Robust LIDAR Localization using Multiresolution Gaussian Mixture Maps for Autonomous Driving

Ryan W. Wolcott and Ryan M. Eustice

Abstract

This paper reports on a fast multiresolution scan matcher for *local* vehicle localization of self-driving cars. State-of-the-art approaches to vehicle localization rely on observing road surface reflectivity with a three-dimensional (3D) light detection and ranging (LIDAR) scanner to achieve centimeter-level accuracy. However, these approaches can often fail when faced with adverse weather conditions that obscure the view of the road paint (e.g., puddles and snowdrifts), poor road surface texture, or when road appearance degrades over time. we present a generic probabilistic method for localizing an autonomous vehicle equipped with a 3D LIDAR scanner. This proposed algorithm models the world as a mixture of several Gaussians, characterizing the z-height and reflectivity distribution of the environment—which we rasterize to facilitate fast and exact multiresolution inference. Results are shown on a collection of datasets totaling over 500 km of road data covering highway, rural, residential, and urban roadways, in which we demonstrate our method to be robust through heavy snowfall and roadway repavements.

1 Introduction

In order to navigate autonomously, the prevalent approach to self-driving cars requires precise localization within an *a priori* known map. Rather than using the vehicle's sensors to explicitly extract lane markings, traffic signs, etc., metadata is embedded into a prior map, which reduces the complexity of perception to a localization problem. State-of-the-art methods (Levinson et al., 2007; Levinson and Thrun, 2010) use reflectivity measurements from three-dimensional (3D) light detection and ranging (LIDAR) scanners to create an orthographic map of ground-plane reflectivities. Online localization is then performed with the current 3D LIDAR reflectivity scans and an inertial measurement unit (IMU).

Reflectivity-based methods alone can fail when the road appearance is degraded over time or occluded by harsh weather. In this work, we seek a fast, optimal scan matcher that allows us to quickly localize a vehicle within a prior map by exploiting the 3D structure of the scene in addition to ground-plane reflectivities.

We propose the use of a pair of Gaussian mixture maps—a two-dimensional (2D) grid structure where each grid cell contains a Gaussian mixture model. One such map characterizes the distribution over z-height (i.e., vertical structure) and another for capturing the distribution over reflectivity (appearance). Gaussian mixture maps allow us to fully extract all point cloud data while mapping and compressing the distributions into a compact, parametric representation.

When used for localization, we can again use all online

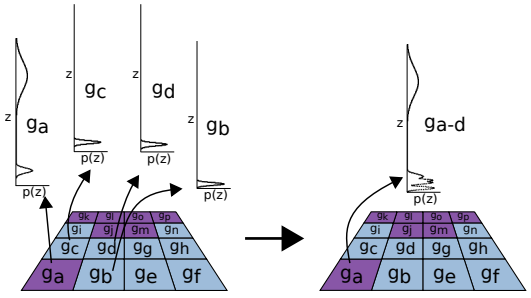


Fig. 1: Overview of our proposed LIDAR localization scheme. We propose the use of Gaussian mixture maps—a 2D grid over xy where each cell in the grid holds a one-dimensional Gaussian mixture model that accurately models the distribution of points contained in this infinite-height cell. We consider two representations that independently model z-height and reflectivity of points, then perform registration in these maps by formulating a branch-and-bound search over multiresolution, rasterized versions of the Gaussian mixture maps where coarser resolutions provide an upper-bound over the finer resolutions. This methodology finds the guaranteed optimal registration over a user-specified search space. The figure on the left depicts a z-height Gaussian mixture map, where the grid is colored by the difference between the two Gaussian modes in the cell, *blue* indicates 2 overlapping mixture components around the ground-plane and *purple* indicates two distinct modes captured including ground-plane and superstructure; the figure on the right shows the multiresolution look-up tables that our method uses.

Draft manuscript, April 9, 2017.

This paper was presented in part at the IEEE International Conference on Robotics and Automation, Seattle, WA, USA 2015 (Wolcott and Eustice, 2015).

