Distributed Terrain Estimation Using a Mixture-Model Based Algorithm

Jonathan R. Schoenberg and Mark Campbell
Sibley School of Mechanical and Aerospace Engineering
Cornell University, Ithaca, NY 14853
{jrs55, mc288}@cornell.edu

Abstract – *This paper describes a planar grid-based distributed terrain height estimation algorithm for use in a distributed data fusion sensor network. Each sensor node in the network represents the terrain using a Gaussian mixture to represent the elevation density in each grid cell. The local sensor node uses a rigorous probabilistic analysis of sensor measurement errors to associate individual measurements with multiple grid cells to account for in-plane uncertainty. Representing the elevation density in each grid cell as a sum of Gaussian distributions leads to a convenient channel filter implementation for distributed data fusion. The new information exchanged between nodes is assimilated locally to generate a global terrain height estimate on each sensor node. The distributed terrain height algorithm is demonstrated in a laboratory test environment using data from three sensor nodes over a 60 second data collection. The distributed terrain height algorithm is shown to perform equivalent to the centralized case even in the presence of communication failures.*

Keywords: Distributed data fusion, terrain mapping.

1 Introduction

Mobile robots require an accurate representation of their operating environment for path planning, localization, and exploration. The representation of the environment required by the robots depends on both the mission and capabilities of the mobile robot. Planetary exploration robots [1], or Unmanned Ground Vehicles (UGVs) performing security functions [2] need rich representations of the environment, including 3D terrain maps. The terrain maps can be generated via a variety of different sensors which provide depth information, including laser-line scanners, stereo vision, or depth from camera motion algorithms [3].

Robots equipped with laser range finders that scan a single line have been used to generate 2D occupancy grid maps [4] which represent the terrain as a planar grid, with each cell having a probability of being occupied or unoccupied. Elfes [4] identified and Moravec [5] implemented a natural extension of 2D occupancy grids with stereo-vision using voxels (3D pixels) to represent the occupancy status of a given rectangular prism. Alternative techniques have been developed for terrain mapping that have a more compact representation than 3D occupancy grid maps. Roth and Wibowo [6] present a technique for translating point data, such as that obtained from laser line scanners, to triangular meshes that handle spurious data and holes in the point clouds. Thrun, *et al.* [7] present a 3D mapping technique based on fine-grained multi-polygon surface models. Thrun *et al.* [8] make the polygon representation more compact by using a real-time variant of the EM algorithm to cluster range data and simultaneously estimate the number and shape of planes in the scan data. While these techniques provide rich and dense maps, they do not provide online evaluation about the accuracy of the maps.

Other 3D mapping techniques include 2.5D representations that model the height of grid cells in a planar map assuming independence of heights in neighboring grid cells. Bares *et al.* [9] create a 2.5D map using laser range finders that assigns the maximum height return in each grid cell to the terrain height. A similar technique was used by Kleiner and Dornhege [10] after creating Gaussian distributed terrain height estimates for each grid cell. Miller and Campbell [11] propose a mixture-model based technique, where the elevation of each grid cell is modeled as a mixture of Gaussian terrain height estimates. The mixture-model based algorithm provides an estimate of the elevation uncertainty in each grid cell. Pfaff *et al.* [12] propose a modified elevation map technique to handle environments with multiple dominant elevations; a dominant terrain height in each grid cell is selected. Triebel *et al.* [13] propose multi-level surface maps that where multiple elevations in a grid cell are represented by vertical and horizontal patches. Rivadeneyra *et al.* [14] combined the techniques of [11] and [13] to create probabilistic multi-level sets. All of these mapping techniques create dense maps that are different from those created from point obstacles [15] which are common in simultaneous localization and mapping (SLAM) procedures. Miller's mixture model based technique is used in this paper, because it provides a dense elevation map represented probabilistically, but it is extended to multiple robots.

Multi-platform mapping has been performed using a variety of different techniques. Fox *et al.* [16] point out that even in the context of SLAM, when the initial robot positions are known, or the robots are operating in a common coordinate system, multi-robot mapping is a simple extension of single robot mapping for centralized map creation. Thrun [17] performs multi-robot 3D mapping where the 3D map (polygons) from each local sensor node is broadcast to all other nodes and the received local map is inserted into the global map and fused using a quadratic error measure. This technique, relies on each robot localizing themselves in the map of the other robot prior to transmitting map information. Ryde and Hu [18] join 3D occupancy grid maps from different sensor nodes through an exhaustive search process. Carpin [19] proposes a technique based on spectral information in occupancy grid maps to fuse local maps into a global estimate. Martinez-Cantin [20] forces each robot to share a common map while performing marginal-SLAM, and fuses maps by matching features. Fox *et al.* [16] do not require initial positions of the robots to be known, but instead have the robots actively seek each other to determine their relative locations in order to fuse maps. All of these techniques essentially insert local maps into a global map, but none allow uncertainty or confidence associated with the map to be updated in the process.

The objective of sensor networks employing distributed data fusion (DDF) networks is to generate globally consistent estimates on each sensor node without the use of a centralized processing node, without a common communication bus (node-to-node communication only) and without requiring each node to have global knowledge of the network topology (neighborhood knowledge only) [21]. The advantages of DDF in a sensor network include robustness and modularity [22]. The channel filter is a convenient approach to performing DDF for general probability distribution representations [22]. None of the dense multi-robot mapping techniques described above adhere to the DDF paradigm. Nettleton [21] demonstrate how discrete feature maps, such as those generated in the SLAM paradigm, can be fused in a distributed data fusion paradigm, but these maps are not dense.

