C-LOG: A Chamfer Distance Based Algorithm for Localisation in Occupancy Grid-maps

Lakshitha Dantanarayana*, Gamini Dissanayake, and Ravindra Ranasinge

Centre for Autonomous Systems, University of Technology, Sydney. NSW, Australia

Abstract

A novel algorithm for localising a robot in a known two-dimensional environment is presented in this paper. An occupancy grid representing the environment is first converted to a distance function that encodes the distance to the nearest obstacle from any given location. A Chamfer distance based sensor model to associate observations from a laser ranger finder to the map of the environment without the need for ray tracing, data association, or feature extraction is presented. It is shown that the robot can be localised by solving a non-linear optimisation problem formulated to minimise the Chamfer distance with respect to the robot location. The proposed algorithm is able to perform well even when robot odometry is unavailable and requires only a single tuning parameter to operate even in highly dynamic environments. As such, it is superior than the state-of-the-art particle filter based solutions for robot localisation in occupancy grids, provided that an approximate initial location of the robot is available. Experimental results based on simulated and public domain datasets as well as data collected by the authors are used to demonstrate the effectiveness of the proposed algorithm [1].

Keywords: robot localisation, distance functions, chamfer distance, optimisation, sensor models, environment representation

2016 MSC: 00-01, 99-00

*Corresponding author

Email address: lakshitha.dantanarayana@uts.edu.au (Lakshitha Dantanarayana*)

[1] Preliminary results of this work were initially published in [1]

1. Introduction

Localisation, or determining the pose (position and orientation) of a robot on a given map is a prime requirement for a robot operating autonomously in an environment. In situations where absolute tracking systems such as Global Positioning System (GPS) are not available, measurements obtained from sensors mounted on the robot are used for localisation. Algorithm used for localisation depends on the method used for representing the map of the environment.

When the map can be represented using geometric primitives such as points or line segments, extended Kalman filter (EKF) based algorithms are capable of efficiently estimating the robot pose within the map by fusing information gathered from robot odometry and observations to these primitives [2]. EKF based methods are generally computationally efficient but require an initial guess of the prior location of the robot. Therefore, such methods are incapable of solving problems such as the kidnapped robot problem, where the initial location of the robot is unknown. Furthermore, these methods exploit only a small proportion of the information available from sensors such as laser range scanners or RGB-D cameras due to the dimensionality reduction performed at the feature extraction step of the algorithm. EKF based methods usually require deriving motion and observation models and their Jacobians. These are specific to a given robot/sensor combination. Furthermore, extracting, defining, and explicitly associating geometric features and landmarks from the environment to the observations is also sensor specific, presenting an additional challenge.

An occupancy grid that classifies an environment into cells that are either occupied or free is one of the earliest and commonly used approaches for representing a metric map. When the environment description is available in the form of an occupancy grid map, particle filter based approaches are the preferred choice [3] for robot localisation due to their ability to exploit all the measurements available in a range scan. The particle filter based approaches use a

sensor model and a set of particles representing hypothesised robot locations to estimate the true pose of the robot. A sufficiently large number of particles, adequate to describe the probability density function of the robot pose, are required in order to generate location estimates with acceptable accuracy. Particle filters are relatively easy to implement and are capable of global localisation: the ability to deal with the situation when a suitable initial estimate for the robot pose is unavailable. The widely used adaptive Monte-Carlo localisation (AMCL) [4, 5, 6, 7, 8, 9], that is also available as a part of the popular Robot Operating System (ROS) [10] is a particle filter based approach for localisation. Within the particle filter framework, it is not straightforward to identify outliers or dynamic objects. In order to address this problem, AMCL uses a "mixture-model" which categorises the range readings by statistically analysing the probable causes of such outliers and penalising these observations during the particle update step. However, Thrun et al. [7] caution that this method would only work in certain limited situations and the categories should be analysed according to the environment.

Particle filters for localisation can be easily adapted to operate under a wide range of sensors and robot kinematic models. However, to be effective, particle filter implementations need to be tuned using a range of user defined parameters. As the computational burden of a particle filter is proportional to the number of particles used, further tuning is required to dynamically maintain the number of particles at an optimum level. The latest ROS implementation of the particle filter consists of 24 tunable parameters [11].

Optimisation based methods have also been proposed for robot localisation in the literature. These methods predominantly focus on feature based maps rather than on occupancy grid maps. In [12] a genetic optimisation algorithm is used to localise a mobile robot on a map consisting of geometric beacons. Genetic algorithms are also used in [13] for localising on a satellite image geomap of an outdoor environment using a laser range finder. Kwok et al. [14] proposes the use of evolutionary computing techniques which include genetic algorithms, particle swarm optimisation, and the ants system for feature based

localisation and demonstrates their effectiveness and robustness to noise and dynamic environments. Localisation of nodes in a wireless sensor network is a prominent application which relies heavily on optimisation based methods. Mao et al. [15] explains how different techniques are applied to this unique problem and how optimisation based methods can solve the wireless sensor networks localisation problem.

Scan matching is another popular method for robot localisation where an optimisation strategy that minimises the misalignment between observations from a sensor, typically a scan from the laser range finder and a map is used to estimate the robot location. Algorithms for scan matching proposed in the literature include Iterative Closest Point (ICP) [16, 17], Iterative Closest Line (ICL) [18] or Iterative Closest Surface (ICS) and probabilistic likelihood methods [7, 19]. In ICP, each laser endpoint in the query scan is associated with a point, line, or surface in the reference scan (or the map in case of localisation) using a distance metric such as Euclidean distance, after which a rigid body transformation [20] is used to compute the best alignment. A new set of data associations using the computed rigid body transformation is then used to repeat this process until convergence. In probabilistic scan matching methods, sensor error which is the difference between actual sensor measurement and the predicted sensor reading is used to update the likelihood of a given hypothesised robot pose. The predicted reading is estimated by algorithms such as ray-casting, which are computationally expensive, or likelihood fields [7] for which environment dependent tuning is essential, as it is an approximation to the ray-casting.

Distance function based maps are increasingly becoming utilised to capture geometries of environments [21, 22, 23, 24, 25]. The distance function not only encodes the occupied regions of the environment, but also provides a continuous measure of the distance, making it a much richer representation in comparison to an occupancy grid map. In KinectFusion, Newcombe et al. [22] extends the representation method proposed by Curless & Levoy [21] that uses Signed Distance Functions (SDFs) to encapsulate the three-dimensional (3D) surfaces that are incrementally constructed with the use of range readings from a depth

sensor. In contrast to 3D occupancy grid maps, which do not have a clear notion of where the surfaces of an environment are or how surfaces can be extracted, the work by Carrillo et al. [24] makes it apparent that there is a clear mathematical strategy for extracting surfaces in environments that are represented by SDFs. Work by Mullen et al. [26] and Chazal et al. [27] uses unsigned distance functions for 3D surface reconstruction and they point out that unsigned distance functions are much more robust to noise and outliers than SDFs.

In this paper, we propose distance functions as means of representing two-dimensional environments. We present a sensor model based on Chamfer distance [28] that can relate measurements from sensors such as laser range finders to a distance function based map. The proposed sensor model does not rely on feature extraction, data association or ray tracing operations. We use this sensor model in an optimisation based strategy that minimises the Chamfer distance to provide efficient means to localise a mobile robot when the initial pose of the robot is approximately known.

