

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

Lakshitha Dantanarayana, Ravindra Ranasinghe and Gamini Dissanayake.

**Abstract**—In this paper, the problem of localising a robot within a known two-dimensional environment is formulated as one of minimising the Chamfer Distance between the corresponding occupancy grid map and information gathered from a sensor such as a laser range finder. It is shown that this non-linear optimisation problem can be solved efficiently and that the resulting localisation algorithm has a number of attractive characteristics when compared with the conventional particle filter based solution for robot localisation in occupancy grids. The proposed algorithm is able to perform well even when robot odometry is unavailable, insensitive to noise models and does not critically depend on any tuning parameters. Experimental results based on a number of public domain datasets as well as data collected by the authors are used to demonstrate the effectiveness of the proposed algorithm.

## I. INTRODUCTION

Localisation of a robot relative to a given map of the environment based on information gathered from sensors mounted on the robot has been a well-studied problem. 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 geometric beacons in the environment [1]. If the environment map is available in the form of an occupancy grid, particle filter [2] has become the most popular technique for robot localisation when a range-bearing sensor such as a laser range finder or a RGB-D sensor like Microsoft@Kinect is available.

Within the past decade, a variety of algorithms for solving the much more complex problem of localising a robot while at the same time building a map of the environment (SLAM) have also become available. While the most popular method for SLAM in the early literature have been based on estimation methods such as EKF [3] and particle filters [4][5], solving SLAM using optimisation techniques has recently emerged as the preferred solution [6][7][8], due to their robustness and ease of use. The motivation for the work presented in this paper was the desire to examine whether the advantages associated with optimisation based techniques extend to solving the problem of robot localisation within an occupancy grid map.

This paper is organised as follows. Section II formulates the robot localisation problem as one of minimising the Chamfer Distance between an occupancy grid map and

a scan from a range-bearing sensor. It also explores the properties of the optimisation problem and presents a strategy for solving this to find the robot pose together with the associated uncertainty. Section III presents related work while an experimental evaluation of the proposed algorithm, based on a number of public domain datasets is presented in section IV. A discussion and conclusions are presented in section V.

## II. PROBLEM FORMULATION

Consider the problem of obtaining the robot pose (position and orientation)  $X_r = (x_r y_r \phi_r)^T$  of a mobile robot with respect to an *a-priori* map  $M$  using a collection of data points  $S$  obtained from a sensor such as a laser range finder. This is essentially an exercise in finding the best alignment between the map  $M$  and the scan  $S$ . If some function  $D(S, M)$  that quantifies the mismatch between  $M$  and  $S$  at some robot pose  $X_r$  exists, then the robot localisation problem can be stated as:

$$\underset{x_r \ y_r \ \phi_r}{\operatorname{argmin}} D(S, M) \quad (1)$$

In typical particle filter implementations [9], the mismatch between the map and a range observation is computed using the difference between actual range measurement and the expected range to a corresponding point on the map. Ray tracing from a particle placed at a potential robot pose is used to determine correspondences. Given its iterative nature, ray tracing is not a suitable strategy for defining  $D(S, M)$  in (1), if the latter is to be solved using a numerical process.

The computer vision literature is abound with distance measures between image contours, for example Chamfer Distance [10], Hausdorff Distance [11], that do not require defining explicit corresponding point pairings. Chamfer Distance gives the average mismatch between  $M$  and  $S$  for all points, rather than the worst mismatch between  $M$  and  $S$  as in Hausdorff distance. In this work it is proposed to use the Chamfer Distance to compute  $D(S, M)$ .

### A. Chamfer Distance

The Chamfer Distance ( $CD$ ) has been successfully used for template matching where a semblance of the template image edge map is located within a query image edge map. Let  $U = \{u_i\}$  and  $V = \{v_j\}$  be sets of template and query image edge maps respectively. The Chamfer Distance between  $U$  and  $V$  is given by the average of distances

Centre for Autonomous Systems, Faculty of Engineering and IT, University of Technology, Sydney, 15 Broadway, Ultimo NSW 2007, Australia  
{Lakshitha.Dantanarayana, Ravindra.Ranasinghe, Gamini.Dissanayake}@uts.edu.au

between each point  $\mathbf{u}_i \in U$  and its nearest edge in  $V$ ,

$$d_{CD}(U, V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i - \mathbf{v}_j|. \quad (2)$$

Here  $n$  is the number of points in  $U$ .