This paper demonstrates generating dense terrain maps from multiple robots in a common coordinate system. The local terrain maps are generated using Miller's mixture-model based approach which is good for memory scaling, accuracy, and formally dealing with measurement errors. The mixture-model based algorithm extends to multi-robot mapping by identifying compact information sets representing the map that are shared in a distributed sensor network. The channel filter is used for DDF and allows each local sensor node to update and maintain a terrain map that formally incorporates map information from other sensor nodes.

The remainder of the paper includes a summary of the mixture-model based terrain estimation algorithm for a single sensor node in Section 2. Section 3 extends the algorithm to the distributed data fusion paradigm using the channel filter. Sections 4 and 5 describe the laboratory experiments and results, and Section 6 finishes with conclusions.

2 Single Node Mixture-Model Based Terrain Estimation

The mixture-model based terrain estimation algorithm introduced by Miller and Campbell [11] translates laser scanner terrain detections into an elevation distribution for the height of the terrain in each cell in a planar grid. The algorithm begins by transforming the terrain detections to an inertial coordinate system accounting for uncertainties in the sensor alignments and measurements errors. Next, the terrain detections are probabilistically associated to cells in the terrain grid. An estimate of the elevation distribution in the cell is then generated from the measurements associated in each grid cell. The elevation density in each grid cell is assumed independent from one grid cell to the next. The key feature of [11] is representing each cell as a mixture, which greatly eases the ability to accumulate additional measurements into the terrain estimate, as shown in the remainder of this section.

The mixture model terrain estimation algorithm in [11] begins by defining a planar grid of N_c cells in an inertial coordinate system. Each rectangular grid cell is defined such that the j^{th} cell extends from E_{j-} to E_j+ in the Easting direction and N_{j-} to N_{j+} in the Northing direction. This places the center of grid cell j at $EN_j = [E_j\, N_j]^T$ where $E_j = \frac{1}{2}(E_{j-} + E_{j+})$ and $N_j = \frac{1}{2}(N_{j-} + N_{j+})$. The extent of each grid cell is also easily defined such that $\Delta E_j = (E_{j+} - E_{j-})$ and $\Delta N_j = (N_{j+} - N_{j-})$. The planar grid need not have a uniform cell size, but it is convenient to assume one without loss of generality.

The transformation of range and angle laser scanner measurements $\underline{r}$ to an inertial Cartesian coordinate system requires knowing the orientation of the sensor, defined by parameters $\underline{p}$, in the inertial coordinate system. For an East-North-Up (ENU) coordinate system, the transformed laser measurements are defined by a nonlinear function $f(\underline{p}, \underline{r})$ defined in (1).

$$\underline{r}^{ENU} \triangleq \begin{bmatrix} E \\ N \\ U \\ 1 \end{bmatrix} = f(\underline{p}, \underline{r}) \tag{1}$$

where defining $\underline{r}^{ENU}$ as a 4×1 vector allows for the use of 4×4 transformation matrices when converting between coordinate systems.

Unfortunately, no sensor measurement or alignment parameters are perfect. In fact, errors can arise from a variety of reasons including miss-calibration, thermal noise, and encoder quantization for gimbaled sensors. The observed sensor orientation parameters and laser measurements are corrupted by noise from the truth as defined in (2) and (3).

$$\hat{\underline{p}} = \underline{p} + v_{\underline{p}} \tag{2}$$

$$\hat{\underline{r}} = \underline{r} + v_{\underline{r}} \tag{3}$$

where $\nu_{\underline{p}}$ and $\nu_{\underline{r}}$ are the Gaussian distributed measurement errors that are assumed independent from each other with zero mean and covariances $Q_{\underline{p}}$ and $Q_{\underline{r}}$.

The transformed laser measurements with errors, are defined by $f(\underline{p}+\nu_{\underline{p}},\underline{r}+\nu_{\underline{r}})$, which is linearized about the observed values $\hat{\underline{p}}$ and $\hat{\underline{r}}$ in (2) and (3):

$$\underline{r}^{ENU} \approx f(\hat{\underline{p}},\hat{\underline{r}}) + \left.\frac{\partial f}{\partial \underline{p}}\right|_{\underline{p}=\hat{\underline{p}},\underline{r}=\hat{\underline{r}}} \nu_{\underline{p}} + \left.\frac{\partial f}{\partial \underline{r}}\right|_{\underline{p}=\hat{\underline{p}},\underline{r}=\hat{\underline{r}}} \nu_{\underline{r}} \tag{4}$$

$$= f(\hat{\underline{p}},\hat{\underline{r}}) + J_{\underline{p}}(\hat{\underline{p}},\hat{\underline{r}})\nu_{\underline{p}} + J_{\underline{r}}(\hat{\underline{p}},\hat{\underline{r}})\nu_{\underline{r}} \tag{5}$$

where $J_{\underline{p}}$ and $J_{\underline{r}}$ are the Jacobians of the measurement function f taken with respect to $\underline{p}$ and $\underline{r}$.