This paper is organised as follows: Section 2 introduces the distance function based environment representation and the Chamfer distance based sensor model. Section 3 formulates the optimisation problem that is used to localise the robot on the map as one of minimising the Chamfer distance, explores the properties of the optimisation problem and presents a strategy for solving it to find the robot pose. An experimental evaluation of the proposed algorithm based on simulation, public domain datasets as well as data collected in dynamic environments demonstrating the robustness of the proposed algorithm is presented in Section 4. Section 5 provides conclusions and future work.

2. Environment Representation and Sensor Model

In this section we develop a distance function based approach for representing the environment and derive the corresponding sensor model for a laser range finder.

2.1. Distance Functions for Environment Representation

For a given environment populated with objects, a distance transform or distance field is a map of the environment where any point of the distance field holds the shortest distance to the closest object boundary. The Euclidean distance is commonly used as a distance measure while other simple metrics such as City-block distance, chessboard distance, quasi-Euclidean distance or complex metrics such as Wasserstein metric that is used for 3D image reconstruction [27], are used as alternatives depending on the application and the need for computational efficiency.

When V is the set of occupied space in an environment, Euclidean distance function can be expressed by (1) at any given point $\mathbf{x}$ in space.

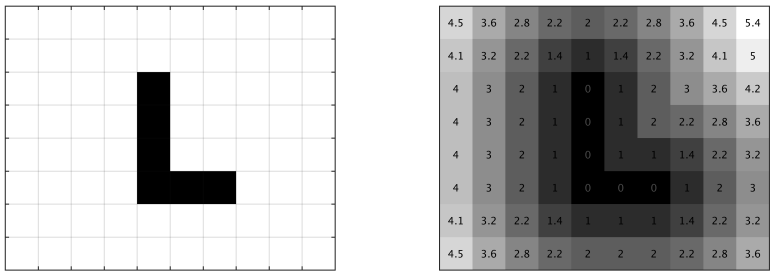
$$d_{DF} = DF(\mathbf{x}) = \min_{\mathbf{v_j} \in V} \|\mathbf{x} - \mathbf{v_j}\| \tag{1}$$

In Equation (1), the distance function is unsigned. However, when representing environments with closed shaped objects the sign of the distance function can be set to be either positive or negative depending on whether the query point $\mathbf{x}$ is outside ot inside the closed contour.

Distance functions can be computed on demand if the environment consists of geometric primitives. Alternatively it is possible to pre-compute distance function values at discrete intervals in space by quantising the environment to pixels (2D environment) or voxels (3D environment).

The operator used to generate a discrete distance function from an occupancy grid map is commonly known as a distance transform. When naïvely implemented, the distance transform process is an exhaustive search which depends on many factors including the resolution of quantisation. However, the algorithm proposed by Rosenfeld & Pfaltz [29] computes the distance transform efficiently by only two passes over any given two-dimensional environment. Furthermore, there are numerous algorithms, some of which rely on graphical processing units, that can compute distance transforms in real-time [30, 31, 32, 33].

Figure 1 represents an occupancy grid map as a binary image and its distance



(a) (b)

Figure 1: (a) An occupancy grid map as a binary image and (b) its distance function.

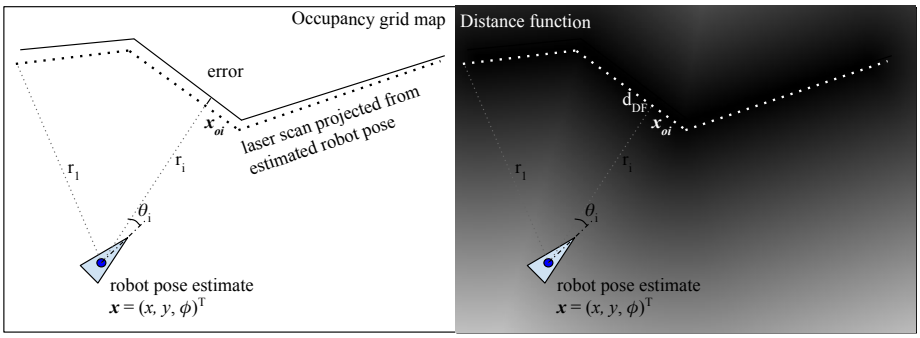
function, where the grey level of the image is used to represent the distance value.

2.2. Formulation of the Sensor Model

A sensor is the device through which the robot "sees" the world. It measures a physical quantity and converts the measurement into a tractable signal for the robot to use. The sensors enable the robot to be aware of its environment and to perform tasks reliably and accurately. The information (measurements of physical quantities) gathered by the sensors is known as *sensor measurements* or *observations*. Sensors that are commonly used in robots include: (i) contact sensors such as bump sensors and wheel encoders, (ii) inertial sensors such as accelerometers, gyroscopes and magnetometers(compasses), (iii) proximity sensors such as infra-red sensors, sonar sensors, radars, and laser range finders, (iv) visual sensors such as cameras and depth cameras, and (v) absolute sensors such as GPS and visual tracking systems. In practice, it is quite common to use multiple sensors on a robot as they can be used in a manner to complement each other to improve the overall accuracy and facilitate fault detection.

The selection of a sensor predominantly depends on the accuracy required by the task, suitability of a sensor for the operating environment of the robot and affordability. For example, even though a GPS based sensor is suitable for outdoor navigation, it cannot be used in indoor environments where the satellite reception is poor and subject to interference.

Sensors such as laser range finders have high accuracy and can be deployed



(a) (b)

Figure 2: Projection of the laser scan from an estimated robot pose, (a) on the binary occupancy grid map and (b) on the distance function.

in a wide range of environments. In the past, the high cost of these sensors have limited their use, but with the recent growth of robotics applications in the community, laser range finders with acceptable accuracy are now available at affordable prices.

The observations captured by a sensor are associated with a *sensor model*, which is an abstract representation of a physical sensor together with how the observations captured by a sensor are processed, interpreted and associated with the internal representation of the environment maintained by the robot.

Consider a laser range finder mounted on a robot placed in an environment that is represented using a distance function as shown in Figure 2. Observations corresponding to a single laser scan consisting of n range readings r at given bearings θ_i can be projected from a given robot pose $\mathbf{x} = (x,\ y,\ \phi)^\top$, using Equation (2) as shown in Figure 2a to obtain the observation vector in Cartesian space $\mathbf{x}_o$.

$$\mathbf{x}_{o_i} = \begin{Bmatrix} x_{o_i} \\ y_{o_i} \end{Bmatrix} = \begin{Bmatrix} x + r_i \cos(\theta_i + \phi) \\ y + r_i \sin(\theta_i + \phi) \end{Bmatrix} \tag{2}$$

Now a vector of distance readings can be extracted from the distance function

at points $\mathbf{x}_{o_i}$.

$$\mathbf{d}_{DF} = \begin{bmatrix} DF(\mathbf{x}_{o_1}) \\ . \\ . \\ DF(\mathbf{x}_{o_i}) \\ . \\ DF(\mathbf{x}_{o_n}) \end{bmatrix} \tag{3}$$

The covariance of this vector for a given robot pose, Σ_{DF} is a diagonal matrix which can be written as,

$$\Sigma_{DF} = \text{diag}(\sigma^2_{DF,\mathbf{x}_{o1}}, ..., ..., \sigma^2_{DF,\mathbf{x}_{o_i}}, ..., \sigma^2_{DF,\mathbf{x}_{o_n}}) \tag{4}$$

where $\sigma^2_{DF,\mathbf{x}_{o_i}}$ can be derived using Equation (5).

$$\sigma^2_{DF,\mathbf{x}_{o_i}} = J_{DF,\mathbf{x}_{o_i}} R \; J^{\top}_{DF,\mathbf{x}_{o_i}} \tag{5}$$

where $J_{DF,\mathbf{x}_{o_i}}$ is the Jacobian of the distance function at the query points $\mathbf{x}_{o_i}$ and the sensor noise is represented by R. Furthermore, $J_{DF,\mathbf{x}_{o_i}} = \frac{\partial d_{DF}}{\partial r}$, provided that the only contributing factor to sensor noise R is the laser range noise σ^2_r.

$$J_{DF,\mathbf{x}_{o_i}} = \frac{\partial d_{DF}}{\partial r} = \frac{\partial d_{DF}}{\partial x_o}\Big|_{\mathbf{x}_{o_i}} \cdot \frac{\partial x_o}{\partial r} + \frac{\partial d_{DF}}{\partial y_o}\Big|_{\mathbf{x}_{o_i}} \cdot \frac{\partial y_o}{\partial r} \tag{6}$$

The values, $\frac{\partial d_{DF}}{\partial x_o}$ and $\frac{\partial d_{DF}}{\partial y_o}$, can be precomputed by using the distance function of the map for improved efficiency.