The Distance Transform is an implicit shape representation whose pixel value indicates the minimum distance from that point to the closest object. Once a distance map DT of an occupancy grid map  $M$  is generated using an unsigned distance function via (3), which specifies the Euclidean distance from each pixel to the nearest edge pixel in  $V$ , it is clear that computing the Chamfer Distance of a scan  $S$  with respect to the map  $M$  is very straightforward.

$$DT(x) = \min_{\mathbf{v}_j \in V} |x - \mathbf{v}_j| \quad (3)$$

The Distance Transform matrix,  $DT$  for any point  $x$  in the map  $M$ , can be pre-computed and stored in a matrix, taking a significant computational burden away from the algorithm. For a given template the distance transform can be computed in two passes over the image [12] and using which the cost function (2) can be evaluated in linear time  $O(n)$  via (4) [13].

$$d_{CD}(U, V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} DT(\mathbf{u}_i). \quad (4)$$

Even though the distance from any point to the nearest object is a continuous value, distance transformation as described above quantises these distances into pixels. Furthermore, the derivatives of a distance transform function are not continuous at points which belong to the map or to the cut locus [14]. As the intention is to use the distance transform to calculate the Chamfer Distance and use it within an optimisation algorithm, an interpolation algorithm based on a cubic spline approximation is used to compute the distance transform and its derivatives at any given location in the map. All future references to DT in this paper refers to the interpolated version of the DT matrix.

It is important to note that in contrast to its vision applications the algorithm proposed in this paper uses a single fixed query image which is the occupancy grid map against multiple templates which are laser scans. Therefore, the distance transform and its derivatives with respect to the image coordinates can be precomputed and stored together with the interpolation coefficients.

1) *Behaviour of the Chamfer Distance in the Vicinity of the True Robot Pose:* To further explore the properties of the Chamfer Distance lets consider the following example. It is assumed that the robot operates in a 2-Dimensional planar environment and the occupancy grid map of the environment is available. Robot position with respect to the map is  $(x_r, y_r)$  and the orientation is  $\phi_r$ . The robot has a 2D laser range scanner which reports range ( $r$ ) and bearing ( $\theta$ ) data with respect to the robot coordinate frame.

The Fig. 1a shows an extracted part of the map from Intel Research Lab dataset [15] and Fig. 1b shows its DT.



Fig. 1: (a) Map and (b) Distance Transform Image for a part of the Intel Research Labs Dataset

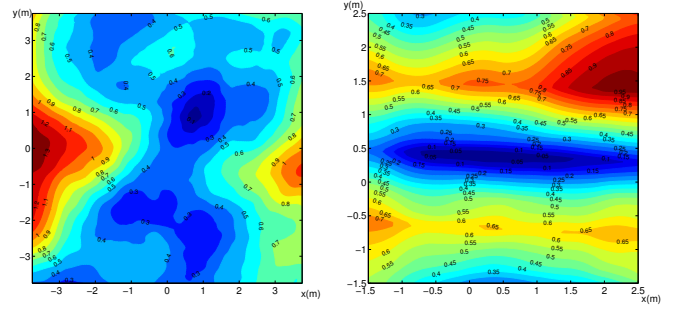


Fig. 2: Chamfer Distance variation in the vicinity of the true robot pose for two parts of the Intel Research Labs dataset.