Using the linearized measurement function, it is possible to generate the posterior distribution of the terrain detection in the inertial coordinate system $r^{ENU} \sim p(E,N,U)$ that is approximated with a Gaussian distribution $p(E,N,U) \approx N(\hat{r}^{ENU},P_{\hat{r}}^{ENU})$. The mean and covariance of the posterior distribution of the terrain detection is given in (6) and (7), conditioned on all the information available (I).

$$\hat{r}^{ENU} = \begin{bmatrix} \hat{e} \\ \hat{n} \\ \hat{u} \\ 1 \end{bmatrix} = E\left[\underline{r}^{ENU}|I\right] = f(\hat{\underline{p}},\hat{\underline{r}}) = \begin{bmatrix} \hat{r}^{EN} \\ \hat{u} \end{bmatrix} \tag{6}$$

$$\begin{aligned} P_{\hat{r}}^{ENU} &= E\left[(\underline{r}^{ENU}-\hat{\underline{r}}^{ENU})(\underline{r}^{ENU}-\hat{\underline{r}}^{ENU})^T|I\right] \\ &= J_{\underline{p}}(\hat{\underline{p}},\hat{\underline{r}})Q_{\underline{p}}J_{\underline{p}}^T(\hat{\underline{p}},\hat{\underline{r}}) + J_{\underline{r}}(\hat{\underline{p}},\hat{\underline{r}})Q_{\underline{r}}J_{\underline{r}}^T(\hat{\underline{p}},\hat{\underline{r}}) \\ &= \begin{bmatrix} P_{\hat{r}}^{EN} & P_{\hat{r}}^{EN,u} \\ P_{\hat{r}}^{u,EN} & P_{\hat{r}}^{u} \end{bmatrix} \end{aligned} \tag{7}$$

where $\hat{r}^{ENU}$ and $P_{\hat{r}}^{ENU}$ are written in block form to identify specific portions that are used later in the algorithm. Equations (6) and (7) represent the terrain detection with Gaussian uncertainty associated with the source; the next step is to perform measurement-to-grid association.

The U dimension is marginalized out of each one of the M transformed terrain measurements $\{\hat{r}_i^{ENU},P_{\hat{r}_i}^{ENU}\}_{i=1}^{M}$ in order to generate a planar distribution of the terrain measurement uncertainty (8):

$$\begin{aligned} p_i(E,N) &= \int_{-\infty}^{\infty} p_i(\underline{r})dU = \int_{-\infty}^{\infty} p_i(E,N,U)dU \\ &= \int_{-\infty}^{\infty} \mathcal{N}\left(\hat{r}_i^{ENU},P_{\hat{r}_i}^{ENU}\right)dU \\ &= \mathcal{N}\left(\hat{r}_i^{EN},P_{\hat{r}_i}^{EN}\right) \end{aligned} \tag{8}$$

where each of the M terrain measurements from a single scan of the laser range finder is assumed independent.

This gives the in-plane Gaussian distribution of the terrain measurement, where $\hat{r}_i^{EN}$ and $P_{\hat{r}_i}^{EN}$ are the mean and covariance respectively. The probability of the terrain measurement originating from a specific cell j is computed as:

$$\begin{aligned} \Pr\left(\hat{r}_i^{EN} \in j\right) &= \int_{E_{j-}}^{E_{j+}} \int_{N_{j-}}^{N_{j+}} p_i(E,N)dNdE \\ &\approx (\Delta E_j)(\Delta N_j)\, p_i(E_j,N_j) \end{aligned} \tag{9}$$

where the probability is approximated using a single Riemann sum.

Next, attention turns to the in-cell height distribution of the i^{th} terrain measurement given it occurred at the center of the j^{th} grid cell. The univariate in-cell height distribution is found by taking the conditional distribution of the Gaussian defined by the mean (6) and covariance (7):

$$p(u_{i\in j}|E=E_j,N=N_j) \approx \mathcal{N}\left(\hat{u}_{i\in j},\sigma_{\hat{u}_{i\in j}}^2\right) \tag{10}$$

where the mean and covariance are found via standard conditioning rules of the Gaussian distribution [23]:

$$\hat{u}_{i\in j} = \hat{u}_i + P_i^{u,EN}\left(P_i^{EN}\right)^{-1}\left[EN_j - \hat{r}_i^{EN}\right] \tag{11}$$

$$\sigma_{\hat{u}_{i\in j}}^2 = P_i^u - P_i^{u,EN}\left(P_i^{EN}\right)^{-1}P_i^{EN,u} \tag{12}$$

Now, the set of all laser scanner measurements $\underline{r}^K \triangleq \left\{\{\rho_{ik},\theta_{ik}\}_{i=1}^M\right\}_{k=0}^K$ up to time K is defined, assuming M measurements per scan of the range ρ_{ik} and angle θ_{ik}, along with a set of sensor alignment measurements $\underline{p}^K = \left\{\underline{p}_k\right\}_{k=0}^K$. Each laser scanner measurement in the set $\underline{r}^K$ and sensor alignment measurement $\underline{p}^K$ have been mapped to each grid cell j with an association probability and height estimate $\left\{p_{ik\in j},\hat{u}_{ik\in j},\sigma_{\hat{u}_{ik\in j}}^2\right\}_{k=0}^K$. In practice, it is only necessary to compute the association probability and height estimates for a certain number of cells around the original laser scanner measurement[11].

