Vector Distance Function Based Map Representation for Robot Localisation

Janindu Arukgoda, Ravindra Ranasinghe, Lakshitha Danthanarayana
and **Gamini Dissanayake**
University of Technology Sydney, Australia

Tomonari Furukawa
Virginia Polytechnic Institute and State University, VA, USA

Abstract

This paper introduces the use of the vector distance function (VDF) for representing environments, particularly for the use in localisation algorithms. It is shown that VDF has a continuous derivative at the object boundary in contrast to unsigned distance transform, and does not require an environment populated with closed object as in the case of the signed distance transforms, the two most common strategies reported in the literature for representing environments based on distances to nearest occupied regions. As such VDF overcomes the main disadvantages of the existing distance transform based representations in the context of robot localisation. The key properties of VDF are demonstrated and the use of VDF in robot localisation using an optimization based algorithm is illustrated using three examples. It is shown that the proposed environment representation and the localisation algorithm is effective in providing accurate location estimates as well as the associated uncertainties.

1 Introduction

Environment representation plays a key role in the effectiveness of robot localisation algorithms. When the environment is represented using a set of geometric features such as points or lines, an extended Kalman filter (EKF) can be formulated to estimate the robot pose using information gathered by a sensor such as a laser range finder mounted on the robot. Occupancy grid maps (OGM) are a popular alternative, typically in indoor environments. When an OGM is available, a particle filter [Thrun *et al.*, 2001] is the method of choice for robot localisation. One of the key components of robot localisation algorithms is the observation model that relates the sensor observations to environment geometry. In case of a sensor that observes the environment and returns a set of range and/or bearing measurements, EKFs rely on being able to obtain analytic expressions relating features in the environment and the robot pose. On the other hand, particle filters use ray casting on the occupancy grid to evaluate the likelihood of observations from a hypothesized location.

During the past few years, ability of distance function based methods for representing geometry have been recognized by the robotics community. Distance functions at a given location is the shortest distance to the nearest object. The measure used to represent this distance can be selected based on the particular application. Typical examples are the Euclidean, chessboard and city-block distance [Paglieroni, 1992]. Once a distance metric is defined, the misalignment between a template and an observation can be computed using a variety of metrics, the most widely used being Chamfer Distance [Barrow *et al.*, 1977] and Hausdroff Distance [Huttenlocher *et al.*, 1993].

Distance functions implicitly capture the geometry of the environment and have found use in a number of applications related to robotics. Truncated signed distance functions have been used for real-time dense surface tracking and mapping in KinectFusion[Newcombe *et al.*, 2011]. Unsigned distance functions have been used for ground robot localisation in an optimisation framework in C-LOG [Dantanarayana *et al.*, 2013].

The metric used to define the distance governs the characteristics of the map representation. Unsigned distance transform has discontinuous derivatives at the object boundary and the cut locus, the locus of points at which the distances to multiple nearby boundaries are the same. When used with a localisation algorithm, many of the observations falls near the object boundary as the location estimate approaches its true value. As uncertainty estimates rely on the gradients of the measurement equation, computing the uncertainty of the location estimate requires heuristics to avoid these discontinuities. Signed distance transform has a continuous derivative at the object boundary, and has found use in

3D point cloud registration as mentioned above [Newcombe *et al.*, 2011]. Unicombe et. al. [Unicombe *et al.*, 2017] use a signed distance function based map representation to estimate full 6-DOF pose of an Unmanned Aerial Vehicle (UAV) in an EKF framework, where the UAV flies above a flat ground with roadway marked as a ́figure of 8ː However, these scenarios are confined to environments that consist of closed regions. Although it is possible to represent environments such as 2D occupancy grid maps with signed distance transforms, this requires ray casting at each target location estimate to assign the correct sign, leading to a high computational cost. Furthermore, it has been demonstrated by Mullen et al. (2010) and Chazal et al. (2011) that in 3D surface reconstruction, unsigned distance functions are much more robust to noise and outliers than signed distance functions.

