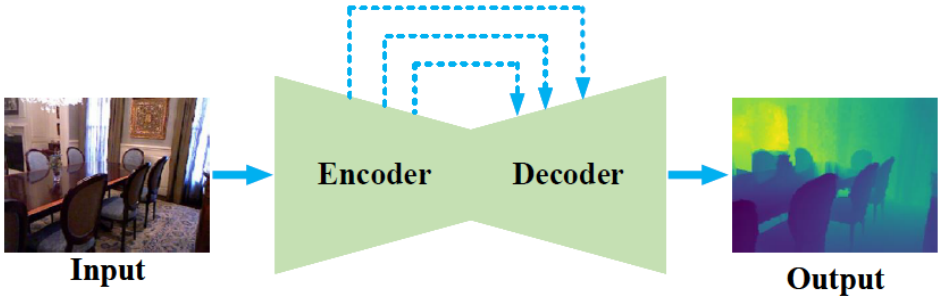


Figure 3.1 – The typical deep learning pipeline for monocular depth estimation involves two main modules. On the left, the encoder network learns depth features layer-by-layer, while on the right, the decoder network reconstructs the depth map. Figure adapted from (DONG et al., 2022).

Adversarial Networks (GANs) were also introduced to produce clearer and more realistic depth maps, due to the challenges in acquiring ground truth depth maps in real-world scenarios (MING et al., 2021).

(RANFTL; BOCHKOVSKIY; KOLTUN, 2021) have proposed the use of vision transformers (ViT) in place of convolutional networks for dense prediction encoding. ViT uses a self-attention mechanism that enables the network to process images at different scales and resolutions. Moreover, they used a Multi-Head Self-Attention mechanism that allows the model to capture complex relationships between different parts of the image. Generally, ViT needs large-scale datasets for training. The combination of the large dataset proposed in (RANFTL et al., 2020) with the architecture proposed in (RANFTL; BOCHKOVSKIY; KOLTUN, 2021) achieved state-of-the-art performance in monocular depth estimation for domestic indoor datasets. The proposed model architecture is presented in Figure 3.2.

Since the model proposed in (RANFTL; BOCHKOVSKIY; KOLTUN, 2021) uses unlabeled data for training, it is unable to output depth metric information, it outputs disparity instead. For a real scenario, its output has to be transformed to metric depth. Ranftl *et al.* (RANFTL et al., 2020) proposed to align the predictions to the ground truth depth based on a least-squares criterion. For a given predicted disparity d and ground truth disparity d^*, they compute a scaled and shifted disparity $\hat{d} = sd + t$, where the scale s and shift t ensure $s(\hat{d}) \approx s(d^*)$ and $t(\hat{d}) \approx t(d^*)$. The least-squares criterion applied to an image with M pixels is presented in Equation 3.1.

$$(s, t) = \underset{s,t}{\operatorname{argmin}} \sum_{i=1}^{M} (sd_i + t - d_i^*)^2 \tag{3.1}$$

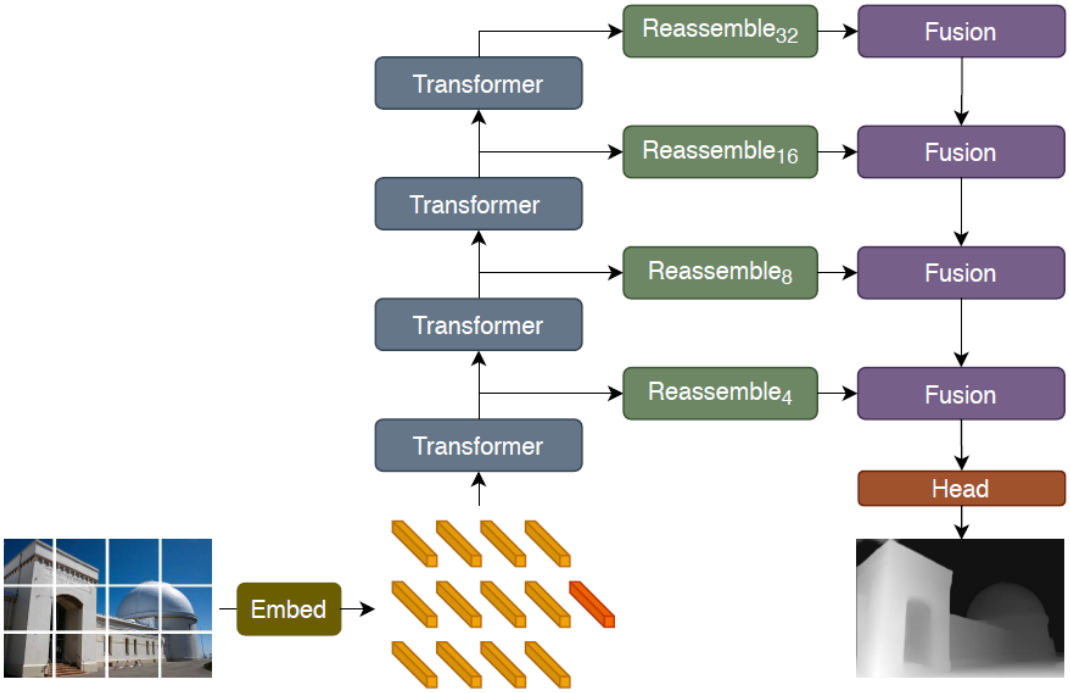


Figure 3.2 – The input image undergoes tokenization (depicted in orange) through the application of a ResNet-50 feature extractor. The image embedding is then enriched with a positional embedding, and a patch-independent readout token (in red) is introduced. These tokens traverse multiple transformer stages. Subsequently, tokens from different stages are reconstituted into an image-like representation at various resolutions (depicted in green). Fusion modules (shown in purple) progressively blend and upsample the representations to generate a finely detailed prediction. Figure adapted from (RANFTL; BOCHKOVSKIY; KOLTUN, 2021).

The possibility of estimating metric depth combined with the capacity of generalization and the state-of-the-art performance in indoor datasets, make DPT-Hybrid (RANFTL; BOCHKOVSKIY; KOLTUN, 2021) a reasonable choice for monocular depth estimation.

4 RELATED WORK

This section presents the related work in robot localization using 2D floor plans as a reference map. It also presents different observation models used in the literature to translate sensor observations to map representation. Additionally, we present how previous researches used the observation models to compute the similarity between the robot observations and observation hypotheses derived from a particular location within the map of the environment.

Several works have proposed the use of RGB-D cameras (ITO et al., 2014; WINTERHALTER et al., 2015; MAFFEI et al., 2020; WATANABE et al., 2020; BONIARDI et al., 2017; BONIARDI et al., 2019a; WANG; MARCOTTE; OLSON, 2019; GAO; KNEIP, 2022) to capture observations of the environment. Usually, these works use measurement models based on distance information. (WINTERHALTER et al., 2015) estimate the 6DoF pose of an RGB-D tablet in a 2D floor plan map using visual-inertial odometry and sparse depth maps extracted from the device. (ITO et al., 2014) used WiFi signal strength to estimate a coarse initial distribution and then applied an MCL strategy using planes extracted from the point cloud projected onto the 2D floor plan. (BONIARDI et al., 2017; BONIARDI et al., 2019a) used 2D LiDARs and a CAD floor plan prior to support long-term localization based on pose graph optimization. (GAO; KNEIP, 2022) also solves the long-term localization problem using LiDARs, but they differ in solving a 6DoF problem with 3D sensors. The main issue with approaches that use RGB-D cameras in a consumer-grade solution is the additional cost and complexity they introduce.

The use of monocular cameras has also been proposed (RIBACKI et al., 2018; BONIARDI et al., 2019b), where they rely mainly on the geometry of edges and boundaries acquired by the camera. (RIBACKI et al., 2018) used a camera pointed at the ceiling to extract boundaries and compute the free space area, which is then used as a measurement model in an MCL strategy. (BONIARDI et al., 2019b) used a convolutional neural network to predict the room layout edges from monocular images and employed an MCL with a sensor model that scores the overlap of the predicted layout edge mask and the expected layout edges generated from a floor plan image. Figure 4.1 presents the edge-extraction network proposed by (BONIARDI et al., 2019b). However, methods that depend on edge and boundaries tend to suffer from occlusions (RIBACKI et al., 2018). In our work, we propose the use of monocular cameras to estimate a depth map from the images, combining the distance information for localization with the high availability of

monocular cameras.

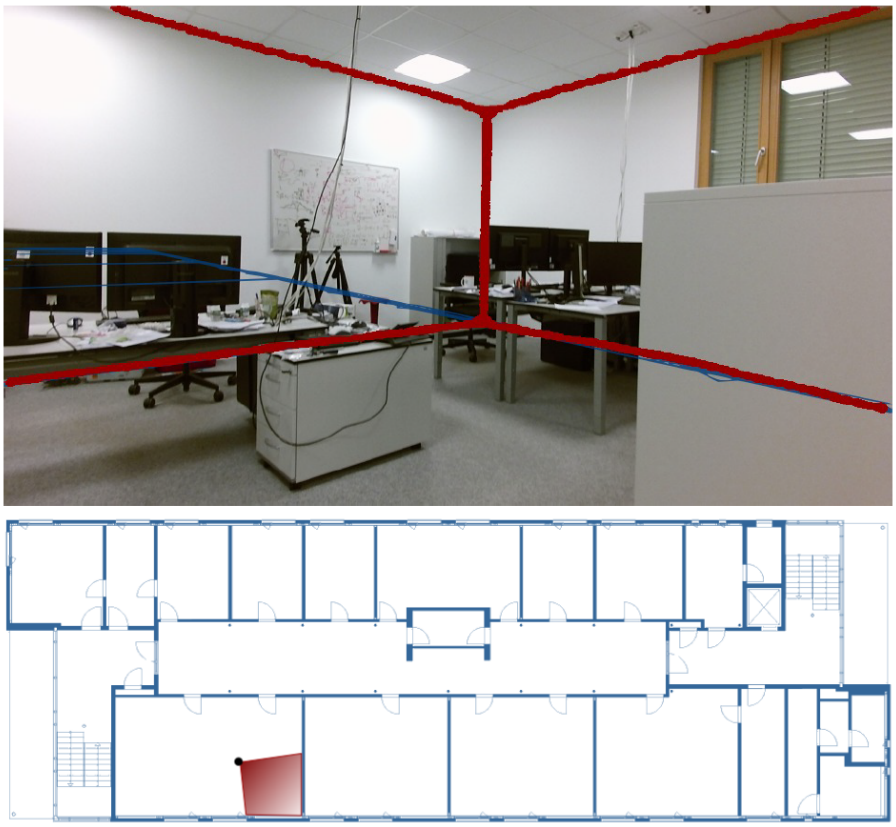


Figure 4.1 – Proposed approach by (BONIARDI et al., 2019b) that uses a network to extract the room layout edges from an image (top) and compares it to a layout generated from a floor plan (bottom) to localize the robot. Figure adapted from (BONIARDI et al., 2019b).

Previous works have proposed diverse approaches to interpreting sensor data and estimating the robot's position within its environment. Feature-based models extract distinct geometric landmarks or keypoints from sensor data, providing robust reference points for localization algorithms. For instance, (LEONARD; DURRANT-WHYTE, 1991b) worked on mobile robot localization by tracking geometric beacons from sonar data. Range-based models leverage distance measurements from sensors like laser range finders to estimate the robot's position relative to surrounding objects (THRUN; BURGARD; FOX, 2005). Correlation-based models compare sensor readings obtained at various robot poses to infer the robot's location. (OLSON, 2009) relies on the likelihood field model to propose efficient multi-level strategies for correlative scan-matching.

Single-valued models represent sensor data as singular values associated with different robot poses, offering a simplified yet effective means of estimating position probabilities. (ZHANG; ZAPATA; LéPINAY, 2012) propose the Similar Energy Region (SER), a single-valued observation model for robot localization. SER assigns a value to each position in free space, representing the sum of the ranges of all readings obtained by the

robot at that position. This model is advantageous for robots capable of 360-degree readings, as its measure of energy is independent of orientation. However, the simple sum of measured ranges can lead to misleading values, i.e., similar results for very different regions.