Let the estimated pose of the robot be  $X_r = (x_r, y_r, \phi_r)^T$ . When the robot makes an observation, the range scanner captures  $n$  number of range ( $r_i$ ) and bearing ( $\theta_i$ ) scans. The template  $S$  that needs to be aligned with the map is given by (5).

$$X_{o_i} = \begin{Bmatrix} x_{o_i} \\ y_{o_i} \end{Bmatrix} = \begin{Bmatrix} x_r + r_i \sin(\theta_i + \phi_r) \\ y_r + r_i \cos(\theta_i + \phi_r) \end{Bmatrix} \quad (5)$$

Using (3), Chamfer Distance can now be estimated as,

$$CD(x_r, y_r, \phi_r) = \frac{1}{n} \sum_{i=0}^{n-1} DT(X_{o_i}) \quad (6)$$

As mentioned before DT matrix as well as the coefficients for cubic spline interpolation for this matrix would be pre-calculated. The Fig. 2a shows the contour map of the Chamfer Distance at the estimated robot pose  $(x_r, y_r)$  is varied in the vicinity of the true pose, which is at  $(1.1m, 1.1m)$ . Meanwhile, Fig. 3 shows the effect of  $\phi_r$  on the Chamfer Distance.

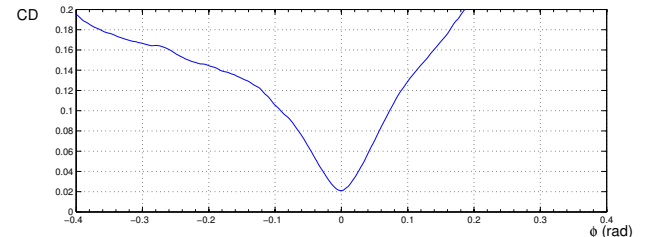


Fig. 3: Chamfer Distance variation with  $\phi_r$  when  $(x_r, y_r)$  are kept at their true values.

It is clear that the Chamfer Distance behaves well in the vicinity of the global minimum, which corresponds to the true pose of the robot. The basin of attraction in the  $xy$  plane appear to be around  $\pm 1m$  for this particular example. Fig. 2a also reveals that there are two local minima about  $3m$  south of the global minimum. Fig. 2b shows the variation of the Chamfer Distance in a region with a long corridor. As expected, the contours correspond to a narrow valley, indicating that there is less information available for localising the robot in direction along the corridor.

### B. Optimisation Algorithm

Robot localisation problem expressed in (1) can now be written as,

$$\underset{x_r \ y_r \ \phi_r}{\operatorname{argmin}} CD(X_o, DT) \quad (7)$$

Where  $DT$  is the distance transform of the occupancy grid map  $M$  and  $X_o$  is the template generated using the laser scan  $S$  in (5) with the potential robot pose  $x_r \ y_r \ \phi_r$ . Therefore we propose to use an optimisation algorithm to solve this problem in order to obtain the optimum pose  $X_r$  that will yield minimum Chamfer Distance,

Given that the objective function in (7) is twice differentiable, this unconstrained non-linear optimisation can be solved using a variety of gradient based techniques. In the experiments presented in section IV Matlab implementation of the Trust-region algorithm was used.

1) *Data Association*: In the context of localisation, data association is the process of matching uncertain measurements to known feature points in the map. As the Chamfer Distance represents the entire laser scan (cluster of laser readings), its value at any given robot pose  $X_r$  represents a measure of the placement of the whole scan, not just any individual laser reading. Therefore, data association does not need to be explicitly considered when the objective function for robot localisation is evaluated.

2) *Gradient Calculation*: The partial derivatives of the objective function with respect to the robot pose  $X_r$  are required for solving the optimisation problem described by (7). These gradients represented in (8) can be expanded with the help of (6) as (9).

$$\nabla CD = \left( \frac{\partial CD}{\partial x_r} \quad \frac{\partial CD}{\partial y_r} \quad \frac{\partial CD}{\partial \phi_r} \right)^T \quad (8)$$

$$\begin{pmatrix} \frac{\partial CD}{\partial x_r} \\ \frac{\partial CD}{\partial y_r} \\ \frac{\partial CD}{\partial \phi_r} \end{pmatrix} = \begin{pmatrix} \frac{1}{n} \sum \frac{\partial DT}{\partial x_{o_i}} \cdot \frac{\partial x_{o_i}}{\partial x_r} \\ \frac{1}{n} \sum \frac{\partial DT}{\partial y_{o_i}} \cdot \frac{\partial y_{o_i}}{\partial y_r} \\ \frac{1}{n} \left( \sum \frac{\partial DT}{\partial x_{o_i}} \cdot \frac{\partial x_{o_i}}{\partial \phi_r} + \sum \frac{\partial DT}{\partial y_{o_i}} \cdot \frac{\partial y_{o_i}}{\partial \phi_r} \right) \end{pmatrix} \quad (9)$$