Another distance function that has been used in computer vision domain is the vector distance function (VDF)[Faugeras and Gomes, 2000; Abdelmunim *et al.*, 2013]. This paper explores the use of VDF for environment representation and examines how an optimization based technique can be used to localise the robot in the map. It is demonstrated that VDF representation leads to well behaved observation models.A new measure of similarity between the robot sensor measurement and the map that can be used in an optimisation framework for robot pose estimation is proposed. A method for computing the uncertainty of the estimated pose is also described.

This paper is organised as follows. Section 2 illustrates the characteristics of the vector distance functions through an example. Robot localisation is formulated as an optimisation problem and a method for computing the uncertainty of the solution this problem is also proposed in Section 2. Results of experiments conducted using multiple datasets are presented in Section 3 demonstrating the effectiveness of the proposed distance function and the localisation algorithm. Section 4 reviews the contributions of this paper and presents concluding remarks.

2 METHODOLOGY

2.1 Environment Representation

This section presents a study of the vector distance function for environmental representation.

Given an occupancy grid map, the unsigned distance function implicitly represents the distance to the closest occupied cell. The signed distance function produces the same, with a sign assigned to the resulting distance based on whether the grid cell is inside or outside an area enclosed by occupied grid cells. However, both these functions do not make use of an additional piece of information that can be obtained; the direction towards the closet occupied cell. For each grid cell, the vector distance function produces the distances to the closest occupied cell along the axes of the coordinate system, implicitly capturing both the magnitude and direction of the vector from the grid cell to the closest occupied cell. The vector distance function can be formulated as follows.

Let M be an occupancy grid map and DT_v denote the vector distance function of M. If $V = \{v_i\} \subseteq M$ is the set of occupied cells, then for any cell in $U = \{u_i\} \subseteq M$, function $DT_v : U \rightarrow R^2$ can be defined as

$$DT_v(u) = (i_u - i_{v^*}, j_u - j_{v^*}) \tag{1}$$

where (i_u, j_u) and (i_{v^*}, j_{v^*}) are the Cartesian coordinates of the cells u and its closest occupied cell v^*.

In order to derive DT_v in linear time, we use a computationally efficient algorithm proposed in [Maurer *et al.*, 2003]. The resulting two orthogonal Distance Transform matrices, denoted by DT_x and DT_y, can be stored separately to represent the map of the environments. It is possible to approximate DT_x, DT_y and their first and second derivatives using cubic splines, to obtain a representation that is amenable to be used with either an optimisation or extended Kalman filter based location estimation algorithm. While these are potentially expensive calculations, for a given map these can be precomputed and stored for later use.

Fig. 1 shows a part of the map used in simulation experiments detailed in Section 3 together with its representation using the unsigned distance transform matrix (DT) and the two distance transform matrices (DT_x and DT_y) of its vector distance function. While DT, DT_x and DT_y are continuous functions, it should be noted that the derivative of DT is not continuous both at the cut loci and at the edge points of the map while DT_x and DT_y as well as their derivatives are discontinuous only at the cut loci. Fig. 2 shows how the variation of DT, DT_x, DT_y and $DT_x^2 + DT_y^2$ along the lines parallel to X and Y axes of the grid map to further illustrate this.The key property that is exploited in the algorithm proposed in this paper is the fact that the vector distance function and its derivatives are unlikely to be required at cut loci during a localisation task.

2.2 Robot Localisation

The goal of the 2D robot localisation problem is to estimate the robot pose $\boldsymbol{x_r} = (x_r, y_r, \phi_r)^\top$ with respect to an a-priory map M using a collection of measurements S obtained by the sensors mounted on the robot such as laser range finders and cameras. Given an estimate for the robot pose $\boldsymbol{x_r}$, if the difference between the sensor measurement and the map can be quantified by some function $F(S, M)$, then the localisation problem can be