Attention now turns to computing the distribution of the elevation in the jth cell given all of the terrain detections, denoted as $p(U_j|\underline{r}^K,\underline{p}^K)$. This is approximated as a Gaussian mixture:

$$p(U_j|\underline{r}^K,\underline{p}^K) \approx \frac{1}{c_j}\sum_{k=0}^K\sum_{i=1}^M p_{ik\in j}\cdot\mathcal{N}\left(\hat{u}_{ik\in j},\sigma_{\hat{u}_{ik\in j}}^2\right) \tag{13}$$

where $c_j = \sum_{k=0}^K\sum_{i=1}^M p_{ik\in j}$ is a normalizing constant. Considering just the first and second moment of this Gaussian mixture, the jth cell's terrain height is categorized by the mean and covariance of the Gaussian mixture [23]:

$$\hat{U}_{GM_j} = \frac{1}{c_j}\sum_{k=0}^K\sum_{i=1}^M p_{ik\in j}\hat{u}_{ik\in j} \tag{14}$$

$$\sigma_{\hat{U}_{GM_j}}^2 = \frac{1}{c_j}\sum_{k=0}^K\sum_{i=1}^M p_{ik\in j}\left(\hat{u}_{ik\in j}^2+\sigma_{\hat{u}_{ik\in j}}^2\right) - \hat{U}_{GM_j}^2 \tag{15}$$

This allows each sensor node to maintain a small set of recursively calculated simple statistics for each grid cell that are defined as the information set Z_j^K:

$$\begin{aligned} Z_j^K \triangleq \Bigg\{ & Z_p^K = \sum_{k=0}^K\sum_{i=1}^M p_{ik\in j},\ Z_{p\hat{u}}^K = \sum_{k=0}^K\sum_{i=1}^M p_{ik\in j}\hat{u}_{ik\in j}, \\ & Z_{p\hat{u}^2}^K = \sum_{k=0}^K\sum_{i=1}^M p_{ik\in j}\hat{u}_{ik\in j}^2,\ Z_{p\sigma_{\hat{u}}^2}^K = \sum_{k=0}^K\sum_{i=1}^M p_{ik\in j}\sigma_{\hat{u}_{ik\in j}}^2 \Bigg\} \end{aligned} \tag{16}$$

The information set Z_j^K is defined to be all of the accumulated information from the initial time up to and including time t_k. The update of the information set takes a simple recursive form for subsequent time steps. The update of one information set parameter is shown here for κ additional time steps of data:

$$\begin{aligned} Z_p^\kappa &= Z_p^K + Z_p^{K+1,K+\kappa} \\ &= Z_p^K + \sum_{k=K+1}^{K+\kappa} \sum_{i=1}^{M} p_{ik\in j} \end{aligned} \tag{17}$$

The information set for each grid cell allows the entire map with probabilistic accuracy information to be stored using four constants for each grid cell. More importantly, as new independent information becomes available from the local sensor node or remote sensor nodes, the information sets are updated with (17) and constant memory for a given grid cell is used.

3 Distributed Terrain Estimation

The distributed terrain estimation algorithm proposed here allows each sensor node to maintain an estimate of the global environment, in a form identical to the local mixture based terrain map on a single sensor node. In addition, it is desired that as all information is communication around the network, the distributed solution approaches the centralized data fusion solution. The channel filter [22] provides a convenient framework to perform DDF. The channel filter keeps track of the common information shared between sensor nodes on opposite ends of a data link. The network topology influences the implementation of the channel filter, and in the proposed approach, a tree-connected, acyclic network topology is assumed. The desire is for each sensor node i and j on a communication link to estimate the state x (terrain height of an individual cell) given the union of the information sets available at each node [22]:

$$p(x|Z_i^K \bigcup Z_j^K) = \frac{1}{c} \frac{p(x|Z_i^K)p(x|Z_j^K)}{p(x|Z_i^K \bigcap Z_j^K)} \tag{18}$$

where $p(x|Z_i^K)$ and $p(x|Z_i^K)$ are the posterior distributions including all information received at sensor nodes i and j, and $p(x|Z_i^K \bigcap Z_j^K)$ is the posterior distribution given all the common information contained in both information sets. One description of the channel filter [22] assumes the posterior distributions are Gaussian and uses the information form of the Kalman filter to maintain an estimate of (18). A similar technique could be used here for each grid cell, where the mean (14) and covariance (15) of the Gaussian mixture are used in the channel filter that assumes Gaussian posterior distribution. Fortunately, the mixture-model algorithm represents the terrain map with a succinct information set in each grid cell that enables tracking the union of the information sets in sensor nodes i and j directly.