We note here that the "*likelihood range finder model*" or the End Point (EP) model proposed by Thrun et al. [7] is a sensor model that uses a distance function based environment representation with a laser range finder. However, this is an empirical approximation using an empirical mixture model introduced to cope with the high computational expense associated with the ray-casting process.

The vector $\mathbf{d}_{DF}$ in Equation (3) is a measure of disparity between a map and a sensor measurement. A scalar measure of disparity has clear computational

advantages in the process of computing the measurement likelihood. In the context of image processing and computer vision, literature is abundant with scalar measures of disparity between distance functions and binary images.

Chamfer distance is one of many such distance metrics available that does not require defining explicit corresponding point pairings. Hausdorff distance [34, 35], another popular method that is used in many applications, captures one point which has the worst mismatch from a set of points as opposed to Chamfer distance which captures average mismatch of all given points. First introduced by Burrow et al. [28] in 1977, Chamfer distance based template matching has gone through many implementations, improvements and value additions over the years which includes making it robust in rotation (i.e. minor orientation changes) [36], scale changes [37], resolution changes, and even robust in high clutter [38].

In computer vision literature, Chamfer distance is defined and used for template matching with binary images, where a semblance of the binary query shape is located within a larger reference image. Let $U = \{\mathbf{u_i}\}$ and $V = \{\mathbf{v_j}\}$ be sets of query and reference images respectively. The Chamfer distance between U and V is given by the average of distances between each point $\mathbf{u}_i \in U,\ n(U) = n$ and its nearest edge in V,

$$d_{CD} = CD(U, V) = \frac{1}{n} \sum_{\mathbf{u_i} \in U} \min_{\mathbf{v_j} \in V} |\mathbf{u_i} - \mathbf{v_j}|. \tag{7}$$

Here n is the number of points in U.

With the use of a distance function, it is possible to reduce the cost function (7) to (8) so that it can be evaluated in linear time, $\mathcal{O}(n)$ [38].

$$d_{CD} = CD(U, V) = \frac{1}{n} \sum_{\mathbf{u_i} \in U} DF(\mathbf{u_i}). \tag{8}$$

The Chamfer distance is a sum of positive distances and is defined for unsigned distance functions.

In the case of two-dimensional template matching using Chamfer distance, the reference image and the template are both binary edge images which can be

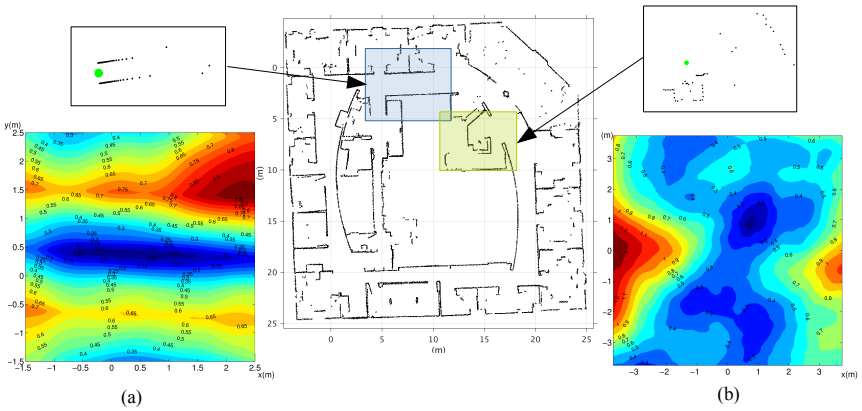


Figure 3: Variation of Chamfer distance against robot location, (x, y) at two different locations of the Intel research labs dataset, in the vicinity of the true robot pose. The corresponding laser scan (not to scale) is given above the contour-plots. ϕ is set to its true value.

obtained using an edge filter on the original images. The highest computational complexity in this context lies on the distance transform process to create the distance function from the reference edge image which should be done for every image frame. However, as discussed before, recent high-speed implementations of distance transform enable faster execution and have even made it possible to use Chamfer distance for people recognition and tracking on surveillance footage in real-time [33]. It is important to note that when the distance function is used to represent a static environment map, its calculation is a one off process and therefore does not impact the computational cost of the localisation process described in this paper.

Using Equation (8), the Chamfer distance for a laser scan obtained from a robot operating in an environment that is represented with the unsigned distance function, DF can be written as shown in Equation (9).

$$d_{CD} = \frac{1}{n} \sum_{i=1}^{n} DF(\mathbf{x}_{o_i}). \tag{9}$$

Figure 3a and Figure 3b present the variation of Chamfer distance relative to a hypothesised robot location $(x,\ y)$ that varies in the vicinity of the true pose,

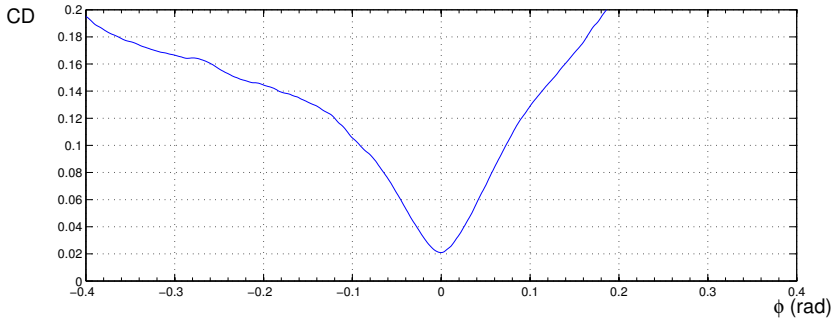


Figure 4: Chamfer distance variation in the vicinity of the true robot pose x and y at their true values and orientation ϕ varied.

with approximate coordinates of the true pose of $(1.1,\ 1.1)m$ and $(0.45,\ 0.45)m$ respectively. If there is no measurement noise, the minimum Chamfer distance, which will be equal to zero, is obtained when the robot is placed at its true pose in the map and the laser scan is perfectly aligned. Figure 4 shows the variation of Chamfer distance when x and y are kept at their true values and the orientation ϕ is varied between ± 0.4 radians for the robot's true location used in Figure 3b.