$\frac{\partial DT}{\partial x_{o_i}}$  and  $\frac{\partial DT}{\partial y_{o_i}}$  in (9) can be obtained by looking up the gradients of the distance transform with respect to global  $x, y$  coordinates. As mentioned previously distance transform and its derivatives can be precomputed using the grid map and stored to make the gradient computation computationally

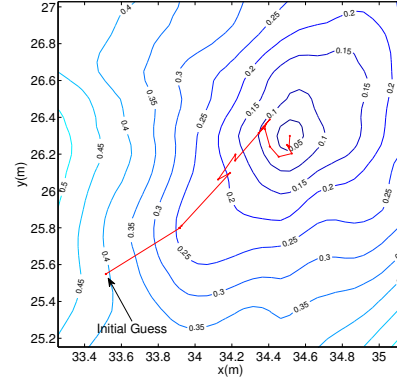


Fig. 4: Trace of  $X^*$  in  $x,y$  plane as the optimisation algorithm proceeds for the map that corresponds to Fig. 2a

efficient. The remaining components of the gradient can be analytically derived from (5).

Algorithm 1 details the steps of solving the localisation problem.

---

#### Algorithm 1 Localise Robot

---

**Require:**  $DT, \frac{\partial DT}{\partial x_o}, \frac{\partial DT}{\partial y_o}$ , Initial pose estimate  $X_1 = (x_1 \ y_1 \ \phi_1)^T$

**loop** for every input Sensor reading =  $S_{r\theta}$

**function** OPTIMISE( $X_1$ )

**return**  $X^* = (x \ y \ \phi)^T$  at  $\min CD$

**end function**

Report current robot pose =  $X^*$

new  $X_1 = \text{CalcNextPoseGuess}(X^*)$

▷ Compute the best guess to the pose estimate at the next time step: Update  $X^*$  using odometry information or use  $X^*$  if odometry is not available.

**end loop**

---

Fig. 4 shows the manner in which the optimisation algorithm converges to the solution from an initial guess which is about  $1.2m$  away from the true location, with an angle offset of  $\frac{\pi}{8}$ . As seen, the initial guess is within the basin of attraction. There are 33 iterations in this instance as the initial location is far away from the actual pose.

### C. Uncertainty of the Pose Estimate

As the final robot pose estimate  $\hat{X}_r$  is obtained through solving an optimisation problem, an explicit function relating the sensor measurements  $(r_i, \theta_i)$  to  $\hat{X}_r$  is not available. Monte-Carlo sampling process was used to obtain the variation of the pose estimate in response to the noise in the range measurements (with  $\sigma_r = 0.02m$ ) and is shown in Fig. 5. Estimates of the uncertainty are low, indicating that the influence of the errors in the map, including quantisation errors which are not considered in the above analysis, are likely to dominate the uncertainty estimate in practice.

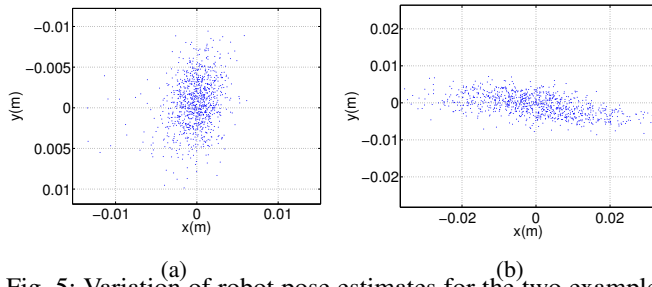


Fig. 5: Variation of robot pose estimates for the two example locations shown in Fig. 2 a-b. As expected, the variation along the  $x$  axis is much higher than that along the  $y$  axis for the location corresponding to Fig. 2b.

It is important to note that, in contrast to EKF or Particle filter based localisation algorithms, the statistics related to the robot pose estimate at the previous time step is not exploited during the proposed optimisation algorithm. If odometry is available, it is possible to compute an estimate for the robot pose using the incremental motion from the robot pose at the previous time step and easily fuse the two results as they are independent of each other.

