LOCALIZATION OF A DRIFTING UNDERWATER VEHICLE USING A TERRAIN-BASED PARTICLE FILTER

Emanuele Raggi
University of Rhode Island, eraggi@uri.edu

Follow this and additional works at: https://digitalcommons.uri.edu/theses

Recommended Citation
Raggi, Emanuele, "LOCALIZATION OF A DRIFTING UNDERWATER VEHICLE USING A TERRAIN-BASED PARTICLE FILTER" (2019). Open Access Master's Theses. Paper 1526.
https://digitalcommons.uri.edu/theses/1526

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master’s Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons@etal.uri.edu.
LOCALIZATION OF A DRIFTING UNDERWATER VEHICLE USING A TERRAIN-BASED PARTICLE FILTER

BY

EMANUELE RAGGI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN

OCEAN ENGINEERING

UNIVERSITY OF RHODE ISLAND

2019
MASTER OF SCIENCE THESIS
OF
EMANUELE RAGGI

APPROVED:

Thesis Committee:

Major Professor       Chris Roman
                      Stephen Licht
                      Paolo Stegagno
                      Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND
2019

Electronic copy available at: https://ssrn.com/abstract=3464971
ABSTRACT

In this paper we present a terrain-aided particle filter to localize a freely drifting underwater vehicle. The vehicle is a bottom imaging Lagrangian float used for habitat classification, monitoring and fish abundance. During operation the vehicle captures down looking images at a controlled altitude above the bottom. Direct navigation information is often, but not always, recorded with a ultra short baseline (USBL) acoustic system. The presented methodology provides an alternate method for georeferencing when USBL is unavailable. The implemented particle filter utilizes a background bathymetry map and visual odometry as a motion mode. The particle filter is implemented using the Robot Operating System (ROS) and Orocos Bayesian Filtering Library (BFL). The Grid Map package is used to store and retrieve the bathymetric data. Results using data collected on field deployments in the Hawaiian islands in 2018 show the method is able to effectively utilize the terrain information and produce drift trajectories which closely match the recorded USBL data. Utilizing the method allows the float system to be deployed with minimal ship-side support while still maintaining the georeferencing critical to the end use of the collected images.
ACKNOWLEDGMENTS

I would like to thank my advisor Christopher Roman, the Committee members Stephen Licht and Paolo Stegagno each of whom has provided patient advice and guidance throughout the research process. Last but not least, my lab mates Dave Casagrande and Kristopher Krasnosky for providing useful insights and help.
PREFACE

This thesis is presented in manuscript form.
# TABLE OF CONTENTS

ABSTRACT ................................................................. ii

ACKNOWLEDGMENTS ................................................... iii

PREFACE ................................................................. iv

TABLE OF CONTENTS .................................................... v

LIST OF FIGURES ....................................................... vii

LIST OF TABLES ........................................................ ix

MANUSCRIPT

1 Localization Of a Drifting Underwater Vehicle Using a Terrain-Based Particle Filter ........................................... 1

1.1 Introduction ....................................................... 2

1.2 Lagrangian Float System .......................................... 3

1.2.1 Sensor measurements ......................................... 4

1.3 Particle filter implementation .................................... 8

1.3.1 Motion models ................................................ 9

1.3.2 Measurement functions ....................................... 9

1.4 Results ............................................................ 12

1.4.1 EKF results ..................................................... 12

1.4.2 Particle filter results .......................................... 14

1.5 Discussion ......................................................... 19

1.6 Conclusions ......................................................... 21

APPENDIX

Electronic copy available at: https://ssrn.com/abstract=3464971
| Page |
|------|
| A. Robot Operating System implementation .................. 27 |
| BIBLIOGRAPHY ................................................. 30 |
# LIST OF FIGURES