The union of the information about the static terrain on a communication channel between sensor node i and j up to time k, is given by [22]:

$$Z_i^K \bigcup Z_j^K = Z_i^K + Z_j^K - Z_{i\bigcap j}^K \tag{19}$$

where $Z_{i\bigcap j}^K$ is the common information contained in sets Z_i^K and Z_j^K. The tree-connected network topology is now advantageous, because all common information between nodes i and j is assured to have come across the $i-j^{\text{th}}$ communication link [22]. Therefore, the common information up to time K is just the union of the information shared previously:

$$Z_{i\bigcap j}^K = Z_i^{K-1} \bigcup Z_j^{K-1} \tag{20}$$

Therefore, the local information set at node i up to time k, given all of the information sets in the neighborhood N_i (sensor nodes connected to i) where $i \notin N_i$ can now be updated:

$$Z_i^K = Z_i^{K-1} + Z_i^{K-1,K} + \sum_{j\in N_i} \left[\tilde{Z}_j^K - Z_{i\bigcap j}^{K-1} \right] \tag{21}$$

where $Z_i^{K-1,K}$ is the new information accumulated at sensor i locally (i.e. from new laser measurements) from t_{k-1} to t_k and $\tilde{Z}_j^K$ is the information received from neighboring sensor node j at time t_k. The received information from neighboring node may be delayed from the time it was generated on the remote sensor node. The locally updated and assimilated information set (21) from node i is placed as the output to any neighboring nodes connected to sensor node i for subsequent transmission.

It is still necessary to update the common information on link $i-j$, $\forall j \in N_i$ for the subsequent time step:

$$Z_{i\bigcap j}^K = Z_i^K + \tilde{Z}_j^K - Z_{i\bigcap j}^{K-1} \tag{22}$$

The common information on link $i-j$ is updated with the received information $\tilde{Z}_j$ from sensor j and the updated and assimilated information Z_i^K sent out from node i as shown in the data flow for sensor node i in Figure 1. The channel filter implementation used here needs a tree-connected topology to ensure all of the information received from node j at node i comes across link $i-j$. In addition, sensor node i needs to maintain a channel filter for each neighboring node, therefore, the branching factor drives the computation and memory requirements on an individual node.

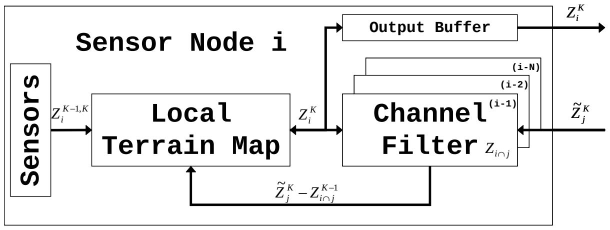

Figure 1: Data flow for sensor node i demonstrates distributed terrain mapping with channel filter.

In the case of communication failure, the local information set at node i is updated (21) without information $\tilde{Z}_j^K$ from nodes whose communication links have failed such that the updated information is $Z_i^K = Z_i^{K-1} + Z_i^{K-1,K}$. The channel filter for the $i-j^{\text{th}}$ communication link remains unchanged while the link is down. If communication failure persists for κ time steps, the local sensor nodes continue to update local information (21) and when communication is restored the information from neighboring nodes $Z_j^{K+\kappa}$ is received and assimilated at node i into $Z_i^{K+\kappa}$, this information from sensor node i is sent out and the channel filter (22) is updated to $Z_{i\cap j}^{K+\kappa}$.

To summarize the distributed terrain algorithm, the information set for each grid cell is updated and received information assimilated locally (21), while the channel filter keeps track of the common information (22) on any communication link for each of the four parameters in each grid cell. Using the locally updated and assimilated information set (21), the final distribution of the in-cell terrain height estimate is computed using (14) and (15).

4 Laboratory Experiment

The data collection was performed in Cornell University's Autonomous Systems Laboratory (ASL) using the Pioneer P3-DX differential drive mobile robot from Mobile Robots Inc. shown in Figure 2. The P3-DX robot base is equipped with a custom built Mini-ITX computer running the Orca Robotics [24] software framework for sensor integration and control. The primary sensor for the terrain height estimation task is the compact Hokuyo URG-04X laser scanner which features a 240$^{\text{o}}$ field-of-view and angular resolution of 0.36$^{\text{o}}$. The laser is pitched downward 45$^{\text{o}}$ and scans along the ground as the robot moves forward in a push-broom fashion.

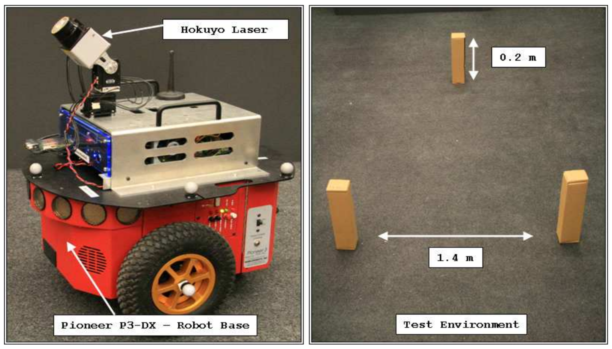

Figure 2: Mobile robot used in Cornell University's ASL equipped with laser scanner and on-board computer and the test environment with three obstacles.

The test environment is 3.1×4.1 meters and is instrumented with a Vicon MX+ precision tracking system that determines position and attitude of 3D objects (notice tracking markers on the mobile robot in Figure 2) in the test environment and is used to localize the robots. The terrain features in the environment consist of three boxes $(20 \times 7.5 \times 4.5)$ cm tall that are meant to simulate traffic cones or other similarly sized obstacles for a full-size traffic vehicle. The boxes are spaced approximately 1.4 m from each other in a triangular pattern.