(MAFFEI et al., 2015) proposed the FSD, which is a similar concept to SER, but addresses the issue of misleading values by confining the measurements to a local circular region. Rather than obtaining an absolute measurement of the free space around the robot, the FSD calculates a kernel density estimate (KDE) to determine the ratio of free space relative to the maximum area within the local region. The FSD can be computed with different sensor modalities: (MAFFEI et al., 2015) used 2D lasers to estimate the free space from walls distance; (RIBACKI et al., 2018) used a camera upward-facing to estimate free space from ceiling boundaries; and (MAFFEI et al., 2020) used RGB-D cameras depth maps to compute FSD from point clouds projected to the 2D plan. Previous works that used depth sensors built a local map using the HIMM method (BORENSTEIN; KOREN, 1991b) to compute the FSD. In this work, we selected the FSD as the observation model and propose to compute it using monocular cameras with the HIMM method.

5 MONOCULAR INTERVAL EXTENDED FSD

This section presents the proposed method for robot localization in an indoor environment. The method can be divided into four parts: 1) estimation of a depth map from a monocular camera image; 2) construction of a local grid map based on the estimated depth map; and 3) computation of the FSD around the robot based on the local map; 4) estimation of the robot position based on a particle filter strategy using the FSD computed as observation model. The proposed method uses DPT-Hybrid model (RANFTL; BOCHKOVSKIY; KOLTUN, 2021) predictions to estimate the depth map. The model outputs disparity, which is converted to metric depth to build the local grid map. Then, the local map is used to estimate the FSD in the robot's surrounding area covered by the kernel. This method can be implemented in a real scenario to localize a robot in an indoor environment, considering the robot has enough processing power to run the depth estimation model in real-time.

In order to compute metric depth from the output of DPT-Hybrid model, we propose to use a simplified version of the least-squares criterion presented in Equation 3.1, where we set the shift t to zero. Thus, for a given dataset, only a scale s is computed for each image in disparity space. Then, we invert each scale s to depth space and compute their average to apply to all images of the dataset. Additionally, the standard deviation σ_{Scale} of the depth scale is computed and serves as an uncertainty measure in the depth estimation. The additional uncertainty presented in the depth estimation compared to using a RGB-D camera is taken into account in particle weighting.

Figure 5.1 presents a diagram of the pipeline used to compute the local map. Initially, the input image (a) is used to compute the depth map (b) with the model. Then, similarly to what (MAFFEI et al., 2020) proposed, we compute a point cloud (c) that is horizontally downsampled by a factor of 30 as presented in (d). Next, the 3D point cloud measurements are projected to 2D. As presented in (e), from the projected 2D points, we select as the distance to the wall the maximum range - to handle partial occlusions caused by furniture - for each orientation discretized in steps of 1º. Finally, a local grid is updated using the HIMM method (BORENSTEIN; KOREN, 1991b). The diagram also presents in (f) the measured distance to the wall for the average metric depth scale and one standard deviation below and above the average.

The correctness of the local grid map updated with HIMM is highly dependent on the accuracy of the estimated metric depth map. Considering the different scales to

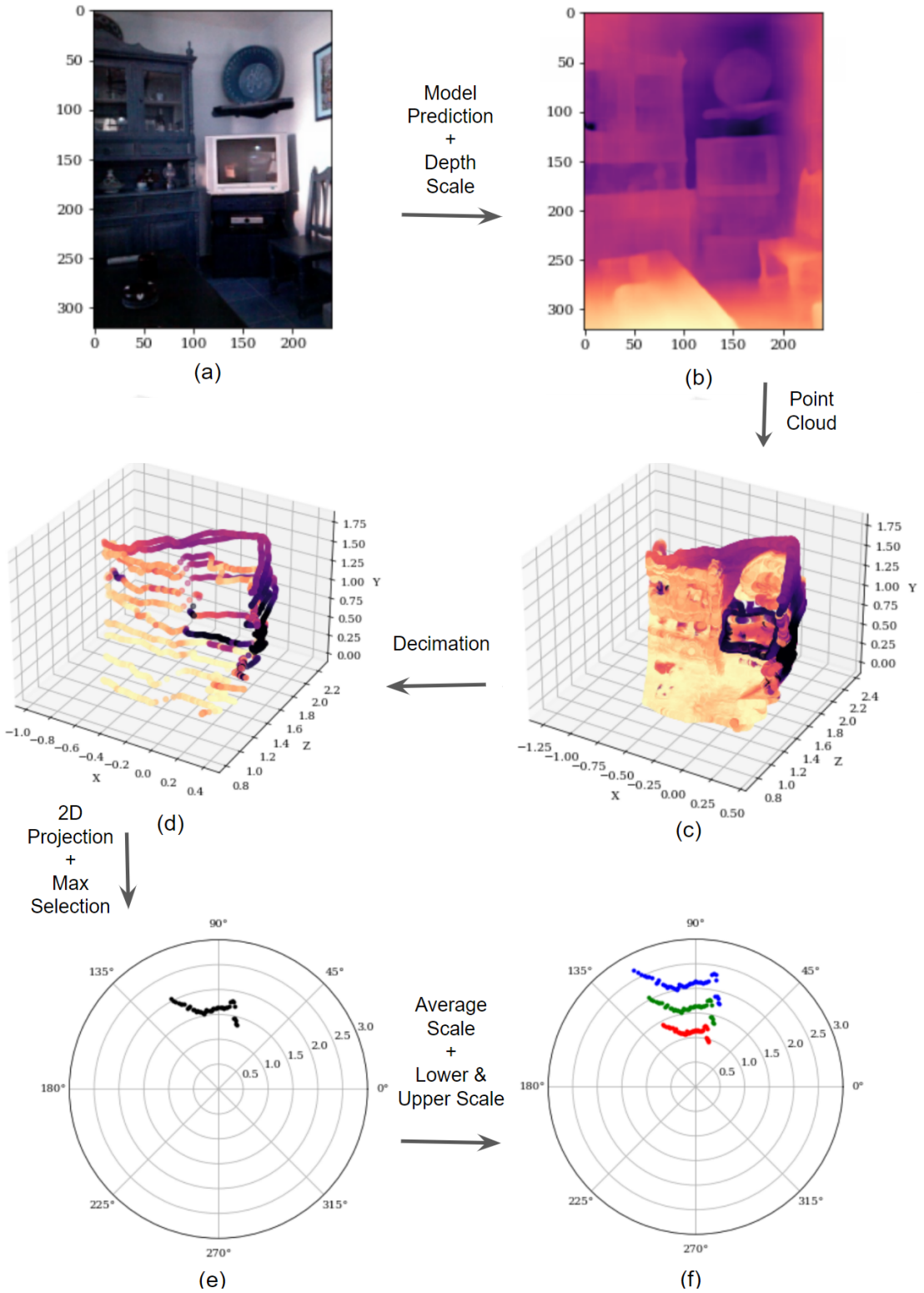


Figure 5.1 – Diagram of Monocular Interval Extended FSD. (a) Input image (b) Model prediction inverted and scaled to obtain metric depth (c) Point cloud obtained from metric depth (d) Decimated point cloud by a factor of 30 (e) 2D projection of decimated point cloud with selection of maximum in each orientation (f) 2D projection using average scale and one standard deviation below and above average scale.

compute metric depth, we can estimate a different local map, which impacts in the FSD computation. Figure 5.2 illustrates the differences in the FSD for different depth scales, where in (b) we used one standard deviation below the average depth scale, in (c) we used the average depth scale, and in (d) we used one standard deviation above the average depth scale. For all different scales, the FSD is computed using a uniform circular kernel with the same radius of $1.5m$. We can see that with a smaller scale, the robot appears to be closer to obstacles, such as the corner of the room shown in the example, than with a larger scale.

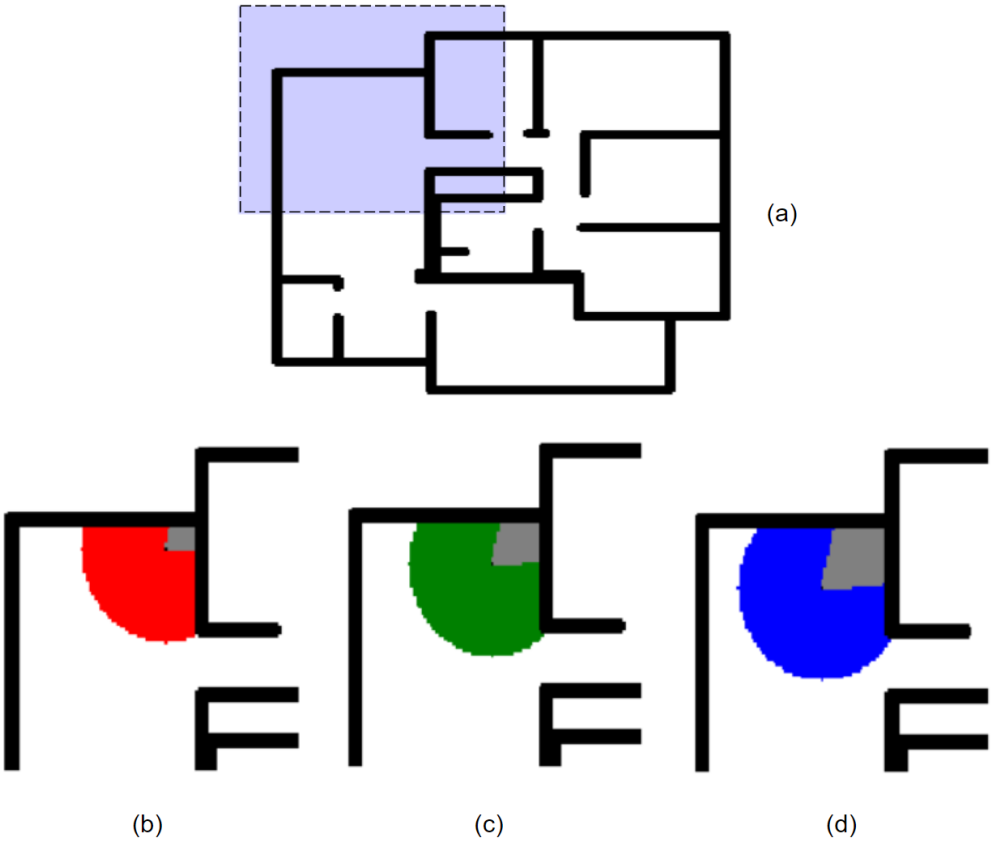


Figure 5.2 – FSD for different monocular depth estimation scales. (a) Floor plan for *pare-s1* map (b) Local map for depth scale one standard deviation below average scale (c) Local map for depth scale equal to average scale (d) Local map for depth scale one standard deviation above average scale.

Since the depth scale influences the FSD computation, the particle weighting process needs to account for its uncertainty. The idea is to increase the measured FSD interval proportionally to the depth scale variation for different images. In the proposed method, we define the Interval Extended FSD $[\Psi(m_r)^*]$ modifying the infimum and supremum interval presented in Interval FSD proportionally to the standard deviation of the depth scale. Given the definition of Interval FSD in Equation 2.8, the uncertainty can be added to the infimum interval and to the supremum interval as presented in Equation 5.1 and

Equation 5.2, respectively.

$$\underline{\Psi(m_0)^*} = (1 - \alpha\sigma_{Scale})^2 \times \Psi(m_0) \tag{5.1}$$

$$\overline{\Psi(m_0)^*} = max(\overline{\Psi(m_0)}, (1 + \alpha\sigma_{Scale})^2 \times \Psi(m_0)) \tag{5.2}$$

where $\Psi(m_0)$ is the measured FSD using the average depth scale, $\overline{\Psi(m_0)}$ is the previous definition of the supremum interval given in (MAFFEI et al., 2020) that only considers the added uncertainty of unknown cells, σ_{Scale} is the standard deviation of the depth scale and α is the percentage of this standard deviation to be used in the interval. The added uncertainty proportional to σ_{Scale} is squared because σ_{Scale} is the uncertainty of the free space radius, which translates into a squared uncertainty for the free space area.