| Figure | Description                                                                 | Page |
|--------|-----------------------------------------------------------------------------|------|
| 1      | Labelled diagram of the URI Lagrangian imaging float. The instrument is 1.3 meters tall and weighs approximately 25 kilograms. | 3    |
| 2      | An example depth plot for a typical float deployment.                       | 4    |
| 3      | A drift trajectory navigated with USBL and surface GPS positions. In this instance the USBL was lost several times, with one outage lasting nearly 30 minues. | 6    |
| 4      | Example image sequence with the individual displacements labelled.          | 7    |
| 5      | Flow charts of the ROS implementations of the particle filter and EKF.     | 9    |
| 6      | Bathymetry map for float the operating area.                               | 11   |
| 7      | Error histogram between the float measured depth and background bathymetry at USBL locations. | 12   |
| 8      | EKF trajectory using USBL measurements and visual odometry.               | 13   |
| 9      | Position uncertainty for the EKF using both USBL and visual odometry measuremants. | 14   |
| 10     | A sample trajectory estimated using only visual odometry.                 | 15   |
| 11     | XY USBL measurements (red dots) and estimated particle filter trajectory (blue) using on the terrain information in the particle filter. | 16   |
| 12     | Sample representation of the particle cloud using terrain navigation only. | 17   |
| 13     | XY USBL measurements (red dots) and estimated particle filter trajectory (blue) using visual odometry and terrain navigation. | 18   |
| Figure | Page |
|--------|------|
| 14     | Image of sample particle distributions for the terrain and visual odometry filter along the trajectory shown over the background bathymetry. The shape of these distributions reflects the shape of the underlying bathymetry. | 19 |
| 15     | XY USBL measurements (red dots) and estimated particle filter trajectory (blue) using visual odometry and USBL positions. | 20 |
| 16     | XY USBL measurements (red dots) and estimated particle filter trajectory (blue) using visual odometry, terrain and USBL navigation. | 21 |
| 17     | Particle spread estimated for each particle filter case shown with the EKF covariance for comparison. | 22 |
| A.1    | ROS Robot Localization flowchart for the EKF. | 28 |
| A.2    | ROS Robot Localization flowchart for particle filter. | 29 |
# LIST OF TABLES

| Table | Page |
|-------|------|
| 1     | List of parameter values used in the particle filter and EKF | 13 |
MANUSCRIPT 1

Localization Of a Drifting Underwater Vehicle Using a Terrain-Based Particle Filter

by

Emanuele Raggi\textsuperscript{1}, Chris Roman\textsuperscript{2}, Kris Krasnosky\textsuperscript{2}, David Casagrande\textsuperscript{2}, Paolo Stagagno\textsuperscript{3},

\textit{In preparation for submission to the Journal of Field Robotics}

\textsuperscript{1} University of Rhode Island, Department of Ocean Engineering  
\textsuperscript{2} University of Rhode Island, Graduate School of Oceanography  
\textsuperscript{3} University of Rhode Island, Department of Electrical Engineering
1.1 Introduction

There are common Gaussian-based filtering methods that can estimate the location of a vehicle given a set of continuous navigation measurements, such as depth, velocity, position and attitude [1] [7] [11]. The common implementation is the Kalman Filter (KF), which uses linear approximations of the state and measurement models to generate a state estimate given a prior state and associated uncertainties. The Extended Kalman Filter (EKF) can be used in cases where the linear approximation needs to be improved by re-linearizing around the current state estimate at each time step. These methods can work for unimodal error distributions but do not lend themselves well to the scope of this problem, where the velocity information is noisy and external position measurements are infrequent. Kalman filters are known to diverge when the initial state estimate is far from the actual solution or the actual errors violate the unimodal assumption.

A better approach for this application will be a particle filter that does not need to assume a unimodal error distribution and is very amenable to terrain added navigation problems. Particle filters can be a better approach for treating non-linear problems [5] [6]. Several applications of particle filters are described that tackle non-linearities [4] [9]. Particle representations can be implemented quite easily to approximate a wide array of distributions, but can require a large number of particles needed to achieve a desired accuracy. This can be computationally demanding [12].

The remainder of this paper describes the development of a particle filter approach appropriate for a drifting underwater vehicle. Section 1.2 describes the float instrument in more detail and discusses the sensors available to the filter. Section 1.3 describes the extended Kalman filter and particle filter implementations. Results for several variants of the particle filter are presented in Section 1.4. Section
1.5 and 1.6 discuss the results further in the context of typical float operations.

### 1.2 Lagrangian Float System

The Lagrangian imaging float, Figure 1, is a small drifting instrument designed to image the seafloor with minimal cost and operational complexity [8]. The float is able to drift over varying bathymetry (down to 100 meters depth) while automatically maintaining a constant altitude above the bottom [10]. While drifting it collects six megapixel down-looking stereo images 1-4 meters above the bottom at fixed (typically 2-3 secs) or variable rates. Images from the Lagrangian float have been used for a variety of applications. Single images, image strip mosaics, stereo-based 3D information (roughness, rugosity), and structure for motion (SFM) terrain estimates have been used for habitat documentation at Scott Reef, Australia, the Au Au Channel off Maui, Hawaii and the Cordell Bank National Marine Sanctuary, California.