Three robots are run in different paths around the environment for a 60 second data collection. The robots are run sequentially to avoid sensing another robot (dynamic terrain) during the terrain mapping. The three paths of the robots are shown in Figure 3, along with the raw laser detections. The three boxes are clearly visible from the raw laser scans. The paths of the robots allow each one to cover a different

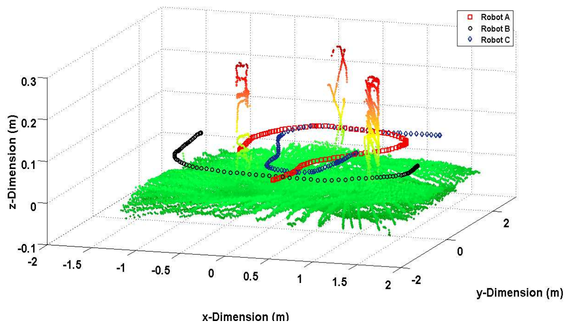

Figure 3: Three mobile robots collect laser scans over 60 seconds for distributed terrain mapping.

portion of the entire space, while the distributed terrain estimation algorithm combines data from each sensor. Robot A covers the area above -0.5 m in the y-dimension and observes all three obstacles. Robot B covers the area below 1.0 m in the y-dimension and only observes the top of the upper obstacle, but observes the entirety of the other obstacles. Robot C covers the longest path in the same amount of time and as a result observes the obstacles quickly.

The robots are connected in a chain topology such that Robot A and B communicate and Robot B and C communicate. The individual communication links are assumed independent of one another, meaning data flowing on one link (i.e. A to B) does not imply data is transfered on the other link (B to C). The robots attempt to communicate at approximately 0.2 Hz, and experiments are run with 100% and 50% probability of communication success on the link between each robot. The challenge of the distributed terrain estimation algorithm is to fuse all of the raw terrain measurements into a common estimate of the terrain on each robot.

5 Results

The first experiment develops the benchmark performance using a centralized terrain estimation algorithm. Each robot communicates all raw measurements when they are taken to a central server; the server performs terrain estimation using the single node terrain mapping algorithm. Additionally, each sensor node operates independently (without communicating to other nodes or receiving data from the central server) to generate a local terrain map. The centralized solution is the goal of the distributed terrain estimation algorithm. The local terrain map generated by sensor node B is shown in Figure 4. Sensor node B has a significant amount of distortion in the map due to vibration induced sensor alignment errors, despite each sensor node having the same measurement error characteristics.

The path the robot taken by Robot B causes the top obstacle $(0.25, 1.75)$ to be observed on a edge of the laser line scanner field of view. This causes only a few terrain measurements to be associated with the grid cells of the obstacle resulting in poor accuracy of the height estimate. The local terrain map for sensor node C is shown in Figure 5. Sensor node C clearly observes each one of the obstacles, but is moving quickly through the environment, which degrades the accuracy compared to a sensor that moves more slowly. The additional error is due to less sensor data per grid cell on average. In addition, sensor node C does not observe any terrain measurements in the upper right portion of the environment.

The centralized terrain map at the end of the data collection is shown in Figure 6 and demonstrates the ability of the three robots to effectively cover the entire exploration space and generate an accurate height estimate for the three obstacles.

Another way of analyzing the data fusion problem is to track the available information about a particular area of the map during the data collection. The mixture-model based terrain estimation algorithm provides a convenient metric, the measurement-to-grid association probability Z_{p_j} (16), for tracking information content of a specific grid cell. For the terrain mapping problem, the most important areas of the map are the obstacles, and the desire is to keep track of the information content over time at sensor node i in the grid cells around the obstacles at a specific time t:

$$\mathscr{I}_{\text{Node}^i}(t) \triangleq \sum_{j \in C_O} Z^i_{p_j}(t) \tag{23}$$

where C_O represents the cells around the obstacles. The information in node i is monotonically increasing, because there is no way to remove association probability from a given grid cell. Therefore, the total information available in the entire sensor network at a given time step t is:

$$\mathscr{I}_{\text{Sensor Network}}(t) \triangleq \sum_i \mathscr{I}_{\text{Node}^i}(t) \tag{24}$$

and the information available over the entire data collection $\mathscr{I}^* \triangleq \mathscr{I}_{\text{Sensor Network}}(t_f)$ is given at the final time step t_f. For the centralized data fusion case, the information available at each sensor node over the 60 second data collection is shown in Figure 7 as a percentage of the total information available in the entire network $\mathscr{I}^*$. Robot B contributes approximately 50% of the total information about the obstacles, because it's path is close to the two lower obstacles. Robot A and C each contribute approximately 25% of the total information, and the central server accumulates 100% of the available information by the end of the data collection.