Partial derivatives of Chamfer distance can be deduced with the use of partial derivatives of DF as shown in Equation (11).

$$\nabla d_{CD} = \begin{pmatrix} \frac{\partial d_{CD}}{\partial x} & \frac{\partial d_{CD}}{\partial y} & \frac{\partial d_{CD}}{\partial \phi} \end{pmatrix}^{\top} \tag{10}$$

$$\begin{pmatrix} \frac{\partial d_{CD}}{\partial x} \\ \frac{\partial d_{CD}}{\partial y} \\ \frac{\partial d_{CD}}{\partial \phi} \end{pmatrix} = \begin{pmatrix} \frac{1}{n} \sum \frac{\partial DF}{\partial x_o}|_{\mathbf{x}_{o_i}} \cdot \frac{\partial x_o}{\partial x} \\ \frac{1}{n} \sum \frac{\partial DF}{\partial y_o}|_{\mathbf{x}_{o_i}} \cdot \frac{\partial y_{o_i}}{\partial y} \\ \frac{1}{n} (\sum \frac{\partial DF}{\partial x_o}|_{\mathbf{x}_{o_i}} \cdot \frac{\partial x_o}{\partial \phi}|_{r_{o_i}} + \sum \frac{\partial DF}{\partial y_{o_i}}|_{\mathbf{x}_{o_i}} \cdot \frac{\partial y_o}{\partial \phi}|_{r_{o_i}}) \end{pmatrix} \tag{11}$$

As with DF, the partial derivatives $\frac{\partial DF}{\partial x_o}|_{\mathbf{x}_{o_i}}$ and $\frac{\partial DF}{\partial y_o}|_{\mathbf{x}_{o_i}}$ can also be pre-computed and stored.

We note here that if a distance transform is used to obtain the distance function from an occupancy grid map, a continuous approximation such as a cubic spline is needed to interpolate the distance function values to estimate distances

and the derivatives in continuous space. The derivatives of the distance transform are discontinuous at boundaries between occupied and unoccupied space as well as cut-loci [39]. Using an appropriate spline approximation, impact of these discontinuities on gradient based optimisation algorithms can be avoided. Apart from splines, Gaussian processs have also been suggested as smoothing functions for distance function [25], but these incur a heavy computational cost in the application presented in this paper.

3. Localisation Algorithm

This section describes a method for localising a robot on a two-dimensional map using information gathered using a laser range finder mounted on a robot. It uses the distance function based representation and the Chamfer distance based sensor model that we presented in Section 2.2.

Robot localisation problem can be solved by finding the robot pose that minimise a cost function C, which is defined as a measure of mismatch between a set sensor reading $\mathbf{z}$ and the map $\mathbf{m}$. The sensor model as described in Section 2.2 essentially defines such a cost function in the vicinity of the true pose.Robot localisation problem can therefore be expressed as,

$$\underset{x\ y\ \phi}{\arg\min}\ d_{CD}(\mathbf{x}_o, DF) \tag{12}$$

where DF is the distance function of the occupancy grid map of the environment $\mathbf{m}$ and $\mathbf{x}_o$ is the template generated using the laser scan $\mathbf{z}$ from Equation (2) with the potential robot pose $\mathbf{x} = (x,\ y,\ \phi)^\top$.

Given that the objective function in Equation (12) is twice differentiable when a cubic spline approximation is used, this unconstrained non-linear optimisation problem can be solved using a variety of gradient based techniques. In the experiments presented in Section 4 the Matlab implementation of the trust-region algorithm was used. The partial derivatives of the objective function with respect to the robot pose $\mathbf{x}$ are required for solving the optimisation

problem described by Equation (12), are given in Equation (10) and Equation (11).

A gate that admits only the values that are smaller than a maximum error as shown in Equation (13) can be used to eliminate the obvious outliers from the laser range finder measurements.

$$d_{DF}(\mathbf{x}_{o_i}) \leq \Delta\phi.r_i + \Delta x + \Delta y \tag{13}$$

where Δx, Δy and $\Delta\phi$ are the maximum expected error in the initial guess. In the experiments $0.15m$ was used for Δx and Δy while $\Delta\phi$ was set to 0.05rad. This is the only tuning parameter required for this algorithm and clearly it is relatively easy to establish.

4. Experimental Results

We use experiments conducted on three datasets to illustrate the capabilities of the proposed localisation algorithm.

Dataset 1, is based on a simulation conducted on ROS stage environment so that the ground truth is available for the evaluation. The robot in this simulation is equipped with a Hokuyo laser that provides laser scans. Table 1 presents the sensor properties and the parameters used in the simulation.

Dataset 2 is a publicly available dataset from the Intel research laboratories, Seattle, USA. In this dataset the robot travels three loops in an office building. Map of the environment and the ground truth are not available. Therefore, laser range scans gathered during the third loop is used with the GMapping [40] algorithm to generate the occupancy grid map for evaluation.

Dataset 3, was collected at the Broadway Shopping Centre, Sydney, Australia. The data was collected during normal operating hours of the shipping centre, and therefore the environment cluttered and crowded. The robot was equipped with a Hokuyo UTM-30LX laser range finder. Odometry was not used.

Table 1: Sensor properties and noise parameters for Dataset 1

Laser range finder	simulated Hokuyo UTM-30LX
Maximum range	$30m$
Minimum range	$0.1m$
Field of view	$270°$
Angular resolution	$1/4°$
No. of readings per scan	1080
Laser range finder measurement noise	$\mathcal{N}(0, 0.02^2 m^2)$
Linear velocity noise	$\mathcal{N}(0, 0.04^2 m^2 s^{-2})$
Angular velocity noise	$\mathcal{N}(0, 0.01^2 rad^2 s^{-2})$

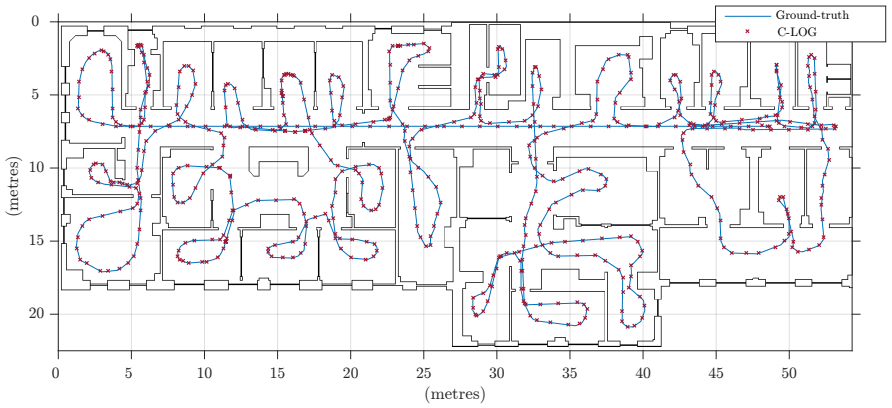
Table 2: Pose errors for Dataset 1

	RMS errors	
	Position ($m^2 \times 10^{-3}$)	Orientation (deg^2)
C-LOG(proposed)	0.30 ± 2.17	0.98 ± 5.43
Particle Filter(AMCL-Beam)	1.46 ± 2.08	8.16 ± 23.40

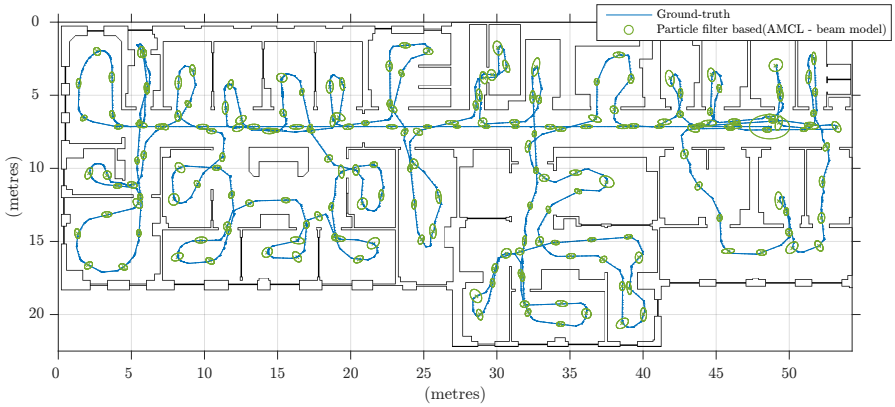
4.1. Accuracy of Robot Pose Estimates

As the ground-truth is available in Dataset 1, we can quantitatively compare the output of the proposed localisation algorithm against ground-truth. As a comparison, we use the particle filter implementation AMCL available for ROS with its beam range finder model. Figure 5 illustrates location estimates obtained with the proposed optimisation algorithm and AMCL while Figure 6 presents the errors of the estimates along the entire robot trajectory against the ground-truth. The average pose errors for the proposed algorithm and AMCL are shown in Table 2.