It's also worthwhile to mention that implicit function theorem can be exploited to analytically compute the influence of the noise in the measurements on the estimated robot pose  $\hat{X}_r$  [16].

### III. RELATED WORK

#### A. Optimisation based Localisation

Use of optimisation based methods for localisation have been reported in the literature. These predominantly focus on feature based maps rather than on occupancy grids. In [17] a genetic optimisation algorithm (GA) is used to localise a mobile robot on a geometric beacon based *a-priori* map. This algorithm was also used in [18] for localising on a satellite image geo-map of an outdoor environment using a laser range finder sensor. Another example of the use of evolutionary computing for feature based localisation is [19].

Localisation using Wireless Sensor Networks rely heavily on optimisation based methods. [20] explains how different techniques are applied to this unique problem and how optimisation based methods can solve this localisation problem.

#### B. Chamfer Distance

Chamfer Distance, first introduced in [10], is a popular technique used to match template images to part a large image. There have been many implementations, improvements and value additions to this algorithm which includes making it robust in rotation (i.e. minor orientation changes) [21], scale changes [22], resolution changes and even in high clutter [13].

In the case of template matching, the highest computational complexity lies on the creation of the Distance Transform map which should be created for every input image. Recent implementation advances to the algorithm of creating the DT includes multi-CPU and GPU based implementations [23] which enable faster execution and

hence make it possible to use Chamfer Distance in people recognition and tracking on surveillance footage in real-time.

### IV. EXPERIMENTAL RESULTS

In this section the proposed C-LOG algorithm is evaluated using two real world data sets:

- **Dataset 1:** Intel Research Lab [15] dataset.
- **Dataset 2:** Massachusetts Institute of Technology, CSAIL Building [24] dataset.
- **Dataset 3:** A dataset collected at the University of Technology, Sydney, Centre for Autonomous Systems.

These datasets consists of robot odometry and information gathered from a laser range finder.

MathWorks®MATLAB 2013a environment running on a single core of an Intel®Xeon®E5-2690, 2.9GHz computer was used for the implementation of C-LOG and conducting experimental evaluations.

In Dataset 1, a robot takes three loops on a floor with multiple rooms. Map of the environment was generated with GMapping algorithm [5] using the information gathered during loop 3. The experiments were carried out using the remainder of the data to ensure the integrity of the evaluation. Dataset 2 does not have multiple independent runs. Therefore, every other laser scan was used to create the map, and rest was used for evaluating the C-LOG algorithm. The Dataset 3 has multiple runs in the environment and was collected using a TurtleBot™ with a Hokuyo® 30m Laser Range Finder. The first run that was used to create the map was done with the environment free of any dynamic objects while the data used for the experiment was collected about eight months after the first run when there were people moving about.

Maps generated from GMapping was processed to remove isolated pixels of size less than 4. A simple gating function (10) was used to eliminate the obvious outliers from the laser range finder data. The gate admits only the values that are smaller than a maximum error. This is the only tuning parameter required for this algorithm and clearly it is relatively easy to establish.

$$DT(X_{o_i}) \leq \Delta\theta.r_i + \Delta x + \Delta y \quad (10)$$

where  $\Delta x$ ,  $\Delta y$  and  $\Delta\phi$  are the maximum expected error in the initial guess. In the experiments  $0.15m$  were used for  $\Delta x$  and  $\Delta y$  while  $\Delta\phi$  was set to  $0.05rad$ .

#### A. Results for Dataset 1

Results from experiments with C-LOG algorithm presented in **Algorithm 1** together with Dataset 1 to localise a robot are presented in this section. At the beginning, an approximate initial robot pose was provided. In subsequent iterations, the optimum pose estimate given by the optimisation algorithm was updated with encoder odometry measurements that were available in the dataset. This updated value was then used as the initial guess to the robot pose for the next iteration.