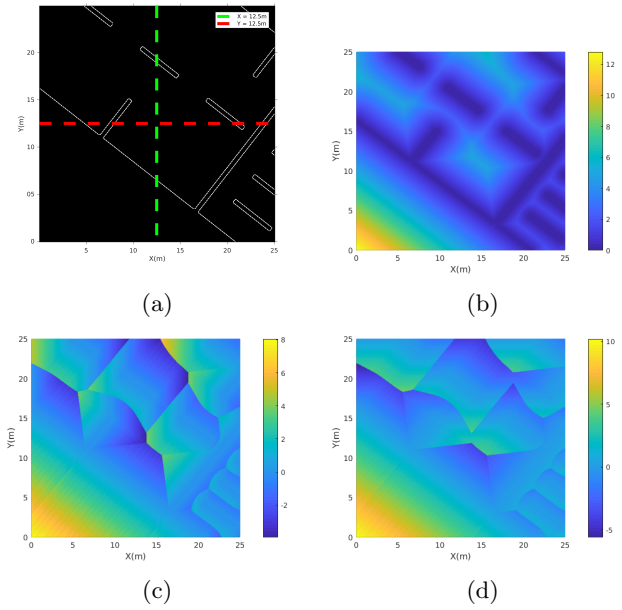

Figure 1: (a) Map (b) Unsigned DT (c) DT_x and (d) DTy for a part of map in simulation

formulated as an optimization problem [Dantanarayana *et al.*, 2013].

$$\hat{\boldsymbol{x_r}} = \underset{x_r, y_r, \phi_r}{\operatorname{argmin}} F(S, M) \tag{2}$$

The disparity between the sensor measurement formulated as a template image and the query image map can be calculated using some metric as discussed in Section 1. In this work, we propose to use a dissimilarity measure inspired by Chamfer Distance, denoted as (VCD). For a template image $U = \{u_i\}$ and a query image $V = \{v_j\}$, VCD is defined as (3).

$$VCD(U, V) = \sum_{u_i \in U} \min_{v_j \in V} |u_i - v_j|^2 \tag{3}$$

The euclidean distance between a pixel u and its nearest edge pixel v^* in an edge map V can be defined as (4).

$$\min_{v \in V} |u - v| = \sqrt{(i_u - i_{v^*})^2 + (j_u - j_{v^*})^2} \tag{4}$$

Since V is represented by a vector distance function, VCD can be easily calculated using its two orthogonal distance transform matrices as (5).

$$VCD(U, DT_v) = \sum_{u_i \in U} (DT_x(u_i)^2 + DT_y(u_i)^2) \tag{5}$$

Given a robot pose estimate $\boldsymbol{x_r} = (x_r, y_r, \phi_r)$ and a 2D laser scan measurement $S_{r\theta} = \{(r_i, \theta_i)\}$ of n range-bearing readings, the template edge image $U = \{X_{oi}\}$ can be formulated as (6).

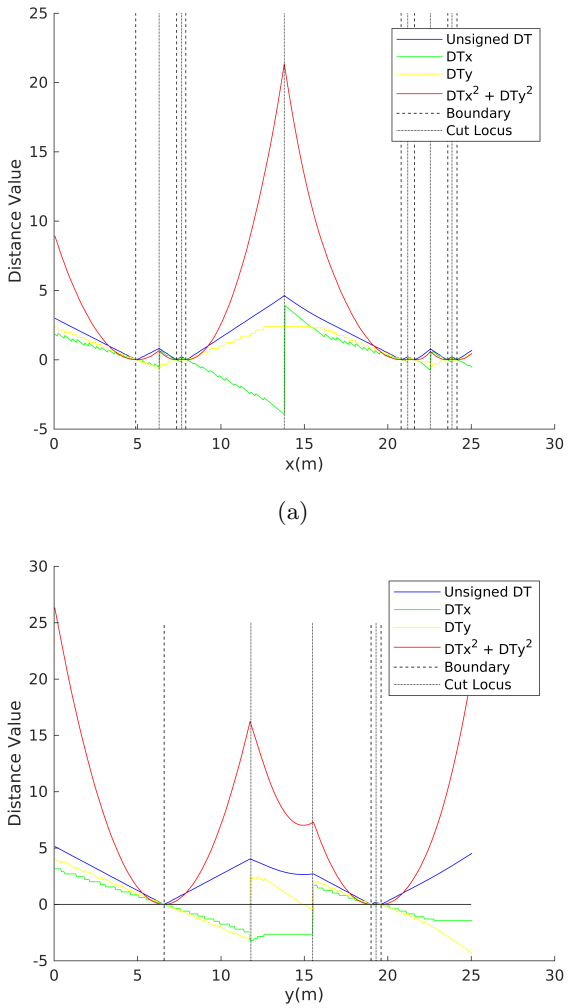

Figure 2: The unsigned DT, DT_x, DT_y and $DT_x^2 + DT_y^2$ along (a) X = 12.5m and along (b) Y = 12.5m for the map in Fig.1. Boundaries and one cut loci are marked in each figure to highlight the salient characteristics of these distance functions.