Figure 7 and Figure 8 qualitatively compare the results from the proposed algorithm and AMCL using the Dataset 2. As it can be seen in Figure 7c, the map recovered by projecting laser scans from the poses estimated by the

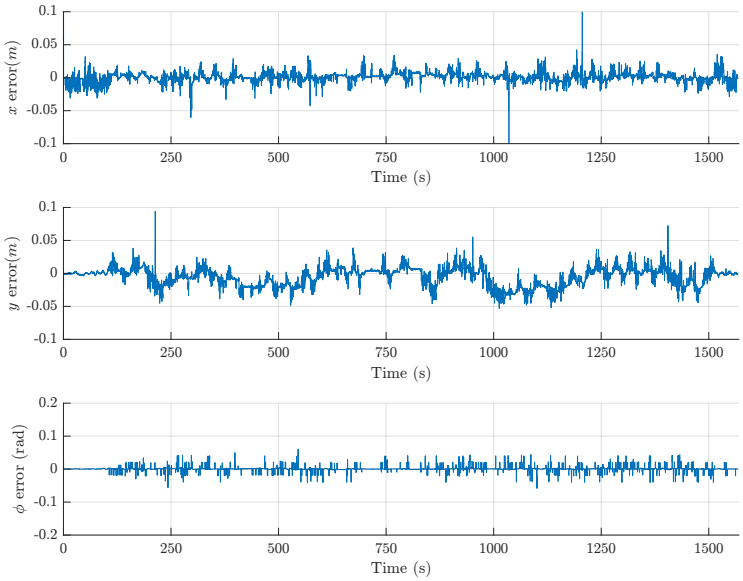


(a)

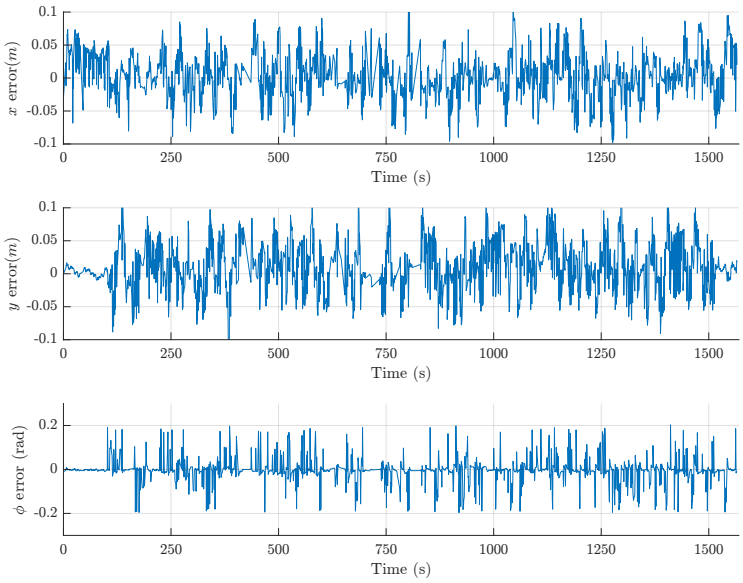


(b)

Figure 5: Estimated robot location and the ground-truth for Dataset 1. (a) Optimisation algorithm and (b) AMCL particle filter algorithm with beam range finder model.



(a)



(b)

Figure 6: Errors in the location estimate for (a) proposed C-LOG algorithm and (b) AMCL particle filter algorithm with beam range finder model.

proposed algorithm has well defined walls that aligns with the original map as opposed to the map recovered from AMCL illustrated in Figure 8c. This indicates that the poses produced by the proposed algorithm is more accurate.

4.2. Performance in Dynamic Environments

As previously mentioned, the Dataset 3 was collected under natural conditions in a crowded environment. Therefore, in this dataset, the laser observations are mostly corrupted by people. Figure 9 is a sparse illustration of the crowd movement during data collection obtained by projecting all the readings from the laser range finder from estimated robot poses. Figure 10 shows the estimated robot poses obtained from the proposed C-LOG algorithm. As ground truth is not available, poses obtained using a SLAM algorithm that was used to construct the map of the environment is shown for comparison. It is clear that C-LOG performs well even in the presence of significant people movement.

To further evaluate the performance of the proposed algorithm under dynamic scenarios, we performed a simulation experiment using Dataset 1. In this experiment we artificially corrupted a percentage of input laser scans with a uniform random distribution of $\mathcal{U}(0,\ r_i)$. Figure 11 shows the root-mean-square (RMS) error at different degrees of corruption. C-LOG algorithm continue to localise without losing track even with up to 60% of the input sensor measurements corrupted by dynamic objects, while AMCL fails at 40% corruption.

Situations where the optimisation algorithm fails to converge was dealt with by processing the next laser scan with the current best estimate of the robot pose.

4.3. Computational Cost

Figure 12 compares the computational cost of the proposed algorithm with AMCL available in ROS. Dataset 1 was used for obtaining the average time to process one laser scan. The laser range finder in Dataset 1 produces 1081 laser readings per scan. For AMCL, the number of particles was set to be