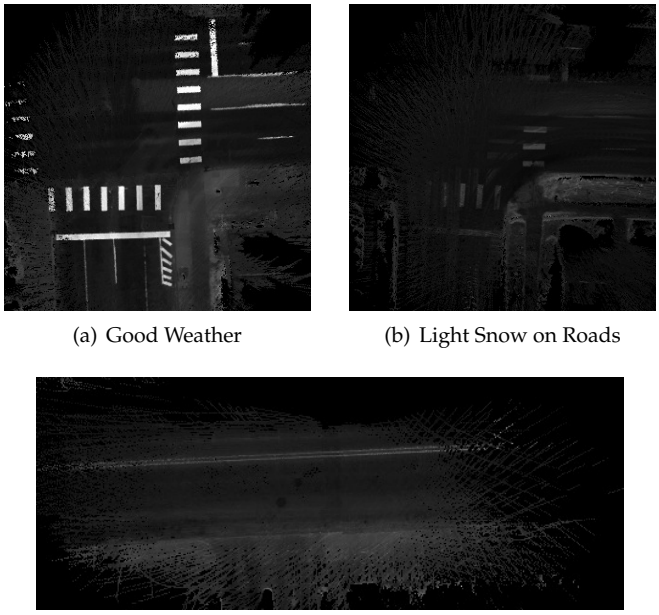
R. Wolcott is with the Computer Science and Engineering Division, University of Michigan, Ann Arbor, MI 48109, USA and Ford Motor Company, Dearborn, Michigan, USA rwolcott@umich.edu.

R. Eustice is with the Department of Naval Architecture & Marine Engineering, University of Michigan, Ann Arbor, Michigan, USA eustice@umich.edu.

point cloud data to register against these maps, thus improving robustness of our method by avoiding the need to extract higher level features to perform registration. While this registration may appear expensive, we present a novel upper-bound through rasterizations of the sum of Gaussian mixtures that enables us to formulate the scan matching problem as a branch-and-bound search. See Fig. 1 for a sample of these maps.

The key contributions of this paper over our previous conference paper (Wolcott and Eustice, 2015) include:

- Data reduction of large point clouds to a compact mixture of Gaussians, capturing both the structure and appearance of a point cloud.
- Online rasterization of these parametric maps that enables fast branch-and-bound registration formulation for real-time, guaranteed-optimal registration, using generic upper-bound rasterizations.
- Robust registration that jointly considers structure and point appearance using a robust cost function for discarding outliers.
- Implementation of our algorithms on a graphics processing unit (GPU) that yields $40\times$ speedup over the central processing unit (CPU), which allows us to localize using all measured 3D points without spatial downsampling.
- Extensive evaluation over several hundred kilometers of road data, in which we demonstrate successful localization through diverse environments including heavy snowfall, construction, and asphalt repaving—all demonstrating robustness to appearance changes.



(a) Good Weather (b) Light Snow on Roads

(c) Poor Texture in Road

Fig. 2: Common snapshots of orthographic LIDAR reflectivity maps. Notice the severe degradation of quality in the snow covered roads and the hallucination of lane markings caused by tire tracks through snow. Also, poor texture is a common occurrence on two-lane roads, which often result in laterally constrained cost functions.

2 Related Work

Automated vehicles require robust localization algorithms with low error and failure rates. One of the most pervasive strategies relies on observation of ground plane reflectivities, a signal that captures lane markings, pavement variation, tar strips, etc. Levinson et al. (2007) initially proposed using a 3D LIDAR scanner to observe the ground-plane reflectivities, with which they were able to build orthographic maps of ground reflectivities and perform localization using the current 3D LIDAR scans and an IMU. Baldwin and Newman (2012) employed a similar approach by using a 2D LIDAR scanner to build 3D swathes as the vehicle traversed the environment. In previous work, we demonstrated that ground-plane reflectivities can also be used to localize a monocular camera in a 3D LIDAR reflectivity map (Wolcott and Eustice, 2014).

Despite attempts by Levinson and Thrun (2010) to model slight changes in appearance of these ground plane maps by considering the variance of the prior map (in addition to previous methods that only captured the mean), appearance based methods can fail when harsh weather is present in the environment—for example, rain puddles and snowdrifts can build up and occlude the view of the informative ground signal, see Fig. 2. Additionally, long two-lane roads with a double lane-marker between them can allow longitudinal uncertainty to grow unbounded due to lack of texture perpendicular to the road. Thus, to increase robustness to these types of scenarios, we are interested in augmenting these appearance methods by exploiting the 3D structure of the scene that is observed with a LIDAR scanner in a fast and efficient manner.