![Figure 1: Labelled diagram of the URI Lagrangian imaging float. The instrument is 1.3 meters tall and weighs approximately 25 kilograms.](image)

During operations the float can be deployed for short (typically 20 minute)
dives in a repeating pattern or longer drifts lasting several hours, Figure 2. These dives are often completed to gain a quick assessment of the bottom and collect information prior to using more expensive assets or an intensive sampling effort. The floats movement is passive and subject to the direction and speed of the local current. The deployments can however be planned to take advantage of persistent currents and provide good spatial coverage. To reduce the operational cost it is advantageous to minimize the peripheral equipment and ship support. Often the float is deployed from a small (< 10 meter) open boat during day operations.

![Float depth profile](image)

Figure 2: An example depth plot for a typical float deployment.

1.2.1 Sensor measurements

The particle filtering method uses measurements of depth, altitude, drift speed over the bottom and direct position measurements provided by acoustic tracking. These measurements occur asynchronously and are incorporated into the filter as they occur in a causal manner.
Depth and altitude

The water depth is determined as a combination of the float’s depth and altitude above the seafloor. Depth is obtained from a Keller American PA33X pressure sensor with a full scale range of 11 dbar. The sensor has a resolution of 2 millimeters, an accuracy of 5 centimeters and reports data at 5 Hz. The altimeter is a 200 kHz Airmar transducer that provides data with 3 centimeter resolution at 1 Hz.

Position measurements

Direct underwater position measurements are made with an Evologics S2C R 18/34 Ultra Short Baseline (USBL) tracking system with a horizontally omnidirectional transducer beam pattern. These measurements are used to estimate the X and Y position for the float when underwater. The depth estimate from the USBL is generally noise and is not used in the filtering. The USBL system is range limited and can often be occluded by the bottom terrain or ship orientation. As such, it is common for the USBL data to drop out for minutes at a time, or simply be unavailable for some deployments when the float is used far from support vessel.

Additional information is collected when the float is on the surface and with a separate surface buoy. The float GPS and Iridium locating system will log the GPS at the start and end of the dive when it is still on the surface. During shallow operations, in less than 40 meters water depth, a surface buoy can be attached to the float with a thin floating tether. A GPS tracker on this buoy will give an approximate location of the float, but will be separated from the float by up to a tether length. Typically the surface buoy is pulled thought the water with a relatively consistent tether offset determined by the wind, current and depth of the float.

Figure 3 below shows an example of the three positioning measurements for
a sample dive.

![Figure 3: A drift trajectory navigated with USBL and surface GPS positions. In this instance the USBL was lost several times, with one outage lasting nearly 30 minutes.](image)

**Visual odometry**

The camera system mounted on the Float takes down-looking overlapping images. The images are processed to produce displacement estimates between sequential image pairs. The camera images are strobe illuminated and taken at a fixed frame rate of 3 seconds per pair and typically overlap. In some instances the overlap will be too small and a displacement measurement will not be possible. The images from each dive are batch processed as an initial step to produce a time series of velocity and displacement estimates. A sample sequence is shown in
The current odometry method uses standard single viewpoint camera calibrations [3] and a FFT-based approach that has been robust to images of varying quality taken over a range of altitudes [2]. The FFT method has proven to be robust to lower quality images that sometimes suffer from low contrast and backscatter caused by the background ambient lighting during daytime operations and water turbidity. The camera system also has a three axis magnetometer to measure heading and rotate the image into the world reference frame as a preprocessing step.

Prior to use by the particle filter odometry time series is first smoothed to obtain an average velocity. This process is done to compensate for the unsteady
surge motion of the float as it drifts near the bottom. The surge motion has motion period consistent with the surface wave period, as is typically less than 15 seconds. A filter time constant of 120 seconds produces more steady motion estimates that integrate well and avoid the surge motion.