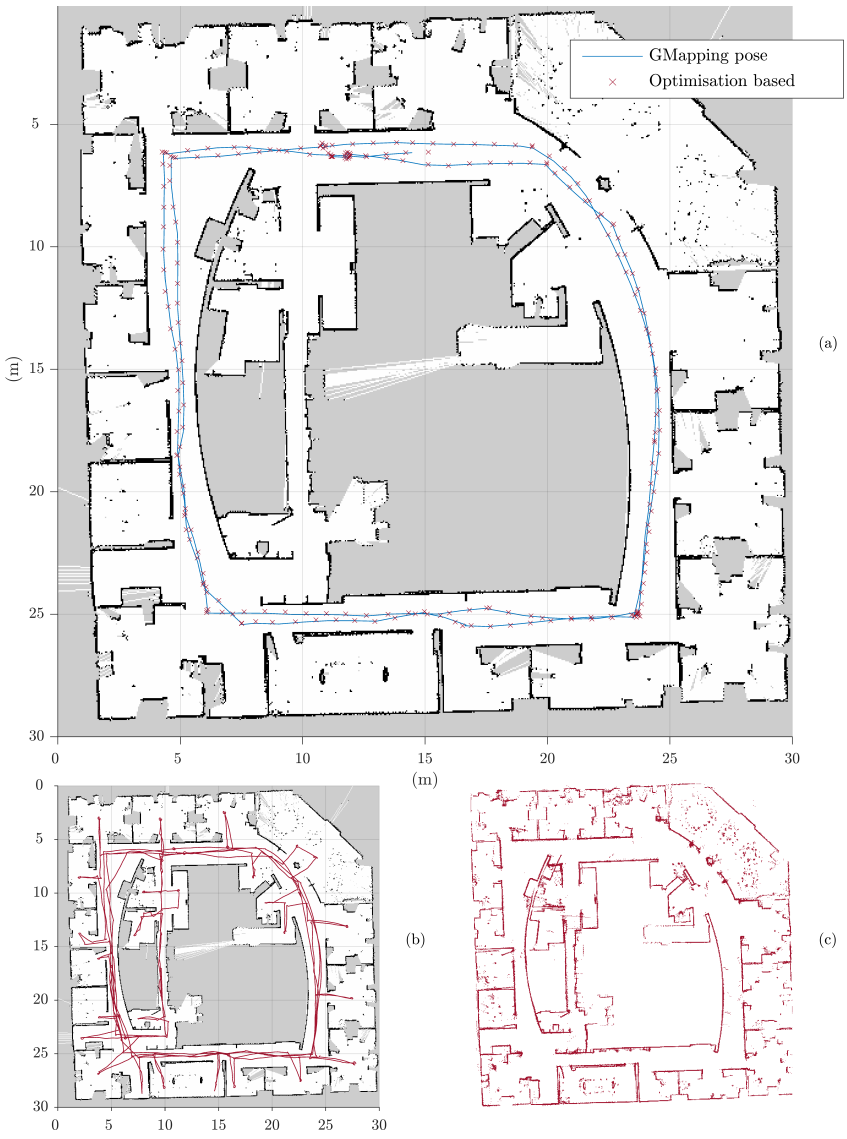


Figure 7: Trajectory of the robot in the Intel Dataset, using C-LOG algorithm. (a) GMapping poses and results obtained from the optimisation based localisation strategy. (b) Complete trajectory of the robot. (c) Projection of the laser scan from the estimated pose.

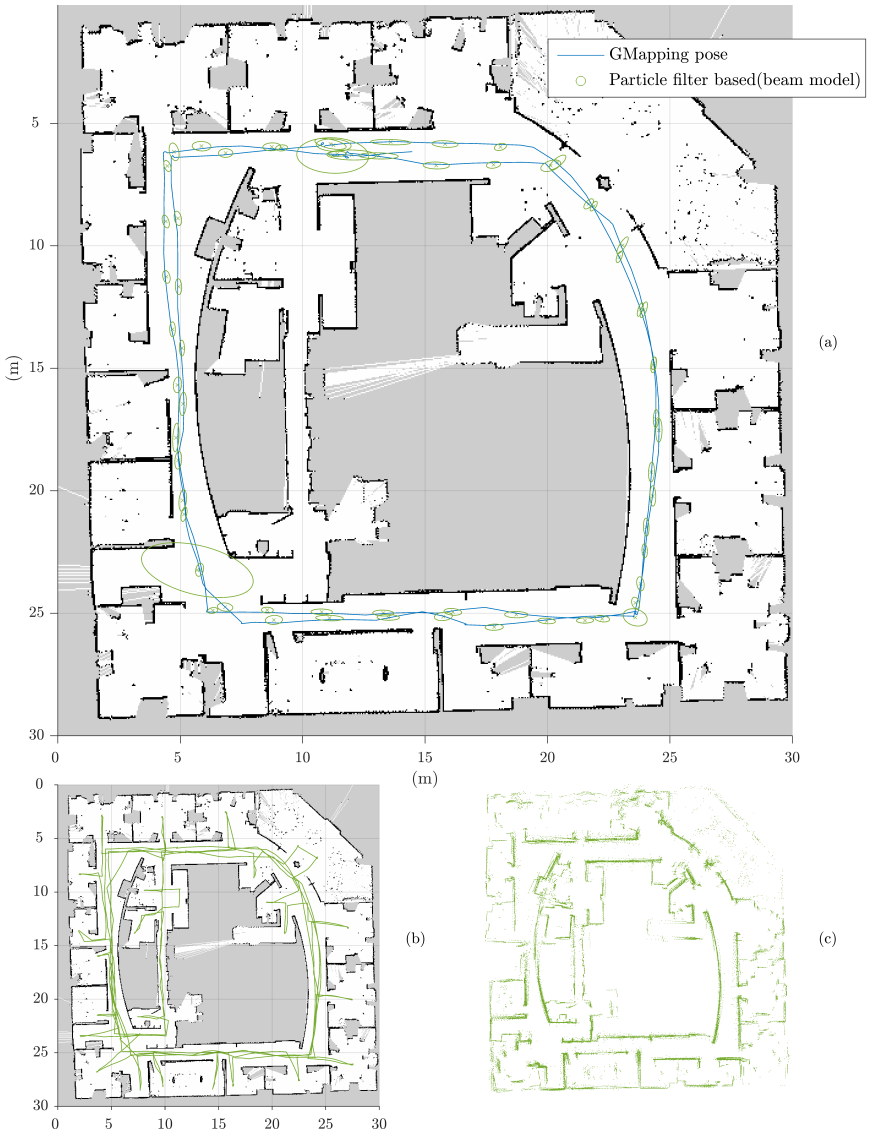


Figure 8: Trajectory of the robot in the Dataset 2. (a) GMapping poses and results obtained from AMCL, using beam based likelihood model. (b) Complete trajectory of the robot. (c) Projection of the laser scan from the estimated pose.

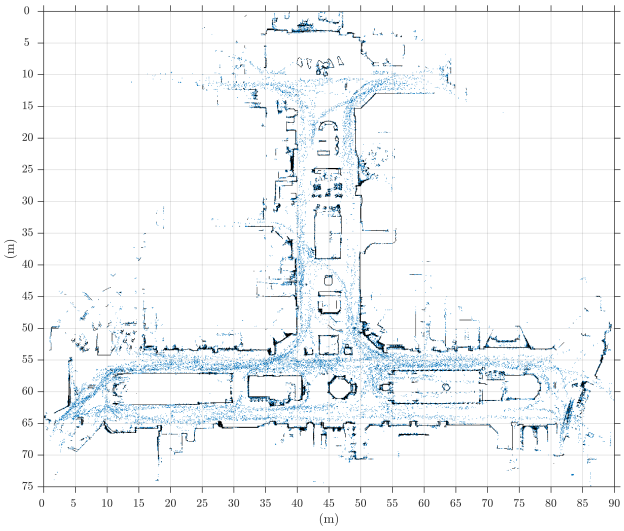


Figure 9: A sparse illustration of crowd movement during collection of Dataset 3 over 29.54 minutes.

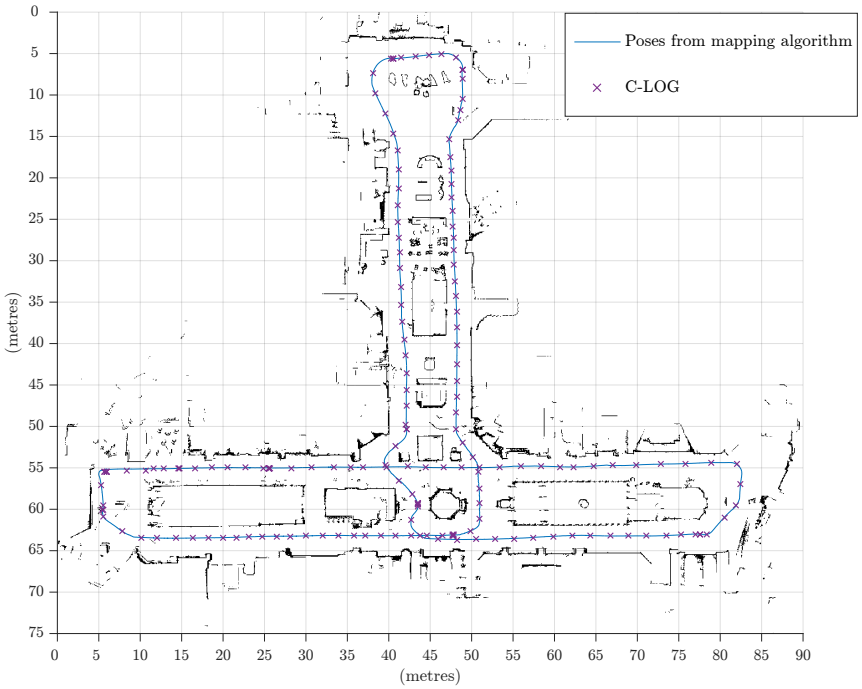


Figure 10: Trajectory of the Robot using C-LOG on the Dataset 3.