Specifically, we are interested in registering a locally observed point cloud to some prior 3D representation of our environment. Many similar robotic applications use iterative closest point (ICP) (Besl and McKay, 1992), generalized iterative closest point (GICP) (Segal et al., 2009), normal distributions transform (NDT) (Magnusson, 2009), or other similar variants to register an observed point cloud to another point cloud or distribution. Registration using these methods typically requires defining a cost function between two scans and evaluating gradients (either analytical or numerical) to iteratively minimize the registration cost. Due to the nature of gradient based optimization, these methods are highly dependent on initial position and are subject to local

minimums.

To overcome local minima and initialize searches near the global optimum, several works have been proposed that extract distinctive features and perform an alignment over these first. For example, Rusu (2009) and Aghamohammadi et al. (2007) presented different features that can be extracted and matched from a raw point cloud. Pandey et al. (2011) bootstrap their registration search with visual feature correspondences (e.g., SIFT). However, these feature-based approaches rely on extracting robust features that are persistent from various viewpoints.

Other research has instead considered using RANSAC to solve the global alignment problem, as is done with the 4-Points Congruent Sets (4PCS) method of Aiger et al. (2008), Mellado et al. (2014)'s Super-4PCS algorithm, and the congruent pyramids of Krishnan and Saripalli (2014). While all of these methods can handle large displacements and low-overlap of point clouds, they only solve global registration and require a refinement using ICP methods for local registration; therefore, many of these methods cannot be considered for real-time application on an autonomous car.

As an alternative to searching for a single best registration for each scan, Chong et al. (2013), Kümmerle et al. (2007), and Maier et al. (2012) all demonstrated localization implementations built upon a Monte Carlo framework. Their approach allows particles to be sampled throughout the environment and evaluated relative to a prior map. This filtering methodology should be more robust to local minima because the particles should ideally come to a consensus through additional measurements—though this is dependent on random sampling and can make no time-based optimality guarantees.

Finally, multiresolution variations on the above algorithms have been proposed that allow expanded search spaces to be explored in a coarse-to-fine manner in hopes of avoiding local minima. This has been applied to ICP (Granger and Pennec, 2002), NDT (Magnusson, 2009; Ulaş and Temelta, 2013; Ripperda and Brenner, 2005), and occupied voxel lists (Ryde and Hu, 2010). These searches use heuristics to greedily guide the coarse-to-fine steps that yield good results in practice, but still cannot guarantee global optimality.

We employ techniques presented by Olson (2009, 2015) to formulate the multiresolution search as a branch-and-bound problem that can guarantee *global* optimality over our search space. In this work, we extend Olson (2009) to handle full-3D point clouds by creating efficient Gaussian mixture maps for fast and accurate inference, which is quite similar to the Gaussian mixture point cloud representation of Jian and Vemuri (2011). As in Maddern et al. (2015), we formulate a joint cost function that allows LIDAR localization using z-height and reflectivity maps, though our approach captures the full distribution rather than just a mean and a fixed variance.

3 Prior Maps from Pose Graphs

The first portion of our localization framework is the offline mapping stage, which generates the map to be used for online localization. Our goal here is to generate a map that is metrically accurate to the environment. To do this, we use the state-of-the-art in nonlinear least-squares, pose-graph simultaneous localization and mapping (SLAM) and measurements from a 3D LIDAR scanner to map the 3D structure in a global frame.

Prior to the offline mapping stage, our robot has no *a priori* knowledge of the environment, thus, we must employ SLAM to build a model of the environment while simultaneously localizing within that environment. We construct a pose-graph to solve the full SLAM problem as shown in Fig. 3, where nodes in the graph are poses (X) and edges are either odometry constraints (U), laser scan matching constraints (Z), or global positioning system (GPS) prior constraints (G). These constraints are modeled as Gaussian random variables; therefore, we model the joint distribution over poses and constraints as

$$P(X,U,Z,G) = \prod_{i=1}^{M} P(\mathbf{x}_i|\mathbf{x}_{i-1},\mathbf{u}_i) \prod_{k=1}^{K} P(\mathbf{z}_k|\mathbf{x}_{i_k},\mathbf{x}_{j_k}) \prod_{a=1}^{A} P(\mathbf{g}_a|\mathbf{x}_a), \tag{1}$$

where there are M poses, K loop closures, and A GPS prior constraints. Thus, to solve the SLAM problem, we seek to find the maximum *a posteriori* (MAP) estimate over the robot poses by minimizing the negative log of the joint probability:

$$X^* = \arg\max_X P(X,U,Z,G) \tag{2}$$

$$= \arg\min_X -\log P(X,U,Z,G) \tag{3}$$

$$= \arg\min_X \sum_{i=1}^{M} \|f_i(\mathbf{x}_{i-1},\mathbf{u}_i) - \mathbf{x}_i\|^2_{\Sigma_i} + \sum_{k=1}^{K} \|h_k(\mathbf{x}_{i_k},\mathbf{x}_{j_k}) - \mathbf{z}_k\|^2_{\Sigma_k} + \sum_{a=1}^{A} \|h_a(\mathbf{x}_a) - \mathbf{g}_a\|^2_{\Sigma_a}, \tag{4}$$

where $f_i(\cdot)$ is our process model, $h_k(\cdot)$ is our scan matching measurement model, and $h_a(\cdot)$ is our GPS measurement model. Each is corrupted by normally distributed noise with covariance Σ_i, Σ_k, and Σ_a, respectively. This summation equates to solving a nonlinear least-squares problem. We use incremental smoothing

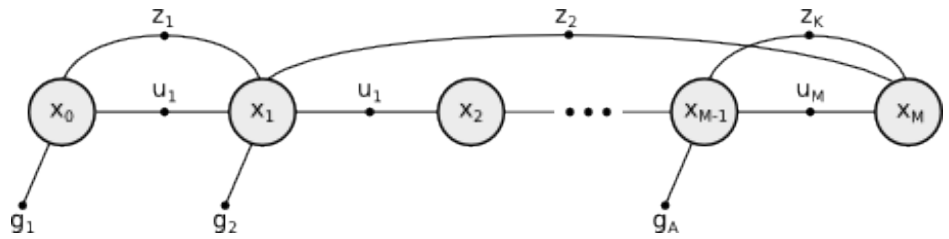


Fig. 3: Factor graph of the pose-graph SLAM problem that we solve in the off-line mapping stage. Here, x_i represents states of the robot, u_m represents incremental odometry measurements, z_k represents laser scan-matching constraints, and g_a are GPS prior measurements.

and mapping (iSAM) (Kaess et al., 2008), which uses incremental QR factorization to solve this nonlinear least-squares problem.

Since map construction is an offline task, we do not construct our pose-graph temporally. Instead, we first construct a graph with only odometry and GPS prior constraints. With this skeleton pose-graph in the near vicinity of the global optimum, we use Segal et al. (2009)'s GICP to establish 6-degree of freedom (DOF) laser scan-matching constraints between poses; adding both odometry constraints (temporally neighboring poses) and loop closure constraints (spatially neighboring poses) to our pose-graph.

4 Gaussian Mixture Maps

The key challenge to fast localization is a prior representation of the world that facilitates efficient inference. We propose using Gaussian mixture maps that discretize the world into a 2D grid over the xy plane, where each cell in the grid contains a Gaussian mixture that characterizes the 3D points contained within this infinite column. To capture both structure and appearance, we construct a pair of independent Gaussian mixture maps that capture the z-height and reflectivity distribution of each cell, respectively.

The Gaussian mixture map over z-height offers a compact representation that is quite similar to a 2.5D map, with the flexibility of being able to simultaneously and automatically capture the multiple modes prevalent in the world—including tight distributions around the ground-plane and wide distributions over superstructure. This representation is quite similar to NDT maps—we see Gaussian mixture maps existing in the space between the 2D-NDT (Biber, 2003) and the 3D-NDT (Magnusson, 2009). Like these approaches, Gaussian mixture maps can be viewed as a Gaussian mixture over the environment, though our maps are a collection of discontinuous one-dimensional Gaussians rather than a continuous multivariate Gaussian. This means that, when registering a point, likelihood evaluation is a function of z conditioned on the corresponding xy cell the point falls in.

Moreover, the z-height Gaussian mixture map is also similar to multi-level surface (MLS) maps from Triebel et al. (2006), which cluster the point cloud into *horizontal* and *vertical* structure components using distance-based heuristics. Rather than reducing our point cloud into similar discrete intervals to characterize the z-height distribution, we instead run expectation-maximization (EM) to fit a Gaussian mixture model for each grid cell to capture the true probabilistic distribution of our observed point cloud.

The reflectivity Gaussian mixture map is a generalized version of the probabilistic reflectivity maps presented by Levinson and Thrun (2010). Their approach fits a single Gaussian per cell, while Gaussian mixture maps can fit more than one mode to capture above-ground appearance features (e.g., signs, building facades, foliage) as well as accurately capture the true distributions at the edge of lane markers.