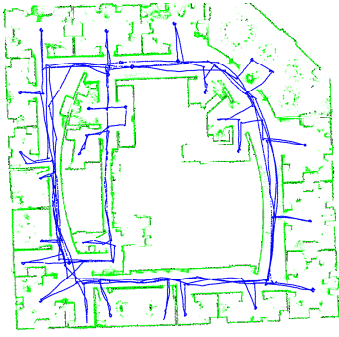


Fig. 6: Pose Map of Dataset 1 obtained using C-LOG with plot of all raw laser reading for each pose illustrating the pose estimates.

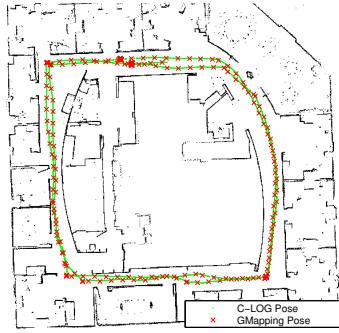


Fig. 7: Comparison of pose estimates from GMapping and C-LOG on Dataset 1.

Fig. 6 shows all the poses on the map with the plot of the raw laser readings superimposed at each pose to from the map. It is clear that the laser scans are accurately aligned with the map throughout the run, demonstrating the capability of C-LOG to solve the localisation problem within an occupancy grid map. The video attachment displays this experiment more clearly with the projection of laser readings from the pose estimate of the algorithm to visually verify how the algorithm performs on this dataset.

Comparison between the pose estimates obtained after GMapping was completed and the poses obtained with C-LOG are shown in Fig. 7.

### B. Robustness of the Algorithm

Scenarios listed below were used to further evaluate the behaviour of the proposed algorithm.

- 1) Artificially reduce the rate at which laser scans are used.
- 2) Ignore odometry observations.
- 3) Ignore odometry observations except orientation estimate. This simulates a scenario where robot is not equipped with encoders but a solid-state gyroscope is available for orientation estimates.
- 4) Corrupt laser range finder readings to simulate dynamic objects in the environment. In this scenario, a percentage of the range readings were replaced with a random number from a uniform distribution in  $[0, \text{actual range}]$ .

Experiment	Mean Squared Error		
	$x(m^2)$	$y(m^2)$	$\phi(rad^2)$
With odometry, every $10^{th}$ scan	0.0081	0.0181	0.0122
Without odometry, every $4^{th}$ scan	0.0013	$3.0954 \times 10^{-4}$	$3.7267 \times 10^{-4}$
With orientation update only, every $5^{th}$ scan	0.0011	$1.6880 \times 10^{-5}$	$1.7246 \times 10^{-6}$
Without odometry, all scans	$2.1508 \times 10^{-4}$	$1.3832 \times 10^{-5}$	$9.2253 \times 10^{-7}$
<b>Presence of Dynamic objects</b>			
10% of the scan corrupted	$3.0809 \times 10^{-4}$	$6.5229 \times 10^{-5}$	$4.4230 \times 10^{-6}$
25% of the scan corrupted	0.0013	$5.6580 \times 10^{-4}$	$1.780 \times 10^{-5}$
50% of the scan corrupted	0.0118	0.1140	$2.7101 \times 10^{-4}$
60% of the scan corrupted	0.0518	0.1308	$9.6825 \times 10^{-4}$

TABLE I: MSE at several runs of the algorithm under different conditions

The Table I shows the Mean Squared Errors (MSE) of the post estimates under different scenarios in comparison with the results presented in Fig. 6. Experiments were stopped when the algorithm failed to converge to the actual pose at some point in the trajectory. In summary, when odometry is available, C-LOG performs well even when every tenth laser scan is used. Without odometry, it is possible to localise with every fourth scan while every fifth scan is adequate if an orientation estimate is available. Furthermore, C-LOG was able to tolerate significant corruption of the range observation resulting in accurate pose estimates with 25% of scans containing false data. It fails to converge only when 60% of the scans contain corrupted data. Note that pose errors particularly in  $x$  and  $y$  are quite large at this error level. It was observed that even beyond these limits, the algorithm would still perform well over the majority of the robot trajectory, but would fail to accurately localise in some places.

### C. Results for Dataset 2

Fig. 8 shows the pose maps from C-LOG and GMapping obtained using dataset 2 demonstrating the effectiveness of the proposed algorithm.