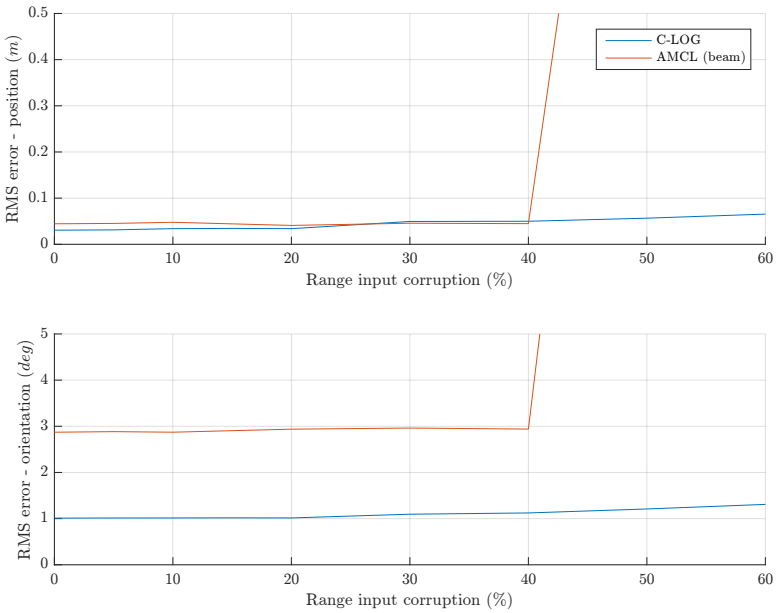


Figure 11: The RMS errors of the proposed algorithm and AMCL (beam model) when input sensor measurements are artificially corrupted to simulate dynamic objects in the environment.

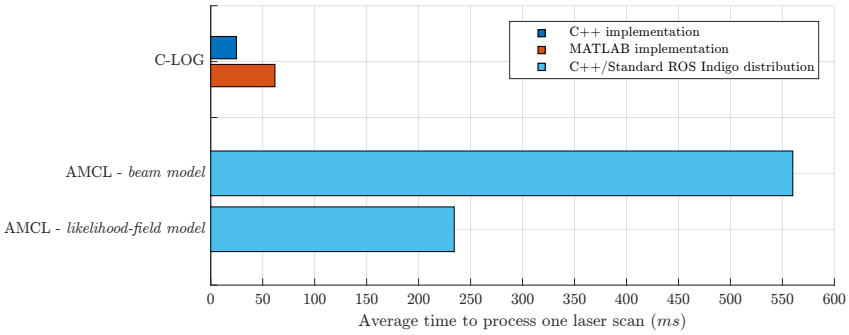


Figure 12: Per scan execution time for localisation algorithms.

fixed at 5000 particles. Execution times for both the standard beam-based and likelihood-field sensor models of AMCL are shown. It can be seen that the AMCL with the beam based likelihood function takes the longest time due to the complexity of the ray-casting process. It should be noted that the code distributed with ROS may not be the most efficient implementation of AMCL.

5. Discussion and Conclusion

In this paper, we presented a novel localisation algorithm for robots equipped with a laser range finder operating in a two-dimensional environment. The proposed algorithm uses a distance function based method for representing the environment and a sensor model that uses Chamfer distance to relate range measurements from the sensor to the map of the environment. The sensor model does not require explicit data association or extraction of features from the sensor readings. An optimisation algorithm that minimises Chamfer distance with respect to the robot pose is shown to be effective in obtaining an estimate for the robot location on the map, provided an approximate initial location of the robot is known. The proposed algorithm does not require odometry measurements.

We used multiple experiments based on a simulation, a public domain dataset, and data collected in a crowded environment to demonstrate the effectiveness of

the algorithm. This algorithm was illustrated to be more accurate and computationally efficient than the widely used particle filter based algorithm AMCL. Experimental results demonstrate that the optimisation based technique proposed in this paper provides a competitive solution to the problem of robot localisation within an occupancy grid. One of the main advantages observed is that the algorithm does not require tuning parameters, except for a relatively large gate for filtering outliers from laser range data. This is due to the fact that the models of process and observation uncertainty are not used within the optimisation algorithm. Future work includes using these models to compute the robot pose uncertainty and fuse odometry observations, if available. Impact of the map resolution on the localisation accuracy as well as possibilities for using a continuous representation of the distance function, as opposed to a cubic spline approximation of the discrete distance transform are under investigation.

References

[1] L. Dantanarayana, R. Ranasinghe, G. Dissanayake, C-LOG: A Chamfer Distance based method for localisation in occupancy grid-maps, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 376–381. doi:10.1109/IROS.2013.6696379.

[2] J. Leonard, H. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation 7 (3) (1991) 376–382. doi:10.1109/70.88147.

[3] C. Stachniss, W. Burgard, Particle Filters for Robot Navigation, Foundations and trends in Robotics 3 (4) (2014) 211–282. doi:10.1561/2300000013.

[4] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte Carlo localization for mobile robots, in: IEEE International Conference on Robotics and Automation, Vol. 2, IEEE, 1999, pp. 1322–1328. doi:10.1109/ROBOT.1999.772544.

[5] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artificial Intelligence 128 (1-2) (2001) 99–141. doi:10.1016/S0004-3702(01)00069-8.

[6] S. Thrun, A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots, The International Journal of Robotics Research 20 (5) (2001) 335–363. doi:10.1177/02783640122067435.

[7] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, The MIT Press, 2005.

[8] D. Fox, Adapting the Sample Size in Particle Filters Through KLD-Sampling, The International Journal of Robotics Research 22 (12) (2003) 985–1003. doi:10.1177/0278364903022012001.

[9] S. S. Srinivasa, D. Ferguson, C. J. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, M. V. Weghe, HERB: A home exploring robotic butler, Autonomous Robots 28 (1) (2010) 5–20. doi:10.1007/s10514-009-9160-9.

[10] Open Source Robotics Foundation, Robot Operating System(ROS) (2016).

[11] B. P. Gerkey, Adaptive Monte-Carlo Localization for ROS (2008).

[12] L. Moreno, J. M. Armingol, S. Garrido, A. d. l. Escalera, M. A. Salichs, A Genetic Agorithm for Mobile robot Localization using Ultrasonic sensors, Journal of Intelligent and Robotic Systems 34 (2) (2002) 135–154. doi:10.1023/A:1015664517164.

[13] C. U. Dogruer, A. B. Koku, M. Dolen, Global Urban Localization of an Outdoor Mobile Robot with Genetic Algorithms, in: European Robotics Symposium 2008, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 103–112. doi:10.1007/978-3-540-78317-6{_}11.