The reminder of this section details construction of these maps and how they are used in a joint framework for robustly registering a vehicle equipped with LIDAR sensors.

4.1 Map Construction

Beginning with our skeleton pose-graph detailed in Section 3, we have a ground truth set of poses, $X = \{\mathbf{x}_i\}_{i=0}^{M}$ optimized into a locally consistent frame. Each ground truth pose has a corresponding point cloud, $\mathcal{P}_i = \{\mathbf{p}_j\}_{j=1}^{n}$, where $\mathbf{p}_j = [x_j, y_j, z_j, r_j]^\top$ is the metric position of a point in space and its corresponding reflectivity as measured by our laser scanner. Each point is motion compensated according to our odometry to account for motion during point cloud acquisition.

We then reproject each of these point clouds into the SLAM optimized frame, accumulating into a single, global point cloud, $\mathcal{P} = \{\mathbf{x}_i \oplus \mathcal{P}_i\}_{i=0}^{M}$, where $\oplus$ denotes the head-to-tail composition operation (Smith et al., 1990) transforming each body-frame point cloud into the SLAM frame. Accumulating every point directly would be inefficient in memory and computation, so we instead use a sparse histogram implemented with a hash table. Thus, we incrementally build two separate sparse histograms $H_z(x, y, z)$ and $H_r(x, y, r)$ whose hash key is a

bitwise concatenation of cell locations,

$$\texttt{key}_m(x, y, m) = \lfloor x/q_{xy} \rfloor \lfloor y/q_{xy} \rfloor \lfloor m/q_m \rfloor \quad (5)$$

where q_{xy} is the corresponding Gaussian mixture map grid resolutions that will be analyzed in Section 7.2, $m \in \{z, r\}$, and q_m is the resolution set to the desired fidelity of z and r. The corresponding hash value is a histogram count that is gradually incremented as more points are added to the sparse histogram. In order to capture the variance of our LIDAR scanner and reduce discretization errors, we blur each point by incrementing neighboring histogram cells according to a Gaussian kernel with standard deviation of 5 cm. This helps us later account for measurement uncertainty in our likelihood evaluation.

Next, we perform weighted EM for each "column" in H_z and H_r to construct Gaussian mixture maps for z-height, $\mathcal{G}_z \leftarrow H_z$, and reflectivity, $\mathcal{G}_r \leftarrow H_r$. These two reductions are independent and can again be generalized as $\mathcal{G}_m \leftarrow H_m$ for clarity.

For a specific cell $(\hat{x}, \hat{y})$, we extract the corresponding histogram "column" of data,

$$E_m(\hat{x}, \hat{y}) = \left\{ \mathbf{e}_i^{(\hat{x},\hat{y})} \right\}_{i=1}^{f} = H_m(x = \hat{x}, y = \hat{y}, :), \quad (6)$$

where $\mathbf{e}_i^{(\hat{x},\hat{y})} = [c_i, m_i]^\top$ is the i^{th} histogram entry with count c_i for the cell centered at $(\hat{x}, \hat{y}, m_i)$; f is the number of cells observed in this histogram column.

We are then interested in condensing this column of data into a Gaussian mixture, $\mathcal{G}_m(x = \hat{x}, y = \hat{y}) = \left\{ \mathbf{g}_j^{(\hat{x},\hat{y})} \right\}_{j=1}^{g}$, where $\mathbf{g}_j^{(\hat{x},\hat{y})} = [w_j, \mu_j, \sigma_j]^\top$ is the j^{th} component of g Gaussians, parameterized by weight, mean, and standard deviation, respectively. This is achieved using the EM algorithm to iteratively estimate likelihood of Gaussian components given the data (**E**xpectation) and re-estimate new components that maximize this expected likelihood (**M**aximization). Histogram counts, c_i, are used to weight the expected likelihood derived in the expectation step.

Contrary to our previous work that iterated through multiple numbers of Gaussians, g, choosing the number of parameters that best fit the data while penalizing proportional to number of mixture components to avoid overfitting (Wolcott and Eustice, 2015), we found that this approach does not scale well with creating large maps and overcomplicates the online usage of the maps. We instead suggest a fixed number of Gaussians so that EM only needs to be run once per map cell. In the worst case, this results in redundant Gaussians summing to the same resulting likelihood distribution. We provide discussion and recommendations in Section 7.2.1 for how many Gaussians to choose and a visual depiction of what each map is capturing can be seen in Fig. 13, Fig. 14, and Fig. 15.