For the distributed data fusion case, the sensor nodes use the channel filter to manage the exchange of information about each grid cell from node-to-node. Each sensor node accumulates local measurements while updating their local terrain map (17) between communicating and assimilating information from neighboring nodes (21). The assimilation

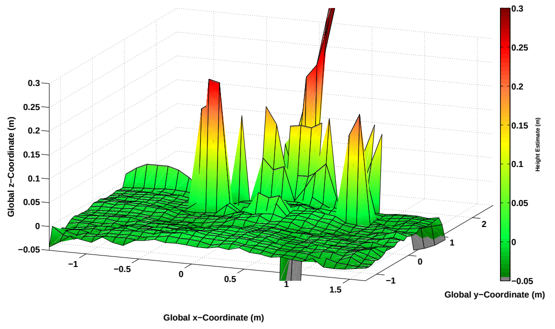

Figure 4: Local terrain height estimate (mean+1σ) for sensor node B shows poor accuracy due to sensor alignment errors and observation of the top obstacle on the edge of the laser scanner field-of-view.

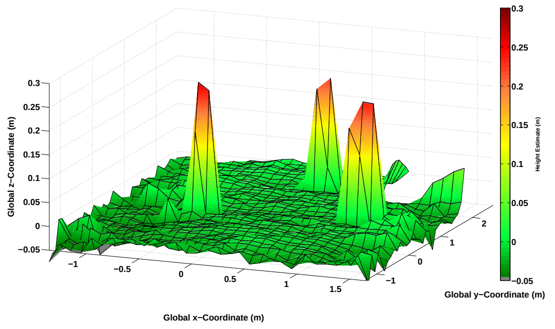

Figure 5: Local terrain height estimate (mean+1σ) for sensor node C shows it does not completely observe the entire area.

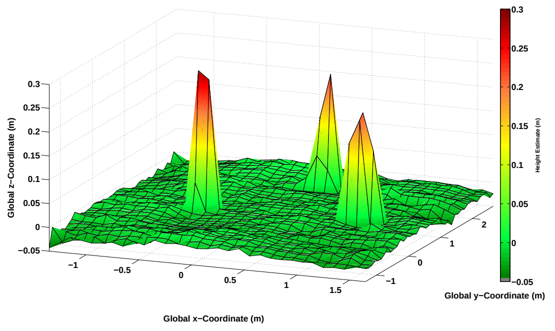

Figure 6: Centralized terrain map (mean+1σ) over the 60 second data collection shows three obstacles with accurate height estimates.

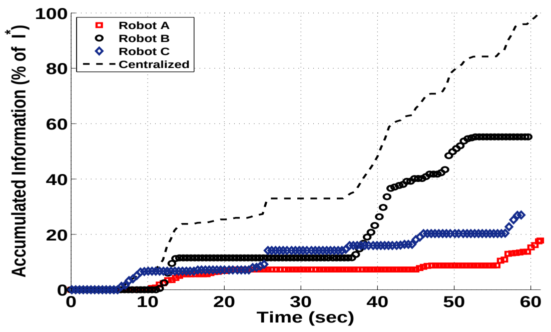

Figure 7: Centralized: Accumulated information about the obstacles for each sensor node and the central server.

of information from neighbors is performed prior to sending map information to neighbors to allow for a single communication attempt when desired. If communication is successful across a channel, the channel filter updates the prior information communicated (22). Nettleton *et al.* [21] identifies that these characteristics of the channel filter leads to scalability of the sensor network and robustness.

The information about the obstacles for the distributed mapping approach is shown in Figure 8. The disjoint jumps in information at a given sensor node correspond to communication and assimilation of neighboring information from the channel filter. Continuous increases correspond to information gain through the actual mapping process on the local sensor node. The time interval between 10 – 15 seconds in the data collection shows Robot A and B accumulating information about the obstacles locally, while Robot C is not gaining any information about the obstacles. However, when the communication occurs at 15 seconds, each sensor node receives the total information available in the entire network about the obstacles.

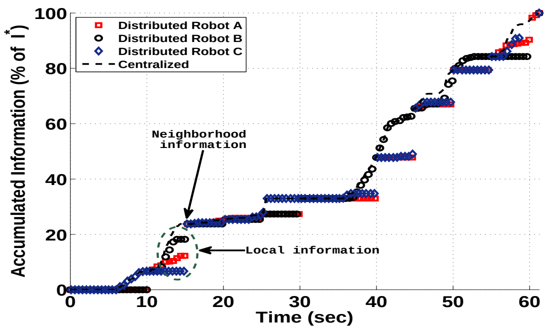

Figure 8: Distributed mapping: Accumulated information about the obstacles for each node in the distributed data fusion network is equivalent to the central server with 100% communication success.

One advantage of the channel filter for DDF is robustness to communication link failure. The channel filter tracks information exchanged between nodes, and during communication failure the channel filter is not updating because no information is being exchanged between nodes. Furthermore, because the communicated information in the distributed mapping problem is the four parameters about each grid cell, there is no need to buffer a sequence of data to transmit. Instead, the output message for communication is simply the cells that have had changes in the local map since the last communication. If the sensor node attempts, but fails to communicate, the output message is simply updated to include any new changes in the local map between the communication failure and a subsequent attempt.