[14] N. Kwok, D. Liu, G. Dissanayake, Evolutionary computing based mobile robot localization, Engineering Applications of Artificial Intelligence 19 (8) (2006) 857–868. doi:10.1016/j.engappai.2006.01.020.

[15] G. Mao, B. Fidan, B. D. Anderson, Wireless sensor network localization techniques, Computer Networks 51 (10) (2007) 2529–2553. doi:10.1016/j.comnet.2006.11.018.

[16] P. J. Besl, N. D. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 239–256. doi:10.1109/34.121791.

[17] S. Thrun, M. Diel, D. Hähnel, Scan Alignment and 3-D Surface Modeling with a Helicopter Platform, in: The 4th International Conference on Field and Service Robotics, 2003.

[18] A. Censi, An ICP variant using a point-to-line metric, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 19–25. doi:10.1109/ROBOT.2008.4543181.

[19] E. B. Olson, Real-time correlative scan matching, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 4387–4393. doi:10.1109/ROBOT.2009.5152375.

[20] B. K. P. Horn, Closed-form solution of absolute orientation using unit quaternions, Journal of the Optical Society of America 4 (4) (1987) 629–642.

[21] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96, ACM Press, New York, New York, USA, 1996, pp. 303–312. doi:10.1145/237170.237269.

[22] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, A. Fitzgibbon, KinectFusion: Real-time dense surface mapping and tracking, in: 2011 10th IEEE International Symposium on Mixed and Augmented Reality, IEEE, 2011, pp. 127–136. doi:10.1109/ISMAR.2011.6092378.

[23] T. Whelan, M. Kaess, M. Fallon, Kintinuous: Spatially Extended KinectFusion, in: RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras, 2012.

[24] H. Carrillo, Y. Latif, J. Neira, J. Castellanos, Towards measuring uncertainty in volumetric signed distance function representations for active SLAM, in: Workshop on Multi VIew Geometry in RObotics (MVIGRO) In conjunction with RSS 2014, 2014.

[25] S. Kim, J. Kim, GPmap: A Unified Framework for Robotic Mapping Based on Sparse Gaussian Processes, in: Field and Service Robotics, Springer International Publishing, 2015, pp. 319–332. doi:10.1007/978-3-319-07488-7{_}22.

[26] P. Mullen, F. De Goes, M. Desbrun, D. Cohen-Steiner, P. Alliez, Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets, Computer Graphics Forum 29 (5) (2010) 1733–1741. doi:10.1111/j.1467-8659.2010.01782.x.

[27] F. Chazal, D. Cohen-Steiner, Q. Mérigot, Geometric Inference for Probability Measures, Foundations of Computational Mathematics 11 (6) (2011) 733–751. doi:10.1007/s10208-011-9098-0.

[28] H. Barrow, J. Tenenbaum, R. Bolles, H. Wolf, Parametric correspondence and chamfer matching: Two new techniques for image matching, in: 5th international joint conference on Articial intelligence - Volume 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1977, p. 659663.

[29] A. Rosenfeld, J. Pfaltz, Distance functions on digital pictures, Pattern Recognition 1 (1) (1968) 33–61. doi:10.1016/0031-3203(68)90013-7.

[30] T. Schouten, E. van den Broek, Fast exact Euclidean distance (FEED) transformation, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Vol. 3, IEEE, 2004, pp. 594–597. doi:10.1109/ICPR.2004.1334599.

[31] J. Schneider, M. Kraus, R. Westermann, {GPU}-Based Real-Time Discrete Euclidean Distance Transforms With Precise Error Bounds, in: International Conference on Computer Vision Theory and Applications (VISAPP), 2009, pp. 435–442.

[32] J. Schneider, M. Kraus, R. Westermann, GPU-Based Euclidean Distance Transforms and Their Application to Volume Rendering, in: A. Ranchordas, J. M. Pereira, H. J. Araújo, J. M. R. S. Tavares (Eds.), Computer Vision, Imaging and Computer Graphics. Theory and Applications, Vol. 68 of Communications in Computer and Information Science, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 215–228. doi:10.1007/978-3-642-11840-1{_}16.

[33] M. Rauter, D. Schreiber, A GPU accelerated Fast Directional Chamfer Matching algorithm and a detailed comparison with a highly optimized CPU implementation, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2012, pp. 68–75. doi:10.1109/CVPRW.2012.6238897.

[34] F. Hausdorff, Grundzüge der Mengenlehre, Veit & co., 1914.

[35] D. P. D. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, Comparing images using the Hausdorff distance, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (9) (1993) 850–863. doi:10.1109/34.232073.

[36] J. Shotton, A. Blake, R. Cipolla, Multiscale categorical object recognition using contour fragments., IEEE transactions on pattern analysis and machine intelligence 30 (7) (2008) 1270–81. doi:10.1109/TPAMI.2007.70772.

[37] G. Borgefors, Hierarchical chamfer matching: a parametric edge matching algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (6) (1988) 849–865. doi:10.1109/34.9107.

[38] M. Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, Fast directional chamfer matching, in: 2010 IEEE Computer Society Conference on Com-

puter Vision and Pattern Recognition, IEEE, 2010, pp. 1696–1703. doi:10.1109/CVPR.2010.5539837.

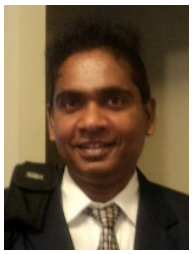
[39] M. Jones, J. Baerentzen, M. Sramek, 3D distance fields: a survey of techniques and applications, IEEE Transactions on Visualization and Computer Graphics 12 (4) (2006) 581–599. doi:10.1109/TVCG.2006.56.

[40] G. Grisetti, C. Stachniss, W. Burgard, Improved techniques for grid mapping with Rao-Blackwellized particle filters, IEEE Transactions on Robotics 23 (1) (2007) 34–46. doi:10.1109/TRO.2006.889486.

Notes on Contributors

Lakshitha Dantanarayana is a Ph.D. student at the Centre for Autonomous Systems, University of Technology, Sydney (UTS), Australia. He obtained his B.Sc.Eng.(Hons.) degree in Electronics and Telecommunications Engineering from the University of Moratuwa, Sri Lanka in 2010. His research is focused on localisation, mapping, SLAM, and navigation for assistive robotics.

Gamini Dissanayake is the James N. Kirby Professor of Mechanical and Mechatronic Engineering at University of Technology, Sydney (UTS). He graduated in Mechanical and Production Engineering from the University of Peradeniya. He taught at University of Peradeniya, National University of Singapore and University of Sydney before joining UTS in 2002. He founded the UTS Centre for Autonomous Systems; a team of fifty staff and students working in Robotics. His work on robot navigation has resulted in one of the most cited journal publications in robotics. He has been involved in developing robots for a range of industry applications including cargo handling, mining, infrastructure maintenance and aged care.



Ravindra Ranasinghe received B.Sc.Eng.(Hons.) degree in 1995, specializing in Computer Science and Engineering, from the University of Moratuwa, Sri Lanka. He received his Ph.D. in wireless communication protocols from the University of Melbourne, Australia. Before joining the Centre for Autonomous Systems at University of Technology, Sydney (UTS), he worked in several technology start-up companies in USA, Australia, and Sri Lanka. He is currently a Senior Research Fellow at the

Centre for Autonomous Systems, University of Technology, Sydney. He is a member of the IEEE since 1997. His research interests include mobile robotics wireless sensor networks, assistive robotics, active object recognition and machine learning techniques for condition assessments.