1.3 Particle filter implementation

The filter presented here has been implemented using the Orocos Bayesian Filter Library (BFL) within the Robot Operating System (ROS). The overall organization of the method is shown in Figure 5a. The data for each float dive are stored in a rosbag. Replaying the rosbag controls the processing rate. On a typical laptop computer the method can run at 10x speed. The data are routed through an uncertainty pre-processor which adds uncertainty values to the ROS messages that will be used by the particle filter. The data messages are handled within the filter by measurement functions that run asynchronously. Each measurement function returns a likelihood of the measurement that is used by the filter to re-sample particles. The bathymetric data is handled independent of the rosbag and stored using the GridMap utility. GridMap is used to provide a depth lookups for the particle filter in the terrain resampling step. The particles are displayed using RViz after they have been transformed to the world coordinate frame.

The particle filter has been tested against a conventional extended Kalman filter (EKF), that was also implemented in ROS (Figure 5b). This filter uses the robot localization ROS package and does not incorporate the bathymetry data.

The particle filter is implemented as a two dimensional filter estimating the X and Y position of each particle. The state of each particle, \( p_k = [x, y, v_x, v_y] \), includes the location in a local coordinate frame and respective velocities. The initial distribution of particles is specified by an initial location and a Gaussian distribution parametrized by \( \sigma_p \). The depth of particles are assigned the most
Figure 5: Flow charts of the ROS implementations of the particle filter and EKF.

recent depth measurement from the pressure sensor. The origin of the local co-
ordinate frame is set from the first valid USBL measurement during the float dive.
Similarly, the EKF uses a two dimensional state vector.

1.3.1 Motion models

The particle filter assumes that the motion of the float at any time is governed
by a random walk. This is implemented by perturbing the velocity of each particle
by a random sample drawn from a normal distribution. This distribution is zero
mean with variance $\sigma_v$. This motion step occurs every second.

The prediction step of the EKF implementation is parameterized by a process
noise $Q_v$ on the state velocity.

1.3.2 Measurement functions

The measurement functions for the particle filter are defined for the USBL
position, visual odometry and measured bathymetry. Each function is implemented
to return a likelihood of the specific measurement given the state of each particle.
These functions have been implemented using Gaussian likelihoods.

**USBL**

The USBL position measurement function is characterized as zero mean with constants $\sigma_x$ and $\sigma_y$. This level of uncertainty matches the spread of measurements from calibration tests where the float was fixed on the seafloor and the ship made measurements from different bearings and ranges.

**Bathymetry**

The terrain likelihood function is implemented as a lookup for each particle using the background bathymetry data stored in GridMap (Figure 6). GridMap loads the data once at the start of the algorithm and the subsequent access is computationally efficient. For the cases tested here the map was stored with 10x10 meter grid cells. The terrain measurements are run a fixed rate, typically 10 seconds. Although the altimeter reports more frequently the slow motion of the float over the bottom, the grid cell size do not require a faster update rate. This rate could be changed if a smaller or larger grid size is used.

The likelihood of a terrain measurement is defined as a Gaussian parameterized by $\sigma_t$. To derive a reasonable value for $\sigma_t$ the distribution of errors between the float’s water depth estimate and the bathymetry map water depth at 3000 USBL positions across several dives was determined (Figure 7). This distribution captures the spread of the best case error that can be expected between the float and the background bathymetry. This distribution is not exactly zero mean, which is likely attributed some non-tidal constant offset. The incorporated data cover a range of times and tidal levels.
Visual odometry

The visual odometry measurement function is defined as a velocity error between an individual particle’s velocity and the measured odometry velocity. This is defined as a Gaussian likelihood parameterized by $\sigma_o(t)$. The time dependence is included because uncertainty values are calculated for each image measurement. This value is determined from the shape of the image matching correlation peak and an altitude scale factor that relates pixel values true scale motion.
1.4 Results

To evaluate the filtering method the EKF and several version of the particle filter were run. This specific float deployment was chosen from a number of possible dives because it has a section of good USBL data, a long USBL hiatus, varied bathymetry and odometry data with typical image overlap and surge motions.

The specific parameters used for the initial conditions, measurement uncertainties, process noise and are shown in Table 1.

1.4.1 EKF results

The EKF was run to establish a best case performance baseline when using both USBL and visual odometry. A comparison case using only visual odometry was also run to provide insight for operational scenarios where continuous USBL
Table 1: List of parameter values used in the particle filter and EKF

measurements are not be feasible. The drift trajectory in Figure 8 shows a very good correspondence with the USBL measurements when they are available. During the 30 minute gap in USBL data the trajectory follows the direction correctly but over estimates the motion. When the USBL returns the trajectory is pulled back to the USBL measurements.