$$X_{oi} = \begin{Bmatrix} x_{oi} \\ y_{oi} \end{Bmatrix} = \begin{Bmatrix} x_r + r_i \sin(\theta - \phi) \\ y_r + r_i \cos(\theta - \phi) \end{Bmatrix} \quad (6)$$

VCD for a given pose estimate $\boldsymbol{x_r} = (x_r, y_r, \phi_r)$ can now be calculated using the vector distance function representation of the map $DT_v(M)$, and the observation vector X_o as (7).

$$VCD(X_o, DT_v) = \sum_{i=1}^{n} (DT_x(X_{oi})^2 + DT_y(X_{oi})^2) \quad (7)$$

Therefore, VCD can be used as the function F in (2) to estimate robot pose by solving the unconstrained nonlinear optimization problem (8).

$$\hat{\boldsymbol{x_r}} = \underset{x_r, y_r, \phi_r}{\operatorname{argmin}} VCD(X_o, DT_x, DT_y) \quad (8)$$

Algorithm 1 Localise Robot

Require: DT_x, DT_y, $\frac{\partial DT_x}{\partial x_{oi}}$, $\frac{\partial DT_x}{\partial y_{oi}}$, $\frac{\partial DT_y}{\partial x_{oi}}$, $\frac{\partial DT_y}{\partial y_{oi}}$, $\frac{\partial^2 DT_x}{\partial x_{oi}^2}$, $\frac{\partial^2 DT_y}{\partial y_{oi}^2}$, Initial guess for pose $X_1 = (x_1\ y_1\ \phi_1)^T$

loop for each input sensor reading = $S_{r\theta}$

function OPTIMISE(X_1)
return $X^* = (x\ y\ \phi)^T$ at $\min VCD$
end function
function UPADTEGUESS(X^*)
return *Return X^* updated using odometry*
end function
end loop

2.3 Uncertainty of the Pose Estimate

Since the estimate for the robot pose $\hat{\boldsymbol{x_r}}$ is obtained via an optimization process, an explicit function that maps the sensor measurements to a pose estimate is not available. Therefore, the uncertainty of the pose estimate $\text{cov}(\hat{\boldsymbol{x_r}})$ due to the noise in sensor readings $\text{cov}(S)$ can be estimated with the aid of the implicit function theorem as (9)[Clarke, 1998]. The Jacobian matrix J and the Hessian matrix H can be calculated by (10) and (11) respectively. When the environment is represented by an unsigned distance function, this approach cannot be directly used to calculate the uncertainty of the pose estimation. As the derivatives of the distance function cannot be evaluated at boundaries because of the discontinuities, and therefore some heuristics to approximate derivatives are required.

$$\text{cov}(\hat{\boldsymbol{x_r}}) = J * \text{cov}(S) * J^T \quad (9)$$

$$J = -H^{-1} * \begin{bmatrix} \frac{\partial^2 VCD}{\partial x_r \partial r} \\ \frac{\partial^2 VCD}{\partial y_r \partial r} \\ \frac{\partial^2 VCD}{\partial \phi_r \partial r} \end{bmatrix} \quad (10)$$

$$H = \begin{bmatrix} \frac{\partial^2 VCD}{\partial x_r^2} & \frac{\partial^2 VCD}{\partial x_r \partial y_r} & \frac{\partial^2 VCD}{\partial x_r \partial \phi_r} \\ \frac{\partial^2 VCD}{\partial x_r \partial y_r} & \frac{\partial^2 VCD}{\partial y_r^2} & \frac{\partial^2 VCD}{\partial y_r \partial \phi_r} \\ \frac{\partial^2 VCD}{\partial x_r \partial \phi_r} & \frac{\partial^2 VCD}{\partial y_r \partial \phi_r} & \frac{\partial^2 VCD}{\partial \phi_r^2} \end{bmatrix} \quad (11)$$