Finally, the particle weight w of a given particle p is kept at 1 if the particle FSD value is inside the interval. For FSD values outside the interval, the weight w is computed as the distance from the infimum interval or the supremum interval depending if the particle FSD is below or above the robot measured FSD. The particle weight w of a given particle p for Interval Extended FSD is defined by Equation 5.3.

$$\omega(p) = \begin{cases} 1, & \text{if } \Psi(m_p) \in [\Psi(m_r)^*] \\ f_\Psi(\Psi(m_p), \overline{\Psi(m_r)^*}), & \text{if } \Psi(m_p) > \overline{\Psi(m_r)^*} \\ f_\Psi(\Psi(m_p), \underline{\Psi(m_r)^*}), & \text{if } \Psi(m_p) < \underline{\Psi(m_r)^*} \end{cases} \tag{5.3}$$

where we took the definition of the particle weight for Interval FSD presented in Equation 2.11 and replaced the infimum and supremum interval for their new definition in Equation 5.1 and Equation 5.2. Thus, the extended interval tends to keep particles with depth scales close to the average scale, increasing the likelihood of the particle filter to converge to the right solution.

We employ a particle filter to address the global localization problem. The use of FSD as the observation model performs dimensionality reduction of the information. It is essential to acknowledge that this reduction diminishes precision and effectiveness. Using a small number of particles for robot localization will likely result in failure. Fortunately, since the FSD is a scalar value and can be stored for all grid map positions, increasing the number of particles significantly is feasible. It results in minimal impact on processing

time, providing a solution to this challenge. Thus, a number of 20000 particles was chosen for the proposed method.

Figure 5.3 presents four stages of the particle filter during a localization execution. We can see the behavior of the particles caused by the use of FSD as the observation model. Initially, all particles are uniformly distributed over the map. Once the robot starts moving, the particles accumulate in similar positions close to corners for all rooms due to the ambiguity of FSD. Following additional movement, only two ambiguous swarms of particles are left. Finally, all ambiguities are solved and the filter converges to the right position.

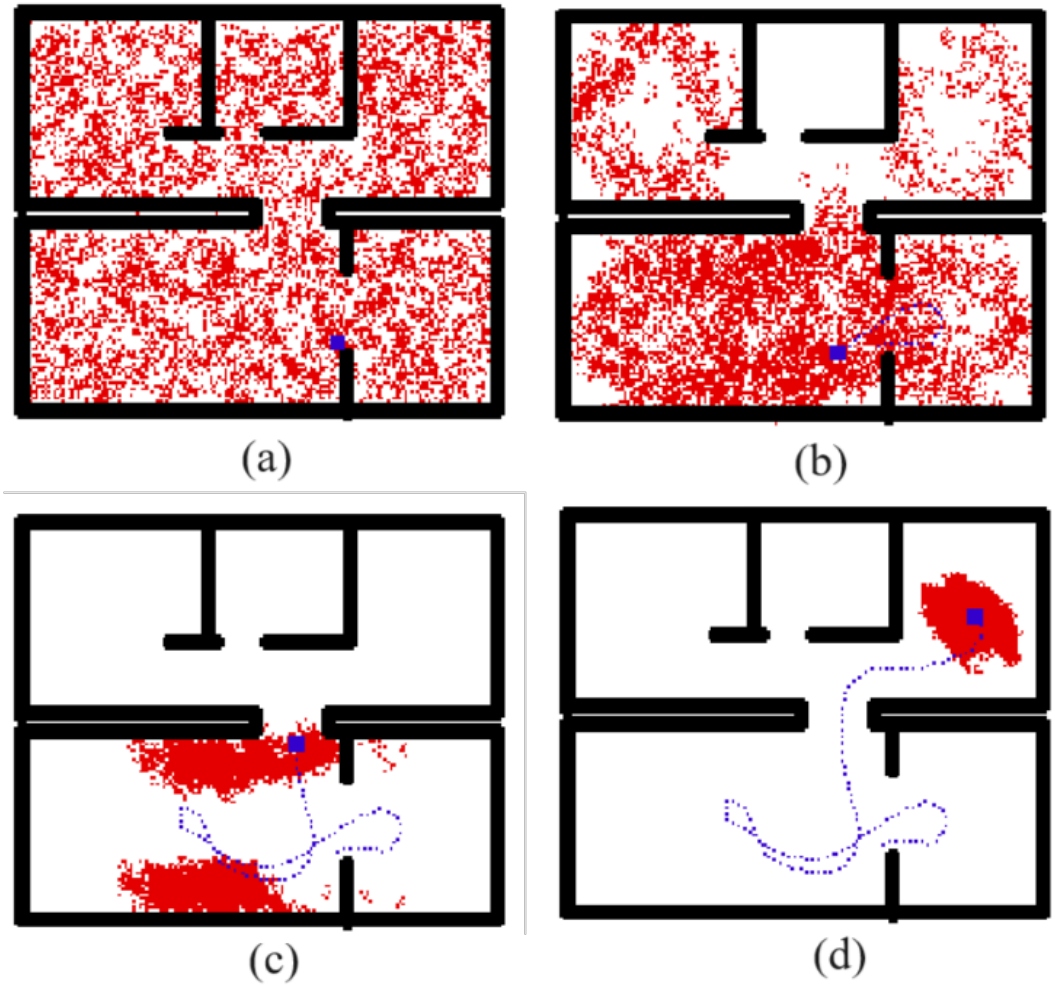


Figure 5.3 – Particle Filter stages for *alma-s2* map. Particles are presented in red, the robot's position is presented in a blue square and its path is presented in a blue dotted line. (a) Uniformly distributed particles; (b) Particles close to corners due to the ambiguity of the observation model; (c) Only two ambiguous swarms of particles left; (d) Particles converge to the robot position.

The use of Extended Interval FSD introduces increased uncertainty in particle weighting, consequently retaining a greater number of particles in ambiguous observation model locations. While this positively influences the success of filter convergence, it

is expected to extend the time required for the filter to converge. Moreover, following convergence, we anticipate observing a higher dispersion of particles. Once the global localization problem is solved, we suggest the use of a method for local localization for a more precise robot position estimation.

6 EXPERIMENTS

We conducted a set of experiments to evaluate the performance of FSD localization based on monocular depth estimation. The experiments were performed using *Robot@home dataset* (RUIZ-SARMIENTO; GALINDO; GONZáLEZ-JIMéNEZ, 2017), which consists of sequences of observations collected with a mobile robot using four RGB-D cameras and 2D laser scanners. The 2D laser was placed at the front part of the robot base, at a height of 0.31 m. The cameras were mounted at a height of 0.92 m with an angular orientation of -45º, 0º, 45º, 90º. Figure 6.1 presents the mobile robot used to collect data for the experiments.

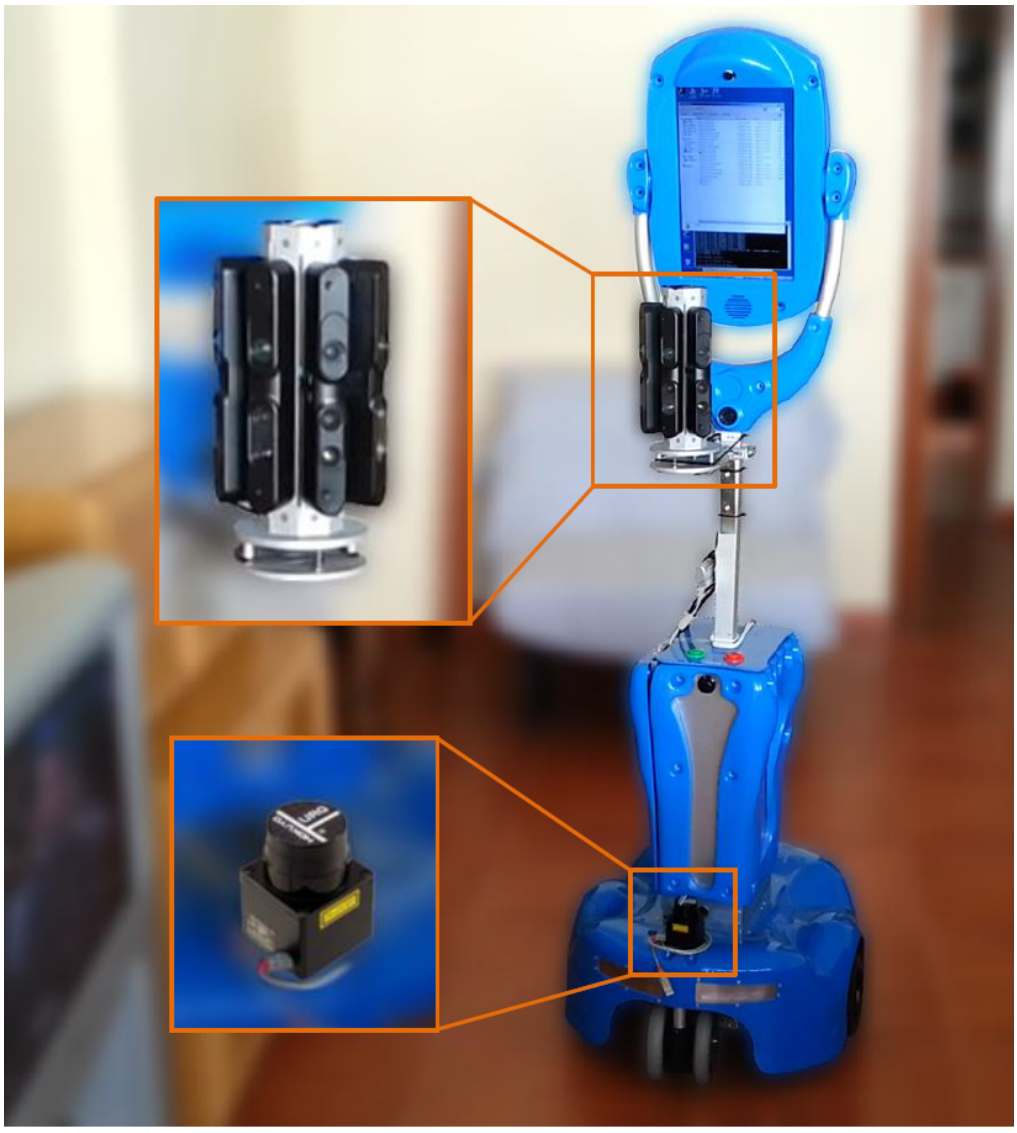


Figure 6.1 – Robotic platform employed to collect the dataset along with details of the sensors mounted on it. Figure adapted from (RUIZ-SARMIENTO; GALINDO; GONZáLEZ-JIMéNEZ, 2017).

For the depth estimation model evaluation and localization experiments performed, we only used the forward-facing camera. We selected four sequences of observations of the datasets recorded in 4 different apartments to execute the experiments. For these maps, a kernel size of $1.5m$ was chosen for the best performance given their average room size. Figure 6.2 presents the maps and trajectories selected from the dataset.