In this work, we present two methods for deriving the Gaussian mixture map over reflectivity. First, we consider the reflectivity of the entire point cloud resulting in $\mathcal{G}_r \leftarrow H_r$. Alternatively, we can use the 3D position of each point in space to extract the ground-plane only using a region growing method emanating from the known ground height around the vehicle; this results in a Gaussian mixture over ground surface reflectivities $\mathcal{G}_{r,\texttt{grd}} \leftarrow H_{r,\texttt{grd}}$. This is a general representation that is identical to the probabilistic maps presented by Levinson and Thrun (2010) when the number of Gaussian components is $g = 1$.

4.2 Registration Formulation

Given a point cloud, $\mathcal{P}$, we seek to find the optimal transformation that maximizes the likelihood of being drawn from the underlying Gaussian mixture maps, $\mathcal{G} = \{\mathcal{G}_z, \mathcal{G}_r\}$. This is directly formulated as the maximum likelihood estimate (MLE) to find the optimal alignment T*,

$$T^* = \underset{T}{\operatorname{argmax}} \, \mathcal{L}(\mathcal{P}|T, \mathcal{G}), \quad (7)$$

where $T = [x, y, z, \phi, \theta, \psi]^\top$ is a 6-DOF transformation that transforms points of $\mathcal{P}$ into $\mathcal{G}$.

The point cloud is made up of a set of n points, $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n$, where $\mathbf{p}_i = [x_i, y_i, z_i, r_i]^\top$ is the metric position and reflectivity of each point. We assume independence between points to arrive at

$$T^* = \underset{T}{\operatorname{argmax}} \prod_i \mathcal{L}(\mathbf{p}_i|T, \mathcal{G}). \quad (8)$$

We further assume independence between z-height of a point and its reflectivity, and use the chain rule over x_i and y_i,

$$\mathcal{L}(\mathbf{p}_i|T, \mathcal{G}) = \mathcal{L}(z_i|T, \mathcal{G}_z, x_i, y_i)\mathcal{L}(r_i|T, \mathcal{G}_r, x_i, y_i)\mathcal{L}(x_i, y_i). \quad (9)$$

We further marginalize out x_i and y_i, realizing that this distribution is fully captured by the blurring in our maps—thus, the corresponding distribution $\mathcal{L}(x_i, y_i)$ is already accounted for in the likelihood measure. This results in the joint likelihood over structure and appearance as

$$\mathcal{L}(\mathbf{p}_i|T, \mathcal{G}) = \mathcal{L}(z_i|T, \mathcal{G}_z, x_i, y_i)\mathcal{L}(r_i|T, \mathcal{G}_r, x_i, y_i). \quad (10)$$

These likelihoods are computed by first transforming the point $[x_i, y_i, z_i]^\top$ by T, resulting in $[x_i', y_i', z_i']^\top = T \oplus [x_i, y_i, z_i]^\top$. This allows us to compute each likelihood by indexing into the corresponding Gaussian mixture maps

and summing over the mixture components,

$$\mathcal{L}(z_i|T,\mathcal{G}_z,x_i,y_i) = \mathcal{L}(z_i'|\mathcal{G}_z(x_i',y_i')) = \sum_j \frac{w_{ij}^z}{\sqrt{2\pi{\sigma_{ij}^z}^2}} \exp\left(-\frac{(z_i'-\mu_{ij}^z)^2}{2{\sigma_{ij}^z}^2}\right), \tag{11}$$

and

$$\mathcal{L}(r_i|T,\mathcal{G}_r,x_i,y_i) = \mathcal{L}(r_i|\mathcal{G}_r(x_i',y_i')) = \sum_j \frac{w_{ij}^r}{\sqrt{2\pi{\sigma_{ij}^r}^2}} \exp\left(-\frac{(r_i-\mu_{ij}^r)^2}{2{\sigma_{ij}^r}^2}\right), \tag{12}$$

where w_{ij}^z, μ_{ij}^z, and σ_{ij}^z are the weight, mean, and standard deviation, respectively, of the j^{th} component of $\mathcal{G}_z(x_i', y_i')$ and w_{ij}^r, μ_{ij}^r, and σ_{ij}^r are the weight, mean, and standard deviation, respectively, of the j^{th} component of $\mathcal{G}_r(x_i', y_i')$.