Each element in the two matrices can be expressed by DT_x, DT_y, $\frac{\partial DT_x}{\partial x_{oi}}$, $\frac{\partial DT_x}{\partial y_{oi}}$, $\frac{\partial DT_y}{\partial x_{oi}}$, $\frac{\partial DT_y}{\partial y_{oi}}$, $\frac{\partial^2 DT_x}{\partial x_{oi}^2}$, $\frac{\partial^2 DT_y}{\partial y_{oi}^2}$ and the first and second order derivatives of (6). As previously mentioned, the matrices DT_x, DT_y and their first and second order derivatives are precomputed and stored in memory. Therefore computing J consists of mainly table lookups.

3 EXPERIMENTAL RESULTS

The objective of this section is to evaluate VDF based localisation in different scenarios using simulation and real world data with ground and aerial robots. We use three datasets.

A) Dataset collected from Player/Stage Simulator: Ground Robot

B) Dataset of Intel Research Lab in Seattle: Ground robot

C) Dataset collected from a hexarotor UAV in Kentland farm, Blacksburg, Virginia, USA: Aerial robot

Experiments on the data were conducted in Matlab® R2017a environment on an Intel® Core i7-7700, 3.60GHz computer. Robot Operating System (ROS) Kinetic Kame distribution release was used as the middleware for data collection with the UAV.

The readers are referred to the work of Dantanarayana et al. (2013) and Unicomb et al. (2017) on evaluating unsigned and signed distance function based localisation with similar datasets.

3.1 Dataset collected from Player/Stage simulator

A dataset was collected from a mobile robot in Player/Stage simulator[Gerkey *et al.*, 2003] equipped with a Hokuyo laser range finder with a 30m range and a 270° field of view, and an odometer. Zero mean Gaussian noise was added to the laser range readings ($\sigma_r = 0.3m$) and the linear ($\sigma_v = 0.04ms^{-1}$) and the angular ($\sigma_\omega = 0.01rads^{-1}$) velocities, to simulate noisy measurements expected in practice. The mobile robot was navigated manually through an environment named 'Hospital' available with the simulator by default. A perfect map of the environment was available with the simulator.