### D. Results for Dataset 3

This demonstrates how C-LOG performs in a real dynamic environment. The Fig. 9 shows the plot of the pose estimates from the algorithm and the plot of the laser on those poses. The trails of moving people on the map can be clearly seen on the figure.

## V. DISCUSSION & CONCLUSION

Experimental results presented in section IV demonstrate that the optimisation based technique proposed in this paper provides a competitive solution to the problem of robot localisation within an occupancy grid. Even with a Matlab implementation, each iteration of the optimisation process took only 20 msec. The algorithm converged within 14





Fig. 8: Robot poses of C-LOG and GMapping with the occupancy grid map constructed using C-LOG poses for Dataset 2.



Fig. 9: Pose Map of Dataset 3 obtained using C-LOG with plot of all raw laser reading for each pose. The fact that the laser scans align well with the map demonstrates the accuracy of the localisation algorithm. The laser readings in the middle of the corridors depict people moving around the environment.

iterations on average, making it near real-time for the Intel data set where laser range finder operates at about 5 Hz.

A C or a C++ implementation and further experimentation to confirm the real-time performance of the proposed algorithm is planned for the immediate future. 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. These models are only required when the robot pose uncertainty is computed and if odometry observations are to be fused to the estimate provided by the optimisation algorithm. Further work to examine whether the observation gate can be replaced with a robust kernel in order to adequately deal with outliers is also planned.

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 during the experiments reported. Using the current best estimate of the robot pose and its uncertainty in order to take appropriate action in such situations is also an avenue for further work.

## REFERENCES

- [1] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 376–382, June 1991.
- [2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, no. February, pp. 1322–1328, IEEE, 1999.
- [3] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229–241, June 2001.
- [4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proceedings of the National conference on Artificial Intelligence*, pp. 593–598, 2002.
- [5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, pp. 34–46, Feb. 2007.
- [6] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2657–2664, IEEE, May 2010.
- [7] S. Huang, Z. Wang, and G. Dissanayake, "Sparse Local Submap Joining Filter for Building Large-Scale Maps," *IEEE Transactions on Robotics*, vol. 24, pp. 1121–1130, Oct. 2008.
- [8] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Transactions on Robotics*, vol. 24, pp. 1365–1378, Dec. 2008.
- [9] B. P. Gerkey, "AMCL for ROS (<http://www.ros.org/wiki/amcl>)," 2012.
- [10] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *5th international joint conference on Artificial intelligence - Volume 2*, (San Francisco, CA, USA), pp. 659–663, Morgan Kaufmann Publishers Inc., 1977.
- [11] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.
- [12] P. Felzenszwalb and D. Huttenlocher, "Distance Transforms of Sampled Functions," tech. rep., University of Cornell, 2004.
- [13] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1696–1703, IEEE, June 2010.
- [14] M. W. Jones, J. A. Baerentzen, and M. Sramek, "3D distance fields: a survey of techniques and applications," *IEEE transactions on visualization and computer graphics*, vol. 12, pp. 581–599, Jan. 2006.
- [15] D. Hähnel, "Intel Research Lab (Seattle) Dataset," 2000.
- [16] J. Clarke, "Modelling uncertainty: A primer," *Tutorial of Department of Eng. Science*, 1998.
- [17] L. Moreno, J. M. Armingol, S. Garrido, A. de la Escalera, and M. A. Salichs, "A Genetic Algorithm for Mobile robot Localization using Ultrasonic sensors," *Journal of Intelligent and Robotic Systems*, vol. 34, no. 2, pp. 135–154, 2002.
- [18] C. Dogruer, A. Koku, and M. Dolen, "Global urban localization of an outdoor mobile robot with genetic algorithms," *European Robotics Symposium 2008*, 2008.
- [19] N. Kwok, D. Liu, and G. Dissanayake, "Evolutionary computing based mobile robot localization," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 857–868, 2006.
- [20] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, 2007.
- [21] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, pp. 1270–81, July 2008.
- [22] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849–865, 1988.
- [23] M. Rauter and 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*, pp. 68–75, IEEE, June 2012.
- [24] C. Stachniss, "MIT CSAIL Building Dataset," 2005.