However, we notice that these points may be drawn from the underlying Gaussian mixture maps *or* are obstacles drawn from a separate distribution. To limit the effect of outliers in our registration formulation, we modify (11) and (12) by mixing them with a uniform distribution. Specifically, the robust likelihoods take the form

$$\mathcal{L}'(z_i|T,\mathcal{G}_z,x_i,y_i) = \alpha\mathcal{L}(z_i|T,\mathcal{G}_z,x_i,y_i) + (1-\alpha)\mathcal{U}(z_i), \tag{13}$$

and

$$\mathcal{L}'(r_i|T,\mathcal{G}_r,x_i,y_i) = \beta\mathcal{L}(r_i|T,\mathcal{G}_r,x_i,y_i) + (1-\beta)\mathcal{U}(r_i), \tag{14}$$

where α and β are mixing parameters that control the region of influence of the underlying Gaussian mixture—having the effect of truncating the Gaussian distribution outside this region of influence. Further, the range of the uniform distributions are inconsequential, though are set to the range of the data ($[-100, 100]$ for z_i, and $[0, 255]$ for r_i). The resulting maximization including robust cost functions then looks like

$$T^* = \underset{T}{\operatorname{argmax}} \prod_i \mathcal{L}'(z_i|T,\mathcal{G}_z,x_i,y_i)\mathcal{L}'(r_i|T,\mathcal{G}_r,x_i,y_i). \tag{15}$$

The addition of the robust cost function is quite important for points in the roadway where Gaussian mixtures of the ground plane z-height typically have an extremely small variance. Without the robust formulation, these points would dominate the cost function and force the registration to overfit to outliers (e.g., obstacles). Moreover, we compute the log-likelihood for numerical stability, which allows us to compute a running sum independently as a parallel reduction.

Further, if the reflectivity Gaussian mixture map is modeled using the ground plane only ($\mathcal{G}_{r,\texttt{grd}}$ is used), then we only evaluate the reflectivity likelihood using ground points from $\mathcal{P}$. Online we use the same region growing method to extract the local ground plane; points not belonging to the ground plane will have a fixed likelihood that has no impact on the cost function. Note, however, that the z likelihood is still computed for all points in $\mathcal{P}$.

Algorithm 1 Full Registration

Input: GMM $\mathcal{G} = \{\mathcal{G}_z, \mathcal{G}_r\}$, Point Cloud $\mathcal{P}$, guess $T_0 = (x_0, y_0, z_0, \phi_0, \theta_0, \psi_0)$, search space X, Y, Ψ
Output: Optimal registration, $T^* = (x^*, y^*, z^*, \phi^*, \theta^*, \psi^*)$
1: $(\hat{x}, \hat{y}, z_0, \phi_0, \theta_0, \hat{\psi}) = \texttt{SEARCH}(x_0, y_0, z_0, \phi_0, \theta_0, \psi_0)$
2: $(x^*, y^*, z^*, \phi^*, \theta^*, \psi^*) = \texttt{HILL-CLIMB}\ (\hat{x}, \hat{y}, z_0, \phi_0, \theta_0, \hat{\psi})$

Algorithm 2 Exhaustive Search

Input: GMM $\mathcal{G} = \{\mathcal{G}_z, \mathcal{G}_r\}$, Point Cloud $\mathcal{P}$, guess $T_0 = (x_0, y_0, z_0, \phi_0, \theta_0, \psi_0)$, search space X, Y, Ψ
Output: Best 2D registration = $(\hat{x}, \hat{y}, \hat{\psi})$
$best = -\infty$
for ψ_i in $\psi_0 + \Psi$ **do**
  apply rotation ψ_i to $\mathcal{P}$
  for x_i, y_i in $\{x_0, y_0\} + XY$ **do**
    $likelihood = \mathcal{L}(\mathcal{P}|x_i, y_i, \mathcal{G})$ ▷ (15)
    if $likelihood > best$ **then**
      $best = likelihood$
      $(\hat{x}, \hat{y}, \hat{\psi}) = (x_i, y_i, \psi_i)$
    end if
  end for
end for

Considering now the optimization to find T^*, we make the observation that a typical wheeled-robotic platform is well constrained in $roll$, $pitch$, and $height$ because (i) most IMUs constrain $roll$ and $pitch$ to within a few degrees due to observation of the gravitational force (note that wheeled platforms only traverse minor roll/pitch) and (ii) any wheeled vehicle must be resting on the ground surface, which constrains $height$ with a prior map. Thus, (7) can be maximized by exhaustively searching over a range of x, y, and $heading$ transformations. As in Olson (2009), we can efficiently compute these by applying the $heading$ rotation to all points *first*, then evaluate at xy translations.