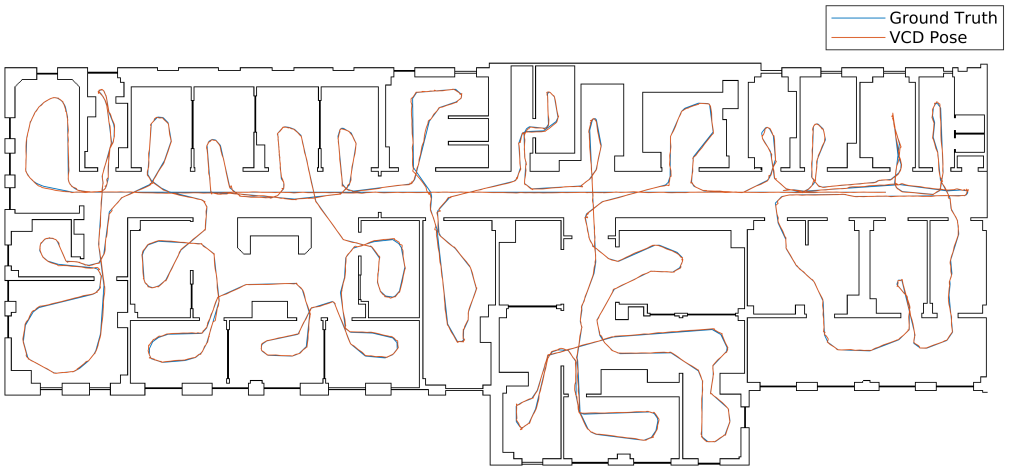

Figure 3: Ground Truth vs VCD Pose estimate for the simulation data set

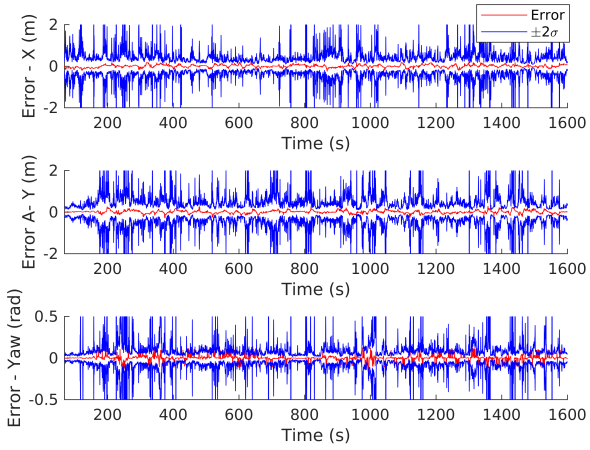

Figure 4: VCD Pose Error and associated $\pm 2\sigma$ Bounds

Fig. 3 shows the true path of the robot and the path generated from the VCD pose estimates. Fig. 4 shows the x, y and yaw errors and the corresponding uncertainties through the entire trajectory.

3.2 Intel Research Lab Date Set

The second dataset contains the odometry and 2D LiDar data collected by a mobile ground robot moving 3 loops inside the Intel Research Laboratory in Seattle [Hähnel, 2000]. First, for the data in loop 3 a map was built using GMapping[Grisetti *et al.*, 2007]. To protect the integrity of the localisation experiments, they were conducted using the data from the first 2 loops. Fig. 5 shows the GMapping pose estimation alongside the VCD pose estimation for the first 2 loops of the mobile robot's journey.

As ground truth is not available for this dataset, the accuracy of the computed uncertainties can not be rigorously evaluated.

3.3 Dataset collected from a hexarotor UAV in Kentland Farm

The third datasets was used to evaluate the proposed localisation approach on a hexarotor UAV. It was collected in Kentland Farm, Blacksburg, VA, USA where the team from Virginia Tech (Team VICTOR) conducted flight tests in preparation for Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017, an international robotics competition involving unmanned ground and aerial vehicles. In tandem with the competition specification, a roadway in the shape of a figure of 8 was taped on the runaway.

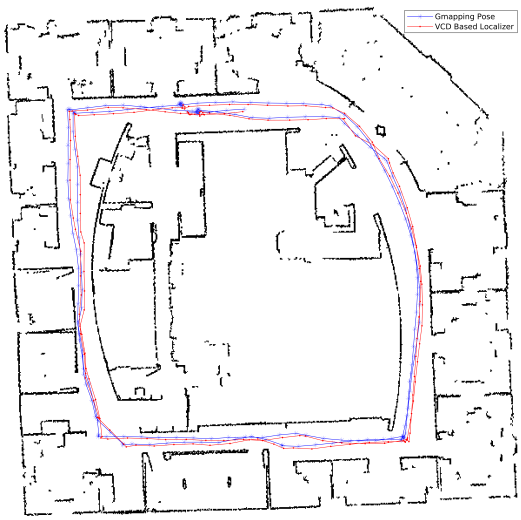

Figure 5: GMapping Pose vs VCD Pose in Intel Lab Dataset

The hexarotor UAV setup shown in Fig. 6 was custom built for the purpose of the competition. It is equipped with a perspective camera, a fisheye camera and a laser range finder which are rigidly fixed to the UAV body pointing downwards, a Real-time Kinematics Global Positioning System (RTK-GPS) module and a Pixhawk® flight controller with an internal inertial measurement unit (IMU). In this environment, the RTK-GPS module provides highly accurate location estimates, which can be used as the ground truth of the UAV pose.

The state estimator in the Pixhawk® flight controller uses its internal IMU to estimate its roll and pitch angles. These estimates are adequately accurate for high frequency UAV control and are available for high level navigation tasks such as path planning and way point following. Furthermore, the high precision laser range finder fixed to the UAV body pointing downwards can then be used to estimate the operating altitude of the UAV. Given these conditions, we can reduce the localisation problem from estimating $\boldsymbol{x_r} = (x_r, y_r, z_r, \phi_r, \theta_r, \psi_r)^\top$ to estimating $\boldsymbol{x_r} = (x_r, y_r, \phi_r)^\top$.