Figure 8: EKF trajectory using USBL measurements and visual odometry.

The corresponding position uncertainty is shown in Figure 9. This clearly shows the loss of confidence during the USBL gap and the smaller periodic increases

Electronic copy available at: https://ssrn.com/abstract=3464971
in uncertainty between the regular USBL fixes.

![Figure 9: Position uncertainty for the EKF using both USBL and visual odometry measurements.](image)

Figure 9: Position uncertainty for the EKF using both USBL and visual odometry measurements.

Figure 10 shows a position track created by integrating only the image displacements over time. In general this trajectory replicates the USBL tracking in direction but is prone to the inherent integration error over time and steadily deviates from the starting point.

### 1.4.2 Particle filter results

The particle filter was evaluated using several combinations of the USBL, terrain and visual odometry measurements. In each instance 5000 particles were used and the initial particle distribution was started at the location of the first USBL measurement. The trajectory of the particle filter was defined by the centroid of the particle cloud at each time step. The spread of the particle distribution was characterized by calculating the standard deviation of the distances between the centroid and the individual particles. This radial spread calculation does not cap-
Figure 10: A sample trajectory estimated using only visual odometry.

ture the full shape of the particle distribution but does provide a single statistic to represent the overall consensus of the particle cloud along the trajectory.

**Terrain only measurements**

The trajectory using only terrain measurements in the particle filter is shown in Figure 11. This trajectory has quite a bit a variability but matches the general motion trend measured by the USBL. In this case the only particle motion in the filter is the non directional random walk.

Figure 12 shows a sample particle cloud along the trajectory that follows the terrain shape.
Figure 11: XY USBL measurements (red dots) and estimated particle filter trajectory (blue) using on the terrain information in the particle filter.

**Terrain and visual odometry**

Figure 13 shows the trajectory when both terrain and visual odometry are included in the filter. This case produces a much more consistent trajectory that is closer to the USBL positions and more direct that in the terrain only case. This shows the value of the odometry data to constrain motion of the float.

**Visual odometry and USBL**

Figure 15 shows the combination of odometry and direct USBL measurements. In this case the filter does a better job of spanning the USBL gap than that EKF and generally follows the USBL fixes. Near the center of the trajectory the trajectory is pulled east of the USBL fixes. This could indicate a directional bias in the odometry data.
Figure 12: Sample representation of the particle cloud using terrain navigation only.

**Terrain, USBL and visual odometry**

Figure 16 shows the most constrained result where the filter uses all three measurements. In this case the float follows the USBL fixes where they are available. In the USBL gap the filter keeps the trajectory consistent with the USBL points with only a small correction when the USBL measurements return.
Summary of uncertainty

A comparison of the uncertainties for the EKF and all four particle filter cases is shown in Figure 17. The EKF results, reproduced here from Figure 8, shows the best case scenario when USBL measurements are available and a rapid growth in uncertainty during the USBL gap. The terrain only particle filter generally has higher uncertainty along the trajectory. This is expanded as it is never constrained by the direct position fix. The USBL and visual odometry particle filter behaves similarly to the EKF. In this case the uncertainty is generally low but increases during the USBL gap. The growth rate is lower than the EKF. The terrain and visual odometry particle filter has generally low uncertainty that is not significantly higher than the version using USBL information. Lastly, the filter using all three measurements has the lowest overall uncertainty and limited growth in uncertainty.
Figure 14: Image of sample particle distributions for the terrain and visual odometry filter along the trajectory shown over the background bathymetry. The shape of these distributions reflects the shape of the underlying bathymetry.

across the USBL gap. This suggests the additional terrain information effectively curbs the error growth associated with just the visual odometry.

1.5 Discussion

The particle filter test cases presented here capture the behavior of the filter when including different measurement sources. The combination of terrain measurements and visual odometry performed well relative to the EKF and particle filters incorporating direct USBL measurements. Some potential biases were ob-
Figure 15: XY USBL measurements (red dots) and estimated particle filter trajectory (blue) using visual odometry and USBL positions.