With our solution within the vicinity of the optimum, we then perform a simple, constrained 6-DOF hill-climbing to lock into the global optimum over our search space, T^*. This allows for the small, but necessary refinements of $height$, $roll$, and $pitch$. Because our registration problem is parameterized by the search boundaries, we are able to use pose priors to improve run-time performance. A detailed overview of registration into our Gaussian mixture map can be found in Algorithm 1 and Algorithm 2.

Algorithm 3 Multiresolution Search

Input: Base and Multires GMM $\mathcal{G} = \{\mathcal{G}_z, \mathcal{G}_r\}$, Point Cloud $\mathcal{P}$, guess $(x_0, y_0, z_0, \phi_0, \theta_0, \psi_0)$, search space X, Y, Ψ
Output: Best registration = $(\hat{x}, \hat{y}, \hat{\psi})$

// init. priority queue with search over coarse resolution
Initialize $PriorityQueue$ ▷ priority = log-likelihood
$coarsest = N$
$\mathcal{P}_{\texttt{rot}} = empty$ ▷ rotated point clouds
for ψ_i in $h_0 + \Psi$ **do**
  // store rotated clouds — do transformations once
  $T = f(0, 0, z_0, \phi_0, \theta_0, \psi_i)$ ▷ $[x, y]$ applied later
  $\mathcal{P}_{\texttt{rot}}[\psi_i] = T \oplus \mathcal{P}$
  for x_i in $x_0 + X/2^{coarsest}$ **do**
    for y_i in $y_0 + Y/2^{coarsest}$ **do**
      $cur.layer = coarsest$
      $cur.[x, y, \psi] = [x_i, y_i, \psi_i]$
      $cur.\mathcal{L} = \mathcal{L}(\mathcal{P}_{\texttt{rot}}[\psi_i] | x_i, y_i, \mathcal{G}[coarsest])$ ▷ (15)
      $PriorityQueue.add(cur)$
    end for
  end for
end for
// iterate priority queue, branching into finer resolutions
while $prev = PriorityQueue.pop()$ **do**
  if $prev.layer == 0$ **then**
    // at finest resolution, can't explore anymore
    // this is the global optimum
    $(\hat{x}, \hat{y}, \hat{h}) = prev.[x_i, y_i, \psi_i]$
    $return(\hat{x}, \hat{y}, \hat{\psi})$
  end if
  // branch into next finer resolution
  for x_i in $\left[prev.x, prev.x + 2^{prev.layer-1}\right]$ **do**
    for y_i in $\left[prev.y, prev.y + 2^{prev.layer-1}\right]$ **do**
      $cur.layer = prev.layer - 1$
      $cur.[x, y, \psi] = [x_i, y_i, prev.\psi]$
      $cur.\mathcal{L} = \mathcal{L}(\mathcal{P}_{\texttt{rot}}[prev.\psi] | x_i, y_i, \mathcal{G}[cur.layer])$ ▷ (15)
      $PriorityQueue.add(cur)$
    end for
  end for
end while

5 Multiresolution Branch-and-Bound

Typically, exhaustively searching for the maximum likelihood is not a realistic, tractable solution. In this section, we replace the expensive, exhaustive search with an efficient multiresolution branch-and-bound search.

5.1 Multiresolution Formulation

The idea behind our multiresolution search is to use a bounding function that can provide an upper-bound over a collection of cells in our reference map. This means that a majority of the search can be executed at a coarser resolution that upper-bounds the likelihood at finer scales. Using tight bounds can transform the exhaustive search presented in the previous section into a tractable search that makes *no* greedy assumptions. The branch-and-bound strategy achieves exactly the same result as the exhaustive search, only arrives at it in a more efficient manner.

For evaluating a single transformation (i.e., (T_x, T_y)), one must evaluate the log-likelihood of each point in a point cloud, then sum all of these for a total log-likelihood. Therefore in the exhaustive case, each point is evaluated against a single Gaussian mixture. In order to search a range of transformations, such as (T_x, T_y) to $(T_x + Nq_{xy}, T_y + Nq_{xy})