To build a map of the environment, the UAV followed a set of waypoints over the area and images were captured using the perspective camera. The captured images were corrected for lens distortion stitched together. The stitched image was converted to a binary image by extracting the edges. The vector distance function for these binary edge images were then calculated.

For localisation, we use the perspective camera. Images were captured in synchronization with the current UAV pose estimation $\boldsymbol{x_r} = (x_r, y_r, \phi_r)^\top$, the pitch and yaw angles from the IMU and the height information from the laser range finder. The images were corrected for lens distortion and the edges were extracted from it. The set of edge points (λ_i, μ_i) were then transformed from the image plane to the ground plane using the assumption that the ground is a flat terrain using (12). R is the rotation matrix explaining the attitude of the UAV and f is the focal length of the camera obtained via calibration.

Figure 6: Bogey5 : One of the UAVs custom built by Team VICTOR for MBZIRC 2017

Fig. 7 shows a comparison between the poses produced by the RTK-GPS and VCD localiser. It can be seen that the localiser fails when the UAV is not on top of the runway. This is to be expected because the system fails to detect and extract features that can be used for localisation. The trajectory of the UAV estimated by the VCD localiser when it detects a sufficient number of features is drawn for visual comparison with the UAV trajectory obtained by RTK-GPS measurements. Fig. 8 shows shows the x, y and yaw errors and the corresponding uncertainties at each step when the localiser is fed with sufficient number of features.

$$\mathbf{x}_{o_i} = \begin{Bmatrix} x_{o_i} \\ y_{o_i} \end{Bmatrix} = \begin{Bmatrix} x_r + z\frac{\lambda_i R_{1,1} + \mu_i R_{1,2} - f R_{1,3}}{\lambda_i R_{3,1} - \mu_i R_{3,2} + f R_{3,3}} \\ y_r + z\frac{\lambda_i R_{2,1} + \mu_i R_{2,2} - f R_{2,3}}{\lambda_i R_{3,1} - \mu_i R_{3,2} + f R_{3,3}} \end{Bmatrix} \quad (12)$$

4 DISCUSSION & CONCLUSIONS

The main contribution of this work is the use of vector distance functions to represent the environment for robot localisation. The vector distance function representation holds advantages over signed and unsigned variations of the distance functions where the ability to represent open curves and the continuity of derivatives at the map boundaries allow it to represent any environment and to calculate the uncertainty of the pose