Served between the visual odometry and the USBL measurements. This could indicate a camera heading calibration offset. The terrain-only filter performed reasonably but its performance could likely be improved by using tighter constraints on the bathymetry measurements and a better treatment of any potential offsets. The non-zero mean of the error histogram shown in Figure 7 indicates that a bias does exist and could be affecting the results. The next step should be a better tidal analysis to separate any time varying and static offsets. The majority of the testing for these results were done on this one sample dive. To further evaluate the parameter choices additional dives should be processed to see if the parameters still provide good results. Lastly, these filters did not include the additional surface GPS measurements available during regular operations. These could also be incorporated into the filter to provide additional constraints at the start and
Figure 16: XY USBL measurements (red dots) and estimated particle filter trajectory (blue) using visual odometry, terrain and USBL navigation end of the dives.

1.6 Conclusions

This thesis has developed a terrain-based particle filter method for navigating an underwater camera system. The results indicate that the terrain information can effectively be used to estimate the drift track of the float. The combination of visual odometry and terrain information proves the most effective at producing a trajectory estimate close to the direct USBL position measurements. This result is encouraging as it provides an alternative approach to localizing the float images when USBL is not available. This method will help reduce the operating costs for the float by reducing the need for peripheral tracking and ship support. Future improvements to the approach could entail better characterization of the depth errors and the inclusion of the surface GPS and tracking buoy data.
Figure 17: Particle spread estimated for each particle filter case shown with the EKF covariance for comparison.
LIST OF REFERENCES

[1] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lake- 
meyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The museum tour-
guide robot rhino. In Heinz Wörm, Rüdiger Dillmann, and Dominik Henrich, 
editors, Autonome Mobile Systeme 1998, pages 245–254, Berlin, Heidelberg, 
1999. Springer Berlin Heidelberg.

[2] David Casagrande. real-time featureless visual odometry for sea floor imaging 
with a lagrangian float. Master’s thesis.

[3] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. 
Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[4] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic 
systems. Journal of the American Statistical Association, 93(443):1032–1044, 
1998.

[5] Imbaby Mahmoud, May Salama, and Asmaa Tawab. Particle / kalman filter 
for efficient robot localization. International Journal of Computer Applications, 
106:20–27, 11 2014.

[6] Francisco Martín. Visual localization based on quadtrees. In Luís Paulo Reis, 
António Paulo Moreira, Pedro U. Lima, Luis Montano, and Victor Muñoz-
Martínez, editors, Robot 2015: Second Iberian Robotics Conference, pages 
599–610, Cham, 2016. Springer International Publishing.

[7] Elias Mueggler, Matthias Faessler, Flavio Fontana, and Davide Scaramuzza. 
Aerial-guided navigation of a ground robot among movable obstacles. 10 2014.

[8] Chris Roman, Gabrielle Inglis, and Bryan McGilvray. Lagrangian floats as 
sea floor imaging platforms. Continental Shelf Research, 31(15):1592 – 1598, 
2011.

[9] Thrun Sebastian, Fox Dieter, Burgard Wolfram, and Dellaert Frank. Robust 
monte carlo localization for mobile robots. Artificial Intelligence, 128(1):99 – 141, 2001.

[10] William Snyder, Chris Roman, and Stephen Licht. Hybrid actuation with 
complementary allocation for depth control of a lagrangian sea-floor imaging 
platform: Snyder et al. Journal of Field Robotics, 35, 05 2017.

[11] P. Stegagno, M. Cognetti, L. Rosa, P. Peliti, and G. Oriolo. Relative localization 
and identification in a heterogeneous multi-robot system. In 2013 IEEE 
International Conference on Robotics and Automation, pages 1857–1864, May 
2013.
[12] Sebastian Thrun. Toward robotic cars. *Commun. ACM*, 53(4):99–106, April 2010.
LIST OF REFERENCES

[1] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The museum tour-guide robot rhino. In Heinz Wörn, Rüdiger Dillmann, and Dominik Henrich, editors, Autonome Mobile Systeme 1998, pages 245–254, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[2] David Casagrande. real-time featureless visual odometry for sea floor imaging with a lagrangian float. Master’s thesis.

[3] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[4] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. Journal of the American Statistical Association, 93(443):1032–1044, 1998.

[5] Imbaby Mahmoud, May Salama, and Asmaa Tawab. Particle / kalman filter for efficient robot localization. International Journal of Computer Applications, 106:20–27, 11 2014.

[6] Francisco Martín. Visual localization based on quadtrees. In Luís Paulo Reis, António Paulo Moreira, Pedro U. Lima, Luis Montano, and Victor Muñoz-Martínez, editors, Robot 2015: Second Iberian Robotics Conference, pages 599–610, Cham, 2016. Springer International Publishing.