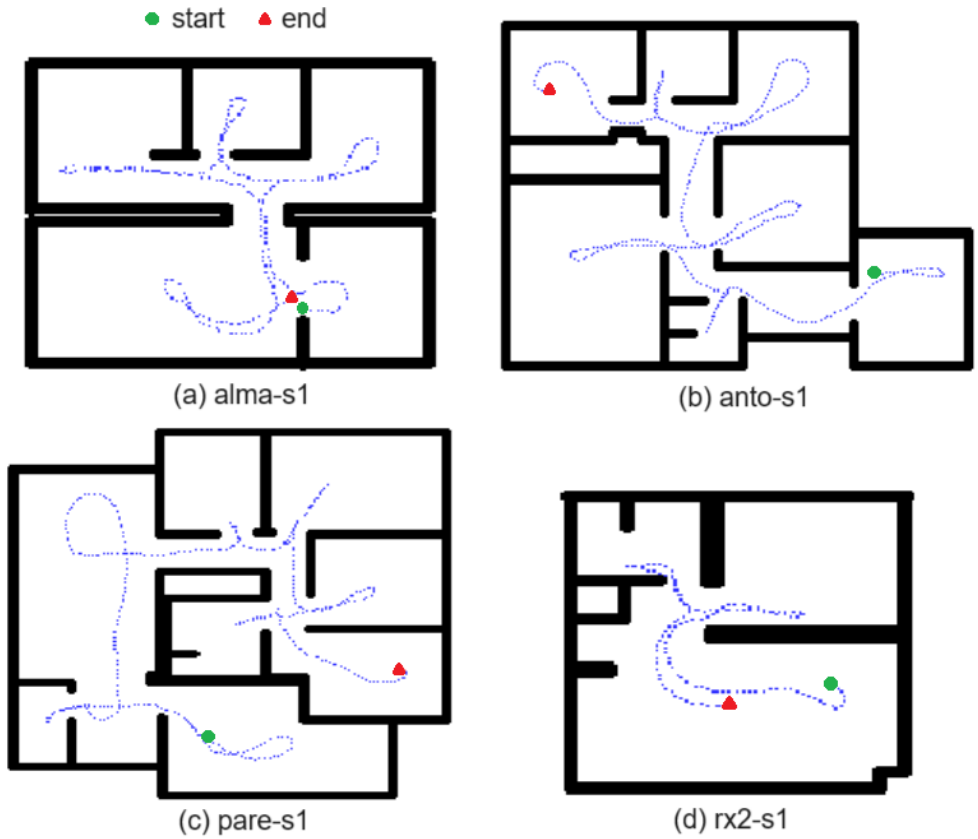


Figure 6.2 – Maps and trajectories of the 4 tested scenarios from the *Robot@home dataset*. (a) alma-s1: area $8.2 \times 6.6m^2$, path length $39.9m$ (b) anto-s1: area $8.7 \times 12.4m^2$, path length $43.7m$ (c) pare-s1: area $10.2 \times 10.3m^2$, path length $43.2m$ (d) rx2-s1: area $5.7 \times 6.1m^2$, path length $15.7m$

In Section 6.1, we present the first part of the experiments consisting in the evaluation of the monocular depth estimation model. The goal of the experiments was to evaluate how accurate is the depth estimation compared to the RGB-D depth channel. We used the model to estimate depth in the sequences of RGB images and calculated the error against RGB-D images. Additionally, since the observation model is derived from the 2D projected ranges, we also computed the error between the range measurements obtained from the estimated depth and the ones obtained from the RGB-D camera.

The observation model using monocular depth estimation is expected to have more uncertainty when compared to the one using the RGB-D camera. In order to verify how

robust is the proposed method *Interval Extended FSD* to noisy measurements, we evaluate the influence of α for different noisy scenarios in FSD estimation. The experiments presented in Section 6.2 were performed comparing the performance of the global localization when using ground truth FSD disturbed with different noise levels.

Finally, in Section 6.3, we evaluated the performance of the global localization particle filter using the estimated depth combined with *Interval Extended FSD* compared to using the RGB-D camera with *Interval FSD*. The idea is to compare our system to another one that has demonstrated superior performance compared to recently developed systems and uses a depth sensor as a baseline. The global localization experiments were also performed using the true value of the FSD as the measurement model. Additionally, experiments only using the movement model were performed. These experiments aimed to evaluate the particle weighting methods against the theoretical best (perfect observation model) and worst (no observation model) performance. A detailed explanation of the experiments is presented in the following sections.

6.1 Monocular depth estimation

This section presents the experiments performed to evaluate the accuracy of the monocular depth estimation model. The model used for the depth estimation task was DPT-Hybrid (RANFTL; BOCHKOVSKIY; KOLTUN, 2021) trained on MIX 6 dataset with about 1.4 million images and fine-tuned on NYUv2 dataset (SILBERMAN et al., 2012). We selected this model because it was trained in a *zero-shot cross-dataset transfer* approach, that is, the model is trained on certain datasets and its performance is tested in datasets that were never seen during training (RANFTL et al., 2020). Also, the model was fine-tuned in a dataset with domestic indoor scenarios. Thus, we expected the model to generalize well and perform well in *Robot@Home* dataset. Examples of the model depth predictions compared to RGB-D camera images and depth channel for all *Robot@Home* dataset maps are presented in Figure 6.3.

6.1.1 Model depth scale estimation

The DPT-Hybrid network was trained with an affine-invariant loss, resulting in predictions that are arbitrarily scaled and shifted. The network outputs disparity, a repre-

sentation inversely proportional to depth (RANFTL; BOCHKOVSKIY; KOLTUN, 2021). To compute absolute depth from these predictions, the disparity predictions have to be aligned with the inverse depth ground truth of the *Robot@Home* dataset and then inverted

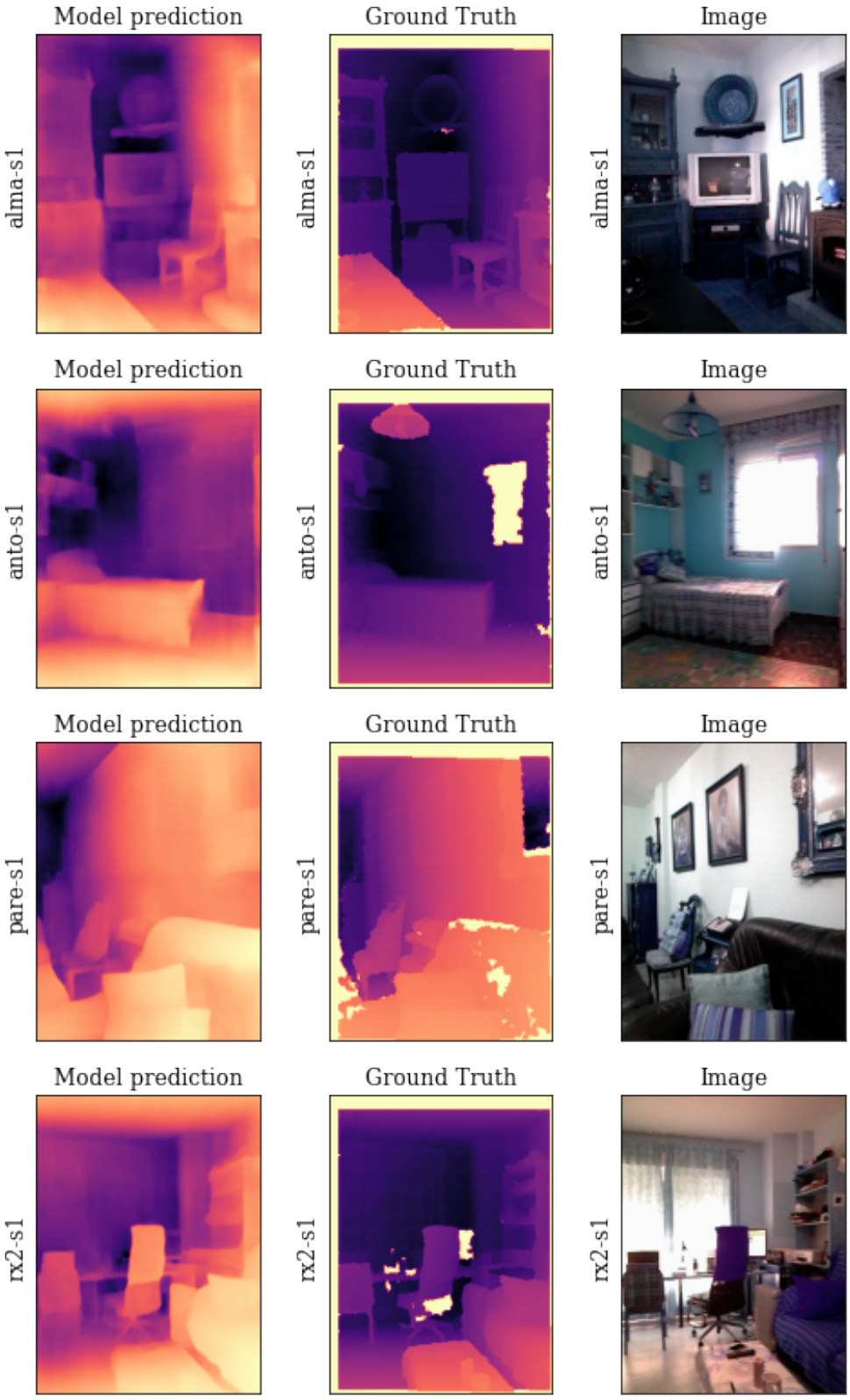


Figure 6.3 – Model predictions compared to ground truth for *Robot@Home* dataset maps

to depth space. This alignment was performed by computing a disparity scale for each *Robot@Home* dataset image prediction that minimizes a least square criterion in inverse depth space described in Equation 3.1 with t set to zero.

In a real scenario, we need a single depth scale to be applied to all model predictions. To compute this single depth scale, we perform the following steps:

- **Alignment to Ground Truth:** Each disparity prediction is aligned to the inverse depth ground truth of the *Robot@Home* dataset by calculating a disparity scale that minimizes the least squares error in inverse depth space.
- **Conversion to Depth Space:** Once aligned, these disparity scales are converted back to depth space.
- **Averaging Across the Dataset:** The depth scales from all dataset examples are averaged to compute a single depth scale.

Figure 6.4 presents a histogram of the computed depth scales for all dataset examples. The histogram shows a wide range of depth scales, varying from half to twice the average scale depending on the scene. This wide range indicates that the average depth scale is subject to uncertainty. We quantify this uncertainty by calculating the standard deviation of the depth scales obtained from the *Robot@Home* dataset examples. The average depth scale is 2161, and its standard deviation is 29.7% relative to this average.

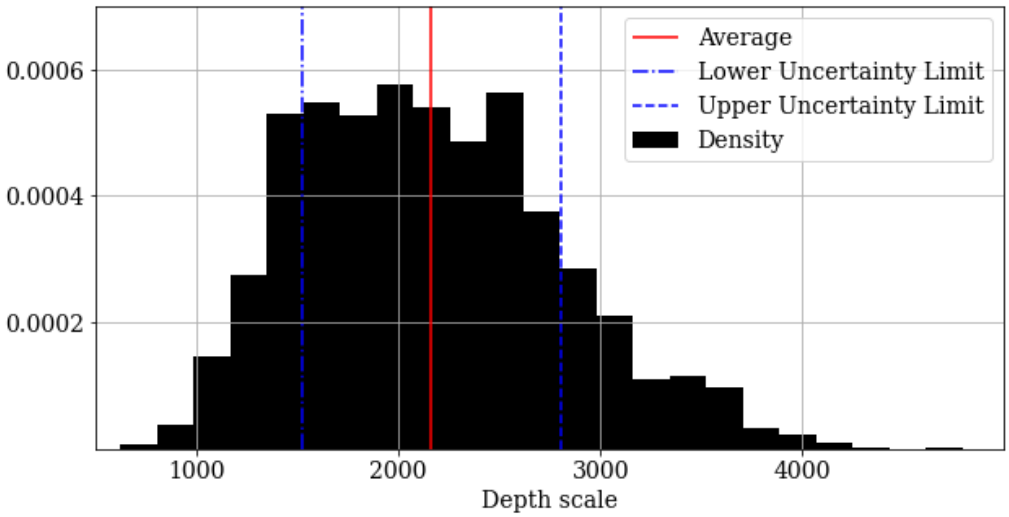


Figure 6.4 – Depth scale histogram over all *Robot@Home* dataset examples

6.1.2 Model performance on images

We evaluated the network performance in the *Robot@Home* dataset four maps with the metrics defined in (RANFTL et al., 2020). The root mean squared error ($RMSE$) and mean absolute value of the relative error ($AbsRel$) were computed between the predicted depth pixels z_i and ground truth depth pixels z_i^*. Lower values for the pixel error metrics $RMSE$ and $AbsRel$ indicate better performance. The percentage of pixels with $\delta = max(\frac{z_i}{z_i^*}, \frac{z_i^*}{z_i}) < 1.25$ was also computed. For the pixel percentage δ metrics, higher values indicate better performance. In order to compute the metrics, we inverted the dataset predictions to depth space and applied the average depth scale to all of them. The RGB-D camera depth channel was used as ground truth. Results are presented in Table 6.1.