The ability of the channel filter to robustly handle communication failure is demonstrated in Figure 9. The accumulated information for Robot A shows the communication channel to Robot B is down from 40 – 60 seconds. During that time, Robot A is continuing to accumulate self-information from terrain measurements, but does not receive any data from Robot B. At 60 seconds, the communication

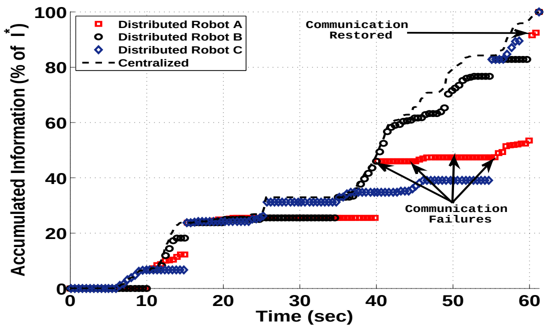

Figure 9: Communication failure: Accumulated information about the obstacles for each sensor node in the distributed data fusion network achieves the centralized fusion node performance with 50% communication failure.

link is restored and the information content jumps dramatically. At the end of the data collection, the robots continue to attempt communication until they are successful. This ensures the DDF performance achieves the centralized data fusion performance by the end of the data collection even in the presence of communication failure.

6 Conclusion

The distributed terrain estimation algorithm is used to update and maintain globally consistent terrain maps at each node in a sensor network. The use of the mixture-model based algorithm for terrain estimation provides compact information sets about each grid cell to share for data fusion. The channel filter is ideally suited for the distributed data fusion using these information sets, because it tracks the previously communication information and successfully updates each node, even in the presence of communication failure. The distributed terrain estimation generates dense maps on each sensor nodes that maintain the metrics about map accuracy and confidence. The algorithm was successfully demonstrated using a 60 second data collection in a laboratory setting.

Acknowledgment

The authors would like to thank the members of the Cornell University Autonomous Systems Laboratory who made data collection possible. This work is supported under the Northrop Grumman Electronic Systems Scholars Program.

References

[1] S. Se, T. Barfoot, and P. Jasiobedzki, "Visual motion estimation and terrain modeling for planetary rovers," in *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, B. Battrick, Ed. Munich, Germany: ESA SP-603, 2005, pp. 101–109.

[2] M. Ani Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie, "Adaptive teams of autonomous aerial and

ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991–1014, 2007.

[3] K. Fregene, D. Kennedy, R. Madhavan, L. E. Parker, and D. Wang, "A class of intelligent agents for coordinated control of outdoor terrain mapping ugvs," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 5, pp. 513–531, 2005, 0952-1976 doi: DOI: 10.1016/j.engappai.2004.12.007.

[4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989, 0018-9162.

[5] H. P. Moravec, "Robot spatial perception by stereoscopic vision and 3d evidence grids," The Robotics Institute Carnegie Mellon University, Tech. Rep., 1996.

[6] G. Roth and E. Wibowo, "An efficient volumetric method for building close triangular meshes from 3-d image and point data," in *Graphics Interface 97*, Kelowna, B.C., Canada, 1997, pp. 173–180.

[7] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 321–328 vol.1.

[8] S. Thrun, C. Martin, L. Yufeng, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard, "A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 3, pp. 433–443, 2004, 1042-296X.

[9] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker, "Ambler: an autonomous rover for planetary exploration," *Computer*, vol. 22, no. 6, pp. 18–26, 1989, 0018-9162.

[10] A. Kleiner and C. Dornhege, "Real-time localization and elevation mapping within urban search and rescue scenarios: Field reports," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 723–745, 2007.

[11] I. Miller and M. E. Campbell, "A mixture-model based algorithm for real-time terrain estimation," *J. Robot. Syst.*, vol. 23, no. 9, pp. 755–775, 2006.

[12] P. Pfaff, R. Triebel, and W. Burgard, "An efficient extension to elevation maps for outdoor terrain mapping and loop closing," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 217–230, 2007.

[13] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, p. 22762282.

[14] C. Rivadeneyra, I. Miller, J. R. Schoenberg, and M. Campbell, "Probabilistic estimation of multi-level terrain maps," in *Robotics and Automation, 2009. Proceedings. ICRA '09. IEEE International Conference on*, Kobe, Japan, 2009, p. In print.

[15] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," pp. 333–349, 1997.

[16] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, 2006, 0018-9219.

[17] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

[18] J. Ryde and H. Hu, "Mutual localization and 3d mapping by cooperative mobile robots," in *9th International Conference on Intelligent Autonomous Systems*. The University of Tokyo, Tokyo, Japan: IOS Press, 2006, pp. 217–224.

[19] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Autonomous Robots*, vol. 25, no. 3, pp. 305–316, 2008, 10.1007/s10514-008-9097-4.

[20] R. Martinez-Cantin, J. A. Castellanos, and N. de Freitas, "Multi-robot marginal-slam," in *International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.

[21] E. W. Nettleton, H. F. Durrant-Whyte, P. W. Gibbens, and A. H. Goektogan, "Multiple-platform localization and map building," in *Sensor Fusion and Decentralized Control in Robotic Systems III*, vol. 4196. Boston, MA, USA: SPIE, 2000, pp. 337–347.

[22] A. Makarenko and H. Durrant-Whyte, "Decentralized bayesian algorithms for active sensor networks," *Information Fusion*, vol. 7, no. 4, pp. 418–433, 2006, 1566-2535 doi: DOI: 10.1016/j.inffus.2005.09.010.

[23] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Wiley and Sons, Inc, 2001.

[24] A. Makarenko, A. Brooks, and T. Kaupp, "On the benefits of making robotic software frameworks thin," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'07)*, San Diego CA, USA, 2007.