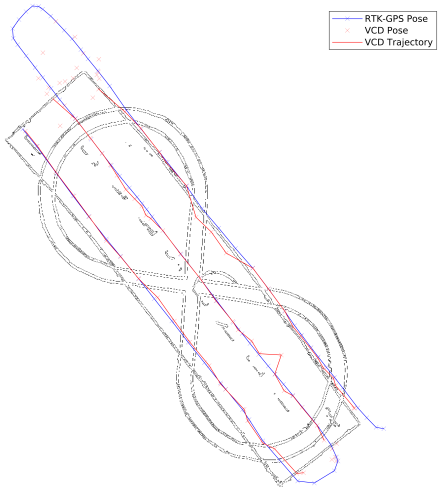

Figure 7: RTK GPS pose vs VCD pose in Dataset C

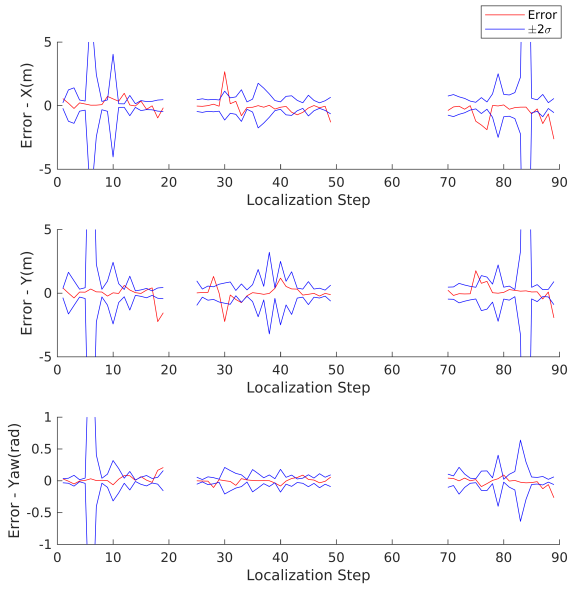

Figure 8: VCD Pose Error and $\pm 2\sigma$ Error Boundaries at each successful localiser step.

estimate. The robot localisation problem is formulated as a minimization of a metric that describes the difference between the sensor measurement and the map. It is demonstrated through simulation and field experiments that the proposed approach can be used to estimate pose and associated uncertainty of ground robots equipped with laser range finders and an aerial robot equipped with a monocular camera.

Even in the single threaded MATLAB implementation, the optimization in average takes less than 25ms. The calculation of uncertainty of pose estimation takes 135ms. However, as all matrix lookups and multiplications can be heavily parallelized and optimized, it is expected that an optimized implementation in C or C++ will be capable of real-time performance on ground and aerial robots.

Future work will involve further experiments on additional datasets particularly including 3D sensors such as RGB-D cameras, evaluating the effectiveness of a continuous representation such as a Gaussian Process to directly capture the VDF in order to enhance the localisation accuracy and the feasibility of extending this work to implement a localisation algorithm based on an extended Kalman filter to further reduce the computational complexity.

ACKNOWLEDGEMENT

This work was funded by the Centre for Autonomous Systems, University of Technology Sydney, Australia and Khalifa University, UAE. The authors would like to acknowledge the members of Team VICTOR who participated in MBZIRC 2017 for their contributions in hardware and software development, and data collection.

References

[Abdelmunim *et al.*, 2013] H. Abdelmunim, A. Farag, and A. A. Farag. Shape representation and registration in vector implicit spaces: Adopting a closed-form solution in the optimization process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):763–768, March 2013.

[Barrow *et al.*, 1977] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical report, DTIC Document, 1977.

[Chazal *et al.*, 2011] Frdric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric Inference for Probability Measures. *Foundations of Computational Mathematics*, 11(6):733–751, 7 2011.

[Clarke, 1998] John C Clarke. Modelling uncertainty: A primer. *Tutorial of Department of Eng. Science*, pages 1–21, 1998.

[Dantanarayana *et al.*, 2013] Lakshitha Dantanarayana, Ravindra Ranasinghe, and Gamini Dissanayake. C-log: A chamfer distance based method for localisation in occupancy grid-maps. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 376–381. IEEE, 2013.

[Faugeras and Gomes, 2000] Olivier D Faugeras and José Gomes. Dynamic shapes of arbitrary dimension: The vector distance functions. In *IMA Conference on the Mathematics of Surfaces*, pages 227–262, 2000.

[Gerkey *et al.*, 2003] Brian Gerkey, Richard T Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, volume 1, pages 317–323, 2003.

[Grisetti *et al.*, 2007] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46, 2007.

[Hähnel, 2000] D Hähnel. Intel research lab (seattle) dataset. 2000.

[Huttenlocher *et al.*, 1993] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993.

[Maurer *et al.*, 2003] C. R. Maurer, Rensheng Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, Feb 2003.

[Mullen *et al.*, 2010] Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets. *Computer Graphics Forum*, 29(5):1733–1741, 9 2010.

[Newcombe *et al.*, 2011] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011.

[Paglieroni, 1992] David W Paglieroni. A unified distance transform algorithm and architecture. *Machine Vision and Applications*, 5(1):47–55, 1992.

[Thrun *et al.*, 2001] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2):99–141, 2001.

[Unicombe *et al.*, 2017] James Unicombe, Lakshitha Dantanarayana, Janindu Arukgoda, Ravindra Ranasinghe, Gamini Dissanayake, and Tomonari Furukawa. Distance function based 6dof localization for unmanned aerial vehicles in gps denied environments. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017.