Table 6.1 – Metrics computed for images

Map	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	$AbsRel$	$RMSE$
alma	0.390	0.718	0.909	0.371	0.611
anto	0.450	0.773	0.930	0.320	0.586
pare	0.391	0.727	0.907	0.344	0.647
rx2	0.416	0.717	0.899	0.384	0.594

The model has a similar performance for all *Robot@Home* dataset maps. Evaluating the δ metrics, we can notice that less than 50% of the pixels have depth estimation error relative to the ground truth lower than 25% ($\delta < 1.25$). Considering relative differences lower than 95% ($\delta < 1.25^3$), more than 90% of the pixels meet this percentage error. When we evaluate *AbsRel* metric, it is shown that the absolute difference between depth estimation and ground truth is between 30% to 40% the ground truth value. This performance is worse than the performance obtained in NYUv2 dataset for this model (RANFTL; BOCHKOVSKIY; KOLTUN, 2021), which is expected since it was fine-tuned for it. The monocular depth estimation *AbsRel* error is expected to be proportional to the expected FSD computation error, since the latter is computed from the local map built with the former.

6.1.3 Model performance on range measurements

In order to obtain a better estimation of the expected error percentage in the FSD estimation, we evaluated the monocular depth estimation performance in the computation of range measurements, presented in Table 6.2. Range measurements are obtained from the projections of 3D depth measurements in a 2D plane. Once they are projected, we take the maximum projected range for a given orientation (discretized in steps of 1º) as the range measurement. The idea here is to filter out dynamic objects and only keep those that are static, like walls (MAFFEI et al., 2020).

Table 6.2 – Metrics computed for range mesurements

Map	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	$AbsRel$	$RMSE$
alma	0.302	0.523	0.612	0.179	0.544
anto	0.350	0.580	0.676	0.201	0.619
pare	0.277	0.526	0.660	0.279	0.775
rx2	0.379	0.579	0.667	0.224	0.554

Results obtained with image decimation (gap=30).

The performance for range measurements is even worse than the performance for images when we evaluate the δ metrics. Less than 40% of the ranges have a difference relative to ground truth lower than 25% ($\delta < 1.25$). Also, less than 70% have this relative error lower than 95% ($\delta < 1.25^3$). On the other hand, the performance when evaluating *AbsRel* is better when comparing range measurements to image pixels. The ranges absolute relative error is bounded from 10% to 30%. Since the range measurements are computed as the maximum depth values for a given orientation and *AbsRel* is computed relative to the ground truth depth, the *AbsRel* is expected to be lower for ranges than for overall pixels.

The experiments presented the expected error in FSD estimation due to the monocular depth estimation errors. Given the results obtained, we expect to have 10% to 30% error in the construction of the local map with HIMM. The free space cells surrounding the robot will be affected by how close to the robot are the walls in the local map, which translates into a wrong estimation of the free space radius, and therefore, a wrong estimation in the FSD around the robot.

6.2 FSD localization robustness to noise

The results of the depth estimation model obtained in Section 6.1 indicate the estimated FSD using a monocular camera presents an error proportional to the square of the estimated depth error - expected to be between 10% to 30%. We performed preliminary experiments aiming to evaluate how FSD localization behaves when the estimated FSD value is disturbed with noise. For these experiments, we took the *Ground Truth FSD* value extracted from the floor plan and applied the disturbance. The noise disturbance consisted of multiplying the *Ground Truth FSD* by a squared Gaussian noise with mean equals to 1 and standard deviation equals to 10%, 20% and 30%.

The idea was to evaluate the particle weighting using *Interval Extended FSD* with different values of α. The experiments were performed in the four trajectories from *Robot@Home* dataset maps. The ground truth of the robot pose and odometry were obtained using SLAM and scan matching techniques, since they were not directly available (MAFFEI et al., 2020). We ran the experiments 30 times using 20000 particles for $\alpha = [0, \frac{1}{3}, \frac{1}{2}, 1]$.

We obtained the particle error and heading difference for each particle compared to the ground truth and computed their weighted mean over time for each experiment. We consider an experiment converged if the mean particle error becomes smaller than *1m* and the angle difference becomes smaller than *20º* and does not exceed *1.5m* and *30º*, respectively, until the end of the trajectory. For experiments that converged, we computed the metrics proposed in (MAFFEI et al., 2020), which consist of **succeed distance** - total distance traveled by the robot until convergence; and the average **mean particle error** after convergence.

Figure 6.5 presents the convergence percentage among all experiments. The convergence for maps *anto-s1* and *pare-s1* is close to 100% for all levels of noise and values of α. For *alma-s1* with 30% noise level, almost half of the experiments did not converge for all α values. In this scenario, the convergence is slightly better for α equals to 1/3. Similarly, for *rx2-s1*, when noise levels are 20% and 30%, there are some experiments that do not converge. For these experiments, when α is increased, the convergence percentage increases, where the best scenario is 1/2 for 20% noise level and 1/3 for 30% noise level.

We observe failures in convergence when noise level is added for maps that have shorter trajectories and simpler topologies. Specifically, *alma-s1* and *rx2-s1* have shorter trajectories and fewer corridors and rooms compared to *anto-s1* and *pare-s1*. More fail-

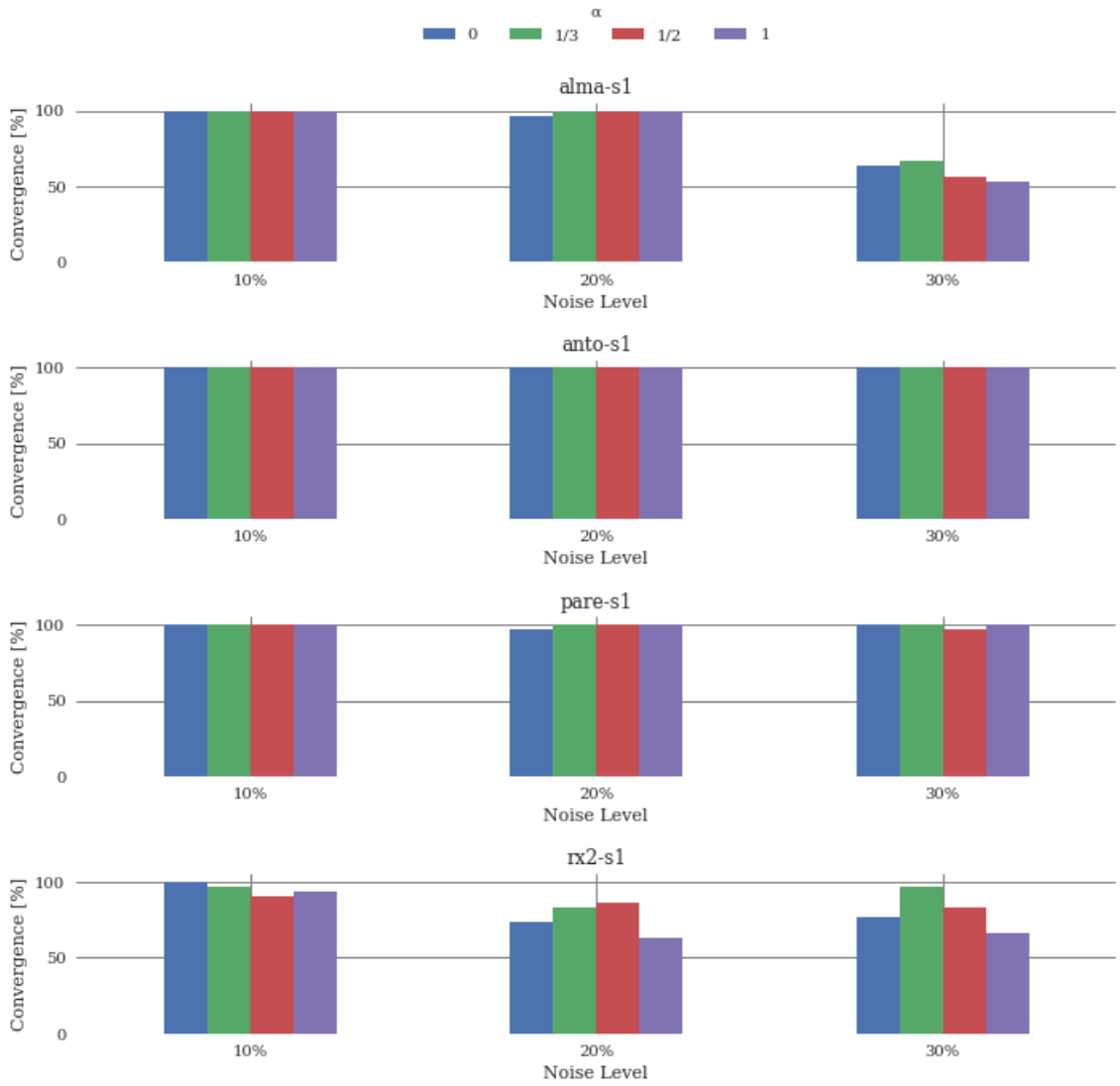


Figure 6.5 – Convergence for *Ground Truth FSD* disturbed with noise for different noise levels and α values.

ures in convergence are expected for these maps because the added uncertainty in the measurements tends to increase the number of observations needed until the filter converges. When the uncertainty is also added to the measurement model ($\alpha > 0$), we see slight improvements in convergence.

The average metrics computed among all experiments that converged are presented in Figure 6.6. Considering *succeed distance*, the greater the noise level, the longer the experiments take to converge for the maps *alma-s1*, *anto-s1* and *pare-s1*. The only exception is for the map *rx-2*, which has the shortest trajectory and the increase in *succeed distance* with noise level is not observed. Additionally, the increase of α also translates into an increase in *succeed distance*. A similar result is observed for *mean particle error* after convergence, where it is greater for greater noise levels and greater α.

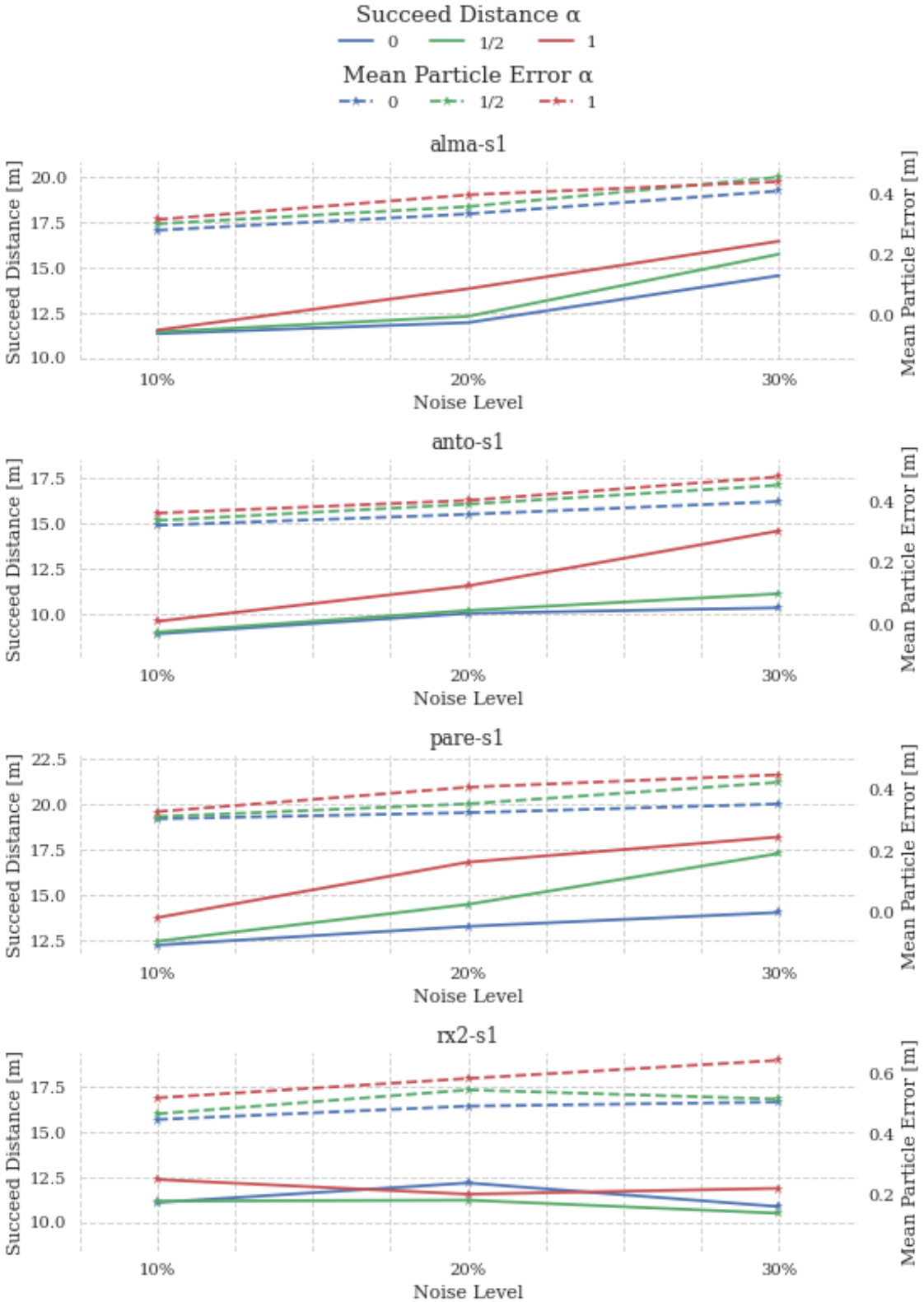


Figure 6.6 – Succeed Distance and Mean Particle Error after convergence for *Ground Truth FSD* disturbed with noise for different noise levels and α values.