[7] Elias Mueggler, Matthias Faessler, Flavio Fontana, and Davide Scaramuzza. Aerial-guided navigation of a ground robot among movable obstacles. 10 2014.

[8] Chris Roman, Gabrielle Inglis, and Bryan McGilvray. Lagrangian floats as sea floor imaging platforms. Continental Shelf Research, 31(15):1592 – 1598, 2011.

[9] Thrun Sebastian, Fox Dieter, Burgard Wolfram, and Dellaert Frank. Robust monte carlo localization for mobile robots. Artificial Intelligence, 128(1):99 – 141, 2001.

[10] William Snyder, Chris Roman, and Stephen Licht. Hybrid actuation with complementary allocation for depth control of a lagrangian sea-floor imaging platform: Snyder et al. Journal of Field Robotics, 35, 05 2017.

[11] P. Stegagno, M. Cognetti, L. Rosa, P. Peliti, and G. Oriolo. Relative localization and identification in a heterogeneous multi-robot system. In 2013 IEEE International Conference on Robotics and Automation, pages 1857–1864, May 2013.
[12] Sebastian Thrun. Toward robotic cars. *Commun. ACM*, 53(4):99–106, April 2010.
APPENDIX

A. Robot Operating System implementation

Figure A.1 shows a more detailed flow-chart of robot localization package and how messages are exchanged. The Uncertainty Pre-Processor (UP) is used to associate predetermined uncertainties with the input data from the float messages. The Filter Madgwick block is used for turning magnetometer data into heading. Robot Localization is the ROS package that solves the Extended Kalman Filter equations. Robot localization will provide the transformation from local coordinate to robot coordinates. Data from the bathymetry map are loaded into the GridMap package and displayed in RViz. The non-parametric particle filter approach, Figure A.2, is similar in structure but replaces the robot localization package with the particle filter in the Bayesian Filter Library.
Figure A.1: ROS Robot Localization flowchart for the EKF
Figure A.2: ROS Robot Localization flowchart for particle filter.
BIBLIOGRAPHY

Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S., “The museum tour-guide robot rhino,” in Autonome Mobile Systeme 1998, Wörn, H., Dillmann, R., and Henrich, D., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 245–254.

Casagrande, D., “real-time featureless visual odometry for sea floor imaging with a lagrangian float,” Master’s thesis.

Hartley, R. I. and Zisserman, A., Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

Liu, J. S. and Chen, R., “Sequential monte carlo methods for dynamic systems,” Journal of the American Statistical Association, vol. 93, no. 443, pp. 1032–1044, 1998.

Mahmoud, I., Salama, M., and Tawab, A., “Particle / kalman filter for efficient robot localization,” International Journal of Computer Applications, vol. 106, pp. 20–27, 11 2014.

Martín, F., “Visual localization based on quadtrees,” in Robot 2015: Second Iberian Robotics Conference, Reis, L. P., Moreira, A. P., Lima, P. U., Montano, L., and Muñoz-Martinez, V., Eds. Cham: Springer International Publishing, 2016, pp. 599–610.

Mueggler, E., Faessler, M., Fontana, F., and Scaramuzza, D., “Aerial-guided navigation of a ground robot among movable obstacles,” 10 2014.

Roman, C., Inglis, G., and McGilvray, B., “Lagrangian floats as sea floor imaging platforms,” Continental Shelf Research, vol. 31, no. 15, pp. 1592 – 1598, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0278434311002433

Sebastian, T., Dieter, F., Wolfram, B., and Frank, D., “Robust monte carlo localization for mobile robots,” Artificial Intelligence, vol. 128, no. 1, pp. 99 – 141, 2001.

Snyder, W., Roman, C., and Licht, S., “Hybrid actuation with complementary allocation for depth control of a lagrangian sea-floor imaging platform: Snyder et al.” Journal of Field Robotics, vol. 35, 05 2017.

Stegagno, P., Cognetti, M., Rosa, L., Peliti, P., and Oriolo, G., “Relative localization and identification in a heterogeneous multi-robot system,” in 2013 IEEE International Conference on Robotics and Automation, May 2013, pp. 1857–1864.
Thrun, S., “Toward robotic cars,” *Commun. ACM*, vol. 53, no. 4, pp. 99–106, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721679