In summary, Figures 6.5 and 6.6 show that using α greater than 0 in the proposed method *Interval Extended FSD* increases the convergence for some experiments. This

occurs when the observation model contains a significant level of noise, so the increase in the interval, i.e., increase of α, in the particle weighting accounts for the greater uncertainty. On the other hand, since it adds more uncertainty to the observation model of the particle filter, it also increases the *succeed distance* and *mean particle error* after convergence.

6.3 FSD Localization

We evaluated different particle weighting strategies for the robot localization problem in the four trajectories from *Robot@Home* dataset maps. We performed the test using only the forward-facing camera to simulate a scenario where only one monocular camera is available. For each method, we ran the experiments 30 times using 20000 particles. For all experiments, we computed the metrics described in Section 6.2 as well as the **Final Error** for those that converged. Additionally, we computed the average of the mean particle error over time among all experiments.

6.3.1 Monocular Interval Extended FSD

Initially, the monocular camera was used to compute FSD as the measurement model for particle weighting. The performance was evaluated using *Monocular Interval Extended FSD* strategy. We performed experiments using the standard deviation of the depth scale σ_{Scale} obtained in Section 6.1 as our uncertainty factor multiplied by α. For larger values of α, we have a wider interval, which translates into more uncertainty in the measurement model. In order to evaluate the influence of the interval width on performance, the experiments were performed for $\alpha = [0, \frac{1}{3}, \frac{1}{2}, 1]$.

Table 6.3 presents the convergence percentage (*Conv*) of experiments and the average and standard deviation of *succeed distance* (*SucD*), *mean particle error* after convergence (*MeanParErrAC*) and *final error* (*FErr*) computed for those that converged. For a successful localization, the mean particle error and heading difference are expected to decrease below a given level. Again, we consider convergence when the mean particle error and heading difference become smaller than *1m* and *20º* and keep below *1.5m* and *30º*, respectively, until the end of the trajectory.

The results presented in Table 6.3 show that all *Monocular Interval Extended FSD*

Table 6.3 – Localization metrics computed for *Monocular Interval Extended FSD* for different values of α.

Dataset	Method	SucD [m]		MeanParErrAC [m]		FErr [m]		Conv [%]
		μ	σ	μ	σ	μ	σ	%
alma-s1	Mon Int Ext FSD (α=0)	**11.43**	0.10	**0.36**	0.03	0.46	0.04	100.0
	Mon Int Ext FSD (α=1/3)	12.45	1.38	0.39	0.05	0.49	0.08	100.0
	Mon Int Ext FSD (α=1/2)	12.76	0.98	0.39	0.03	0.44	0.04	100.0
	Mon Int Ext FSD (α=1)	13.38	0.22	**0.36**	0.02	**0.38**	0.02	100.0
anto-s1	Mon Int Ext FSD (α=0)	**9.77**	0.85	0.45	0.03	**0.41**	0.12	100.0
	Mon Int Ext FSD (α=1/3)	10.24	1.15	0.49	0.05	0.48	0.16	100.0
	Mon Int Ext FSD (α=1/2)	10.35	1.08	0.47	0.03	0.48	0.12	100.0
	Mon Int Ext FSD (α=1)	14.59	0.26	**0.39**	0.01	**0.41**	0.01	100.0
pare-s1	Mon Int Ext FSD (α=0)	23.14	9.13	0.55	0.10	0.69	0.13	23.0
	Mon Int Ext FSD (α=1/3)	18.48	2.17	0.53	0.05	0.71	0.05	87.0
	Mon Int Ext FSD (α=1/2)	18.26	0.59	0.52	0.06	0.69	0.06	**100.0**
	Mon Int Ext FSD (α=1)	**17.49**	0.84	**0.40**	0.02	**0.52**	0.02	**100.0**
rx2-s1	Mon Int Ext FSD (α=0)	**10.68**	1.25	**0.55**	0.10	0.82	0.06	**83.0**
	Mon Int Ext FSD (α=1/3)	10.78	1.13	0.58	0.11	0.80	0.07	73.0
	Mon Int Ext FSD (α=1/2)	11.17	1.41	0.59	0.12	**0.73**	0.11	43.0
	Mon Int Ext FSD (α=1)	-	-	-	-	-	-	0.0

experiments converge for all values of α in *alma-s1* and *anto-s1* maps. Considering the map *pare-s1*, we only observe 100% convergence for $\alpha = 1/2$ and $\alpha = 1$. On the other hand, we do not observe 100% convergence for *rx2-s1* map for any value of α. In this map, for $\alpha = 0$ and $\alpha = 1/3$ the majority of experiments converged, achieving 83% and 73% convergence, respectively; for $\alpha = 1/2$ only 43% of the experiments converged; and for $\alpha = 1$ no experiment converged.

Figure 6.7 presents the average among all experiments of the weighted mean particle error over time for *Monocular Interval Extended FSD* with different values of α. We can observe the error decreases for experiments that have a high rate of convergence. In general, for lower values of α, the error decreases faster, which means a faster convergence. The faster convergence for lower values of α can also be observed in *succeed distance* in Table 6.3. The only exception is for the map *pare-s1*, which presents higher *succeed distance* for small α values, but also presents low rates of convergence.

Considering the end of the experiment in Figure 6.7 for the map *rx2-s1*, we observe the mean particle error has an increase for smaller α values and has a drop for $\alpha = 1$. Although the error drops below *1m*, we still do not observe convergence for $\alpha = 1$

because the particles' heading has a high variance. Figure 6.8 shows the end of experiments with $\alpha = 0$ and $\alpha = 1$ for the map *rx2-s1*, the red dots represent the particles' positions and the blue arrows represent their heading. The image also shows the room where the experiment ends, which is an open area that contains ambiguous FSD values

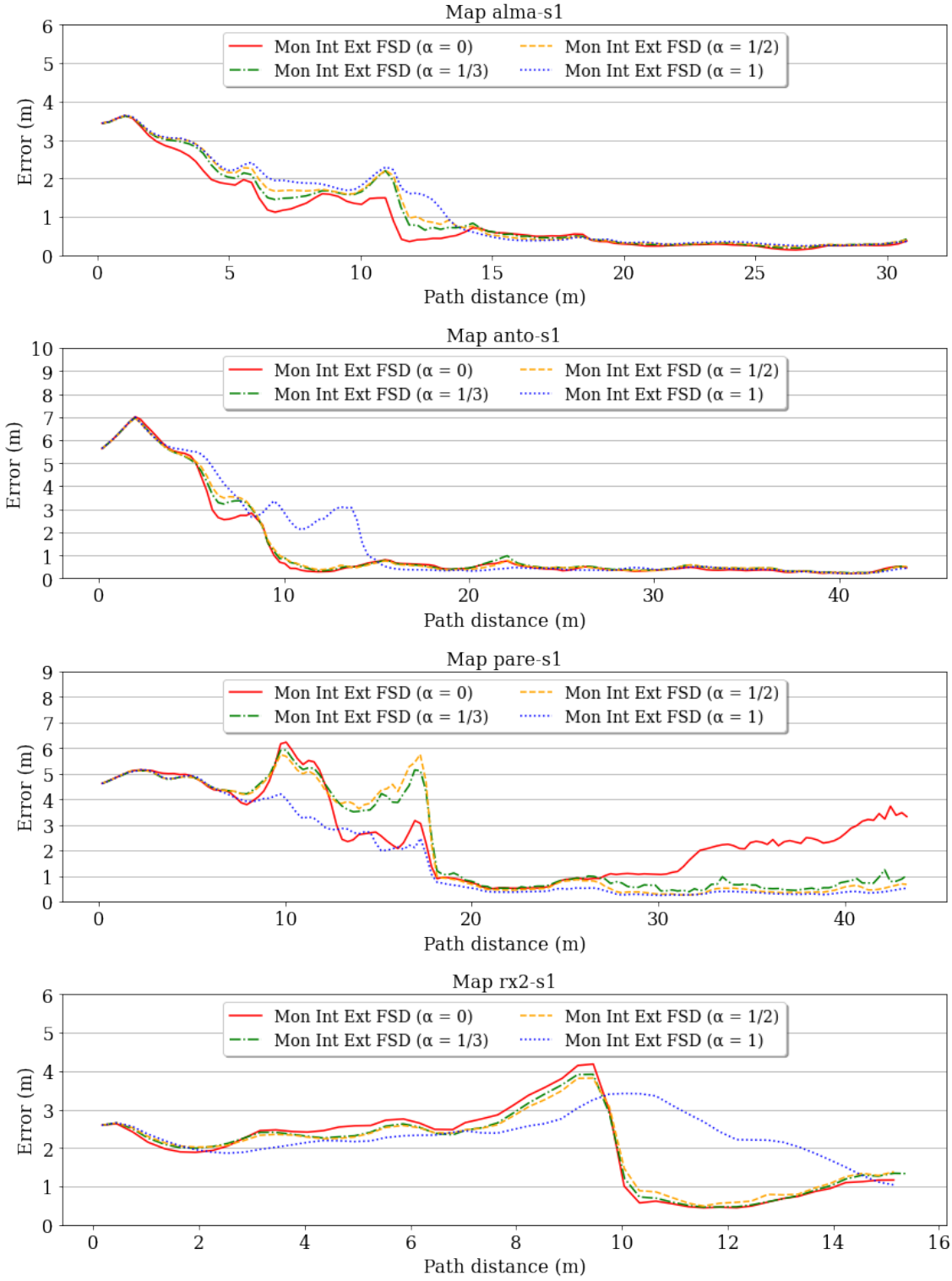


Figure 6.7 – Average among all experiments of the weighted mean particle error over time for Monocular Interval Extended FSD for different values of α.

in its center. This also explains why some experiments present an increase in their error after convergence. Finally, the convergence criteria using both particle position error and particle heading difference, combined with a relaxed criteria after the initial convergence criteria is met, work great in identifying convergence for a variety of situations.

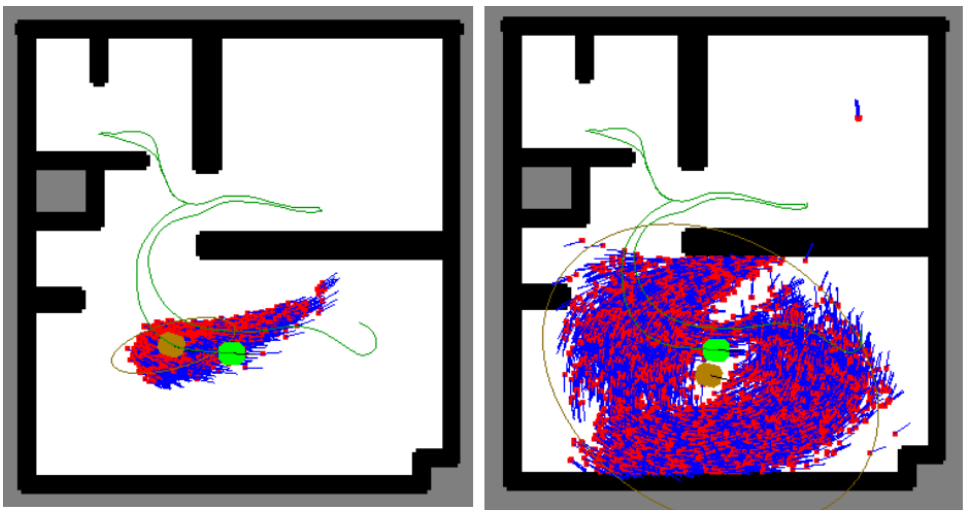


Figure 6.8 – Final particles distribution for the map *rx2-s1*, where the green circle and path are the ground truth position and trajectory, respectively; the dark yellow circle is the estimated position; and the red dots and blue arrows are the particles' position and heading, respectively. (a) Converged experiment ($\alpha = 0$); (b) Not converged experiment ($\alpha = 1$).

As presented in Table 6.3, the influence of α in convergence varies depending on the map. Specifically, for *pare-s1*, the **higher** the α the higher the convergence. Conversely, for *rx2-s1*, the **lower** the alpha, the higher the convergence. The complexity of the map and the trajectory size is considerably different for *pare-s1* and *rx2-s1*. While *pare-s1* is a complex map with a longer trajectory ($> 40m$), *rx2-s1* is a simple map with a shorter trajectory ($< 20m$). The α parameter increases the uncertainty of the measurement model, which translates into an increase in the diversity of the particles. For complex maps, a greater diversity might be helpful for them to converge to the right position when the trajectory is long enough. However, for simple maps with many symmetries in the measurement model, a higher diversity might prevent the particles from converging in shorter trajectories.

Complementing the analysis, Figure 6.9 presents the average of the position of the particle standard deviation over time among all experiments for *pare-s1* and *rx2-s1* for $\alpha = 0$ and $\alpha = 1$. In map *pare-s1* for both values of α, the particles converge to a given position, which is usually the correct one for higher α but the wrong one for lower α. On the other hand, in *rx2-s1*, when α is lower, the experiments usually converge, while for higher α the diversity of the particles keep high until the end of the experiment.

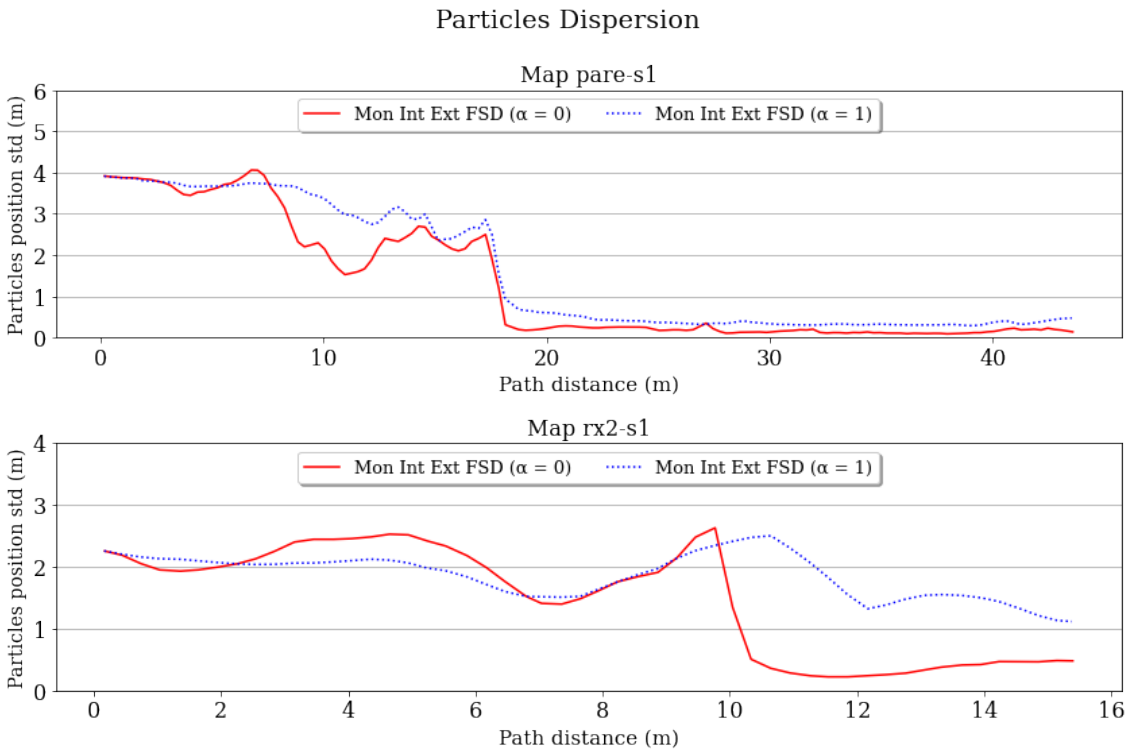


Figure 6.9 – Average particles dispersion over time for *pare-s1* and *rx2-s1*.

Videos of one experiment execution sample for each map were recorded to present the method performance and are available online [1]. The videos present the floor plan with the MCL particles' position in red and the heading in blue. The robot position is presented in green and the estimated robot position is presented in brown. We also present images seen by the robot and the point cloud obtained from the monocular depth estimation prediction.

6.3.2 Comparison to other methods

Following the evaluation, we compared the proposed method with other particle weighting strategies. We selected *Monocular Interval Extended FSD* ($\alpha = 1/3$) as the method for comparison since it yielded the highest average convergence among all maps. It was compared with four strategies: *RGB-D Interval FSD* - obtained from RGB-D depth channel, *Monocular Interval FSD* - obtained from monocular depth estimation without considering scale uncertainty, *Ground Truth FSD* - obtained directly from the 2D floor plan reference map considering known robot position (theoretical approach), and *Motion* - obtained only using odometry for particle weighting. The *Ground Truth FSD* sets

[1] <https://figshare.com/projects/Monocular_Interval_Extended_FSD/197902>

Table 6.4 – Localization metrics computed for different methods

Dataset	Method	SucD [m]		MeanParErrAC [m]		FErr [m]		Conv [%]
		μ	σ	μ	σ	μ	σ	%
alma-s1	Ground Truth FSD	*10.34*	*0.84*	*0.23*	*0.02*	*0.23*	*0.01*	*100.0*
	RGB-D Interval FSD	**11.62**	0.17	0.39	0.03	**0.45**	0.07	100.0
	Mon Interval FSD	11.43	0.10	**0.36**	0.03	0.46	0.04	100.0
	Mon Int Ext FSD (α=1/3)	12.45	1.38	0.39	0.05	0.49	0.08	100.0
	Motion	14.83	1.30	0.40	0.03	0.46	0.08	100.0
anto-s1	Ground Truth FSD	*8.31*	*0.34*	*0.27*	*0.01*	*0.30*	*0.02*	*100.0*
	RGB-D Interval FSD	11.05	1.42	0.45	0.04	0.48	0.14	100.0
	Mon Interval FSD	**9.77**	0.85	0.45	0.03	**0.41**	0.12	100.0
	Mon Int Ext FSD (α=1/3)	10.24	1.15	0.49	0.05	0.48	0.16	100.0
	Motion	17.82	2.45	**0.44**	0.04	0.54	0.10	100.0
pare-s1	Ground Truth FSD	*8.77*	*0.46*	*0.25*	*0.01*	*0.30*	*0.02*	*100.0*
	RGB-D Interval FSD	**17.68**	0.79	0.44	0.02	0.76	0.05	100.0
	Mon Interval FSD	23.14	9.13	0.55	0.10	0.69	0.13	23.0
	Mon Int Ext FSD (α=1/3)	18.48	2.17	0.53	0.05	0.71	0.05	87.0
	Motion	20.49	2.98	**0.43**	0.03	**0.54**	0.03	100.0
rx2-s1	Ground Truth FSD	*9.81*	*0.11*	*0.35*	*0.01*	*0.30*	*0.01*	*100.0*
	RGB-D Interval FSD	**10.15**	0.61	**0.49**	0.06	**0.56**	0.07	100.0
	Mon Interval FSD	10.68	1.25	0.55	0.10	0.82	0.06	83.0
	Mon Int Ext FSD (α=1/3)	10.78	1.13	0.58	0.11	0.80	0.07	73.0
	Motion	-	-	-	-	-	-	0.0

the theoretical upper bound performance while *Motion* sets the theoretical lower bound performance. *RGB-D Interval FSD*, *Monocular Interval FSD* and *Monocular Extended Interval FSD* ($\alpha = 1/3$) performance is expected to lie inside these bounds. Moreover, the performance of both monocular methods is expected to be lower or equal to the RGB-D method, since the monocular estimation uses the RGB-D depth channel as ground truth.

The metrics computed for all particle weighting strategies are presented in Table 6.4. For the maps *alma-s1* and *anto-s1*, all methods converge in 100% of the experiments. For *pare-s1*, the experiments that use a monocular camera in the measurement model fail to converge in some experiments. When *Monocular Interval Extended FSD* ($\alpha = 1/3$) is used, 87% of the experiments converge as opposed to 23% when *Monocular Interval FSD* is used. On the other hand, for *rx2-s1*, no experiment converges when we only use *Motion*. Additionally, when the monocular camera is used, we achieve 83% convergence for *Monocular Interval FSD* and 73% for *Monocular Interval Extended FSD* ($\alpha = 1/3$).

Results of the weighted mean particle position error computed for each experiment

and averaged over time among all experiments are presented in Figure 6.10. As a baseline, the error decreases more rapidly when using *Ground Truth FSD* for particle weighting. Considering only the methods that are implementable in a real scenario, the ones that use RGB-D or monocular camera usually converge faster than when only using *Motion*.

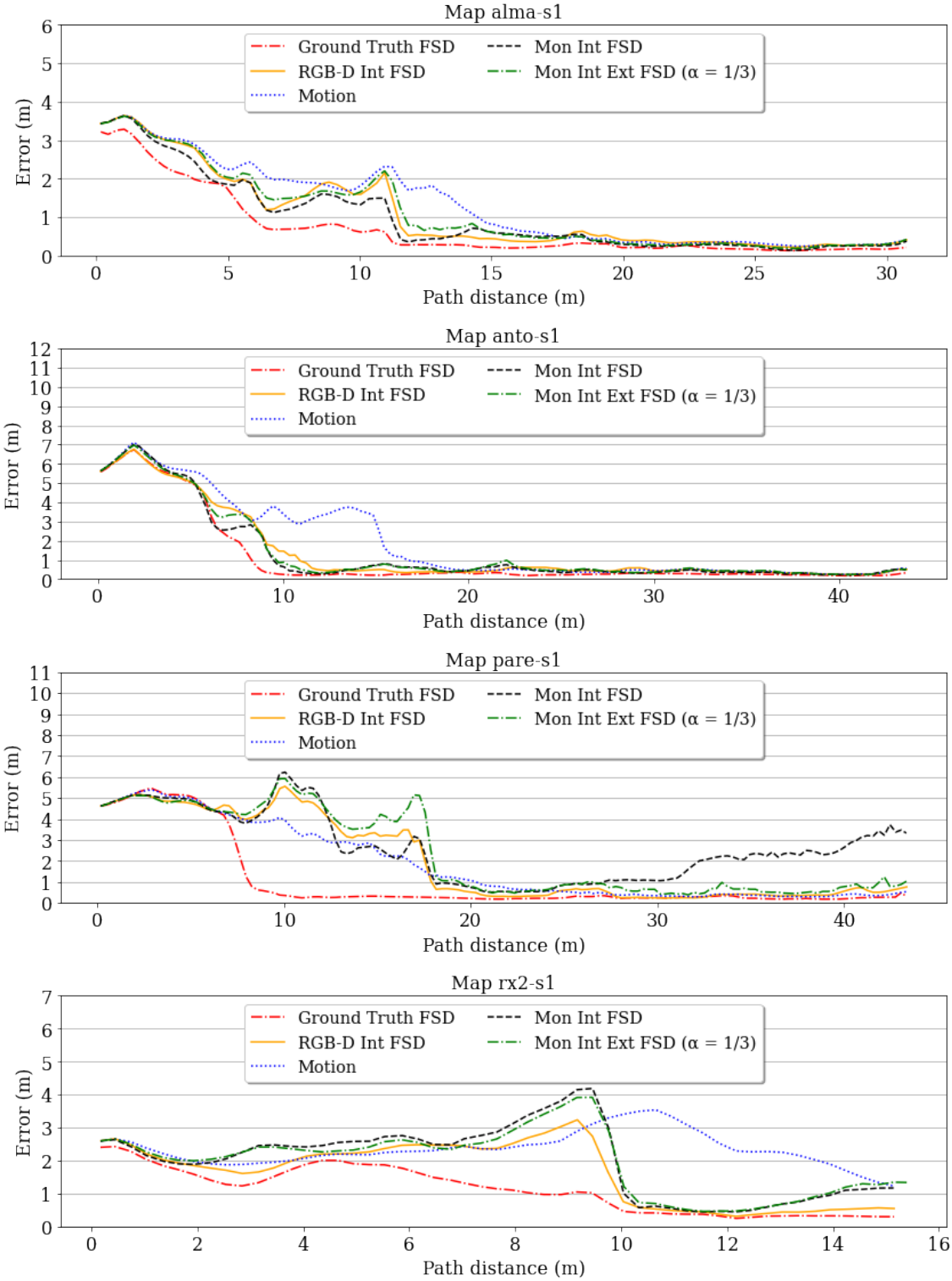


Figure 6.10 – Average among all experiments of the weighted mean particle error over time for different particle weighting strategies.

When we compare the performance from the RGB-D camera with the monocular camera, we observe in Table 6.4 that the RGB-D camera converges faster in 3 out of 4 maps. Moreover, the *succeed distance* of the monocular camera using *Monocular Interval FSD* is usually shorter than *Monocular Interval Extended FSD* ($\alpha = 1/3$).

Finally, the performance of using a low-cost and widely available monocular camera is not as good as the performance of using a RGB-D camera. However, when the monocular camera is used, we observe a better performance when compared to only using *Motion* in both convergence percentage and time to converge. Moreover, the use of *Monocular Interval Extended FSD* as opposed to *Monocular Interval FSD* increases the diversity of the particles and improves convergence percentage in some scenarios. An example is for the map *pare-s1*, where its complexity is a challenge for noisy observation models, so increasing the uncertainty enables convergence to the right position in longer trajectories.

7 CONCLUSION

In this work, we propose an indoor global localization method over a 2D floor plan based on FSD computed from monocular depth estimation. Experimental results using a dataset of image sequences collected in apartments demonstrate the feasibility of using depth information extracted from monocular cameras to compute the measurement model in an MCL strategy. Although monocular depth information is imperfect, we can still obtain satisfactory results in the localization when we add the estimated depth uncertainty to particle weighting.

We address the challenges of localization for consumer-grade robots using inexpensive sensors with a simple setup while considering the uncertainties introduced in such scenario. Our main contributions are: 1) a global localization method over 2D floor plans based on monocular depth estimation; 2) proposal of a new particle weighting strategy using FSD for noisy sensor measurements. The new proposal augments the diversity of the particle filter and thus increases its convergence to the right solution for scenarios where the trajectory is long enough. On the other hand, the increased uncertainty makes the filter take more time to converge and present a higher mean particle error after convergence.

There are a few limitations in the proposed approach that might prevent the success of robot localization. One limitation is relying on the performance of the monocular depth estimation model to compute the FSD. The model performance may not generalize well for new datasets and might be affected by unexpected obstacles and lighting conditions variations. Nevertheless, it is important to notice that the model we used was not trained with the dataset we used in the experiments, and the good results obtained demonstrate our method's resilience. Also, the correctness of depth maps might be affected by camera tilts and drifts. However, the method consistently showcases resilience and we believe it is capable of effectively handling drifts and camera tilts. Although the model presented an acceptable performance for the experiments dataset, we cannot guarantee similar performance for different datasets. More experiments with new datasets should be performed in the future to validate the proposed method generalization.

In order to mitigate the limitation of the model performance, the proposed method adds the depth estimation model uncertainty to the particle weighting strategy. This proposal makes the method more robust against imperfect depth maps. Additionally, evaluating the sensitivity of error with respect to distance in future work could provide valuable insights into further improving the method's robustness. However, it is still essential to

observe a portion of the environmental structure for a reliable localization. So, obstacles caused by furniture are also considered a limitation and might affect matching the FSD derived from observations with the reference FSD.

In the future, we plan to test different models for depth estimation. The main idea is to test models that output metric depth, so there would be no need to compute a metric scale (GUIZILINI et al., 2023). There is also the possibility to train a model with images from the environment where the robot will perform the localization. This might be necessary for larger maps where measurement model ambiguities are higher due to multiple similar areas in the map. Although it demands more time for the setup, a model trained specifically for a given environment tends to perform better and thus improve the localization performance.

REFERENCES

ALHMIEDAT, T. Fingerprint-based localization approach for wsn using machine learning models. **Applied Sciences**, v. 13, n. 5, 2023. ISSN 2076-3417. Available from Internet: <https://www.mdpi.com/2076-3417/13/5/3037>.

ALHMIEDAT, T. et al. A slam-based localization and navigation system for social robots: The pepper robot case. **Machines**, v. 11, n. 2, 2023. ISSN 2075-1702. Available from Internet: <https://www.mdpi.com/2075-1702/11/2/158>.

ALQOBALI, R. et al. A survey on robot semantic navigation systems for indoor environments. **Applied Sciences**, v. 14, n. 1, 2024. ISSN 2076-3417. Available from Internet: <https://www.mdpi.com/2076-3417/14/1/89>.

BHAT, S. F. et al. Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv, 2023. Available from Internet: <https://arxiv.org/abs/2302.12288>.

BONIARDI, F. et al. Robust lidar-based localization in architectural floor plans. In: **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2017. p. 3318–3324.

BONIARDI, F. et al. A pose graph-based localization system for long-term navigation in cad floor plans. In: . [s.n.], 2019. v. 112, p. 84–97. ISSN 0921-8890. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0921889018306092>.

BONIARDI, F. et al. Robot localization in floor plans using a room layout edge extraction network. In: . [S.l.: s.n.], 2019. p. 5291–5297.

BORENSTEIN, J.; KOREN, Y. Histogramic in-motion mapping for mobile robot obstacle avoidance. **IEEE Transactions on Robotics and Automation**, v. 7, n. 4, p. 535–539, 1991.

BORENSTEIN, J.; KOREN, Y. Histogramic in-motion mapping for mobile robot obstacle avoidance. **IEEE Trans. Robotics Autom.**, v. 7, p. 535–539, 1991.

DELLAERT, F. et al. Monte carlo localization for mobile robots. In: **Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)**. [S.l.: s.n.], 1999. v. 2, p. 1322–1328 vol.2.

DONG, X. et al. Towards real-time monocular depth estimation for robotics: A survey[-5pt]. **IEEE Transactions on Intelligent Transportation Systems**, v. 23, p. 1–22, 10 2022.

GAO, L.; KNEIP, L. Fp-loc: Lightweight and drift-free floor plan-assisted lidar localization. In: **2022 International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2022. p. 4142–4148.

GUIZILINI, V. et al. Towards zero-shot scale-aware monocular depth estimation. In: . [S.l.: s.n.], 2023. p. 9199–9209.

HAJI-ESMAEILI, M. M.; MONTAZER, G. Large-scale monocular depth estimation in the wild. **Engineering Applications of Artificial Intelligence**, v. 127, p. 107189, 2024. ISSN 0952-1976. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0952197623013738>.

HUANG, J. et al. Indoor positioning systems of mobile robots: A review. **Robotics**, v. 12, p. 47, 03 2023.

ITO, S. et al. W-rgb-d: Floor-plan-based indoor global localization using a depth camera and wifi. In: . [S.l.: s.n.], 2014. p. 417–422.

KHAN, F.; SALAHUDDIN, S.; JAVIDNIA, H. Deep learning-based monocular depth estimation methods—a state-of-the-art review. **Sensors**, v. 20, n. 8, 2020. ISSN 1424-8220. Available from Internet: <https://www.mdpi.com/1424-8220/20/8/2272>.

KIM, Y. et al. Deep monocular depth estimation via integration of global and local predictions. **IEEE Transactions on Image Processing**, v. 27, p. 4131–4144, 2018. Available from Internet: <https://api.semanticscholar.org/CorpusID:44100626>.

LEONARD, J.; DURRANT-WHYTE, H. Mobile robot localization by tracking geometric beacons. **IEEE Transactions on Robotics and Automation**, v. 7, n. 3, p. 376–382, 1991.

LEONARD, J.; DURRANT-WHYTE, H. Simultaneous map building and localization for an autonomous mobile robot. In: **Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91**. [S.l.: s.n.], 1991. p. 1442–1447 vol.3.

MAFFEI, R. **Translatting Sensor Measurements Into Texts for Localization and Mapping With Mobile Robots**. Thesis (PhD) — UFRGS, 2017.

MAFFEI, R. et al. Fast monte carlo localization using spatial density information. In: . [S.l.: s.n.], 2015. v. 2015, p. 6352–6358.

MAFFEI, R. et al. Global localization over 2d floor plans with free-space density based on depth information. In: **2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2020. p. 4609–4614.

MING, Y. et al. Deep learning for monocular depth estimation: A review. **Neurocomputing**, v. 438, p. 14–33, 2021. ISSN 0925-2312. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0925231220320014>.

OLSON, E. B. Real-time correlative scan matching. In: **2009 IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 2009. p. 4387–4393.

RANFTL, R.; BOCHKOVSKIY, A.; KOLTUN, V. Vision transformers for dense prediction. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2021. p. 12179–12188.

RANFTL, R. et al. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PP, p. 1–1, 08 2020.

RIBACKI, A. et al. Vision-based global localization using ceiling space density. In: **2018 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2018. p. 3502–3507.

RUIZ-SARMIENTO, J.; GALINDO, C.; GONZáLEZ-JIMéNEZ, J. Robot@home, a robotic dataset for semantic mapping of home environments. **The International Journal of Robotics Research**, v. 36, p. 027836491769564, 03 2017.

SILBERMAN, N. et al. Indoor segmentation and support inference from rgbd images. In: FITZGIBBON, A. et al. (Ed.). **Computer Vision – ECCV 2012**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 746–760. ISBN 978-3-642-33715-4.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)**. [S.l.]: The MIT Press, 2005. ISBN 0262201623.

WANG, X.; MARCOTTE, R. J.; OLSON, E. Glfp: Global localization from a floor plan. In: **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2019. p. 1627–1632.

WATANABE, Y. et al. Robust localization with architectural floor plans and depth camera. In: **2020 IEEE/SICE International Symposium on System Integration (SII)**. [S.l.: s.n.], 2020. p. 133–138.

WINTERHALTER, W. et al. Accurate indoor localization for rgb-d smartphones and tablets given 2d floor plans. In: **2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2015. p. 3138–3143.

ZHANG, L.; ZAPATA, R.; LéPINAY, P. Self-adaptive monte carlo localization for mobile robots using range finders. **Robotica**, v. 30, n. 2, p. 229–244, 2012.

ZHAO, C. et al. Monocular depth estimation based on deep learning: An overview. **CoRR**, abs/2003.06620, 2020. Available from Internet: <https://arxiv.org/abs/2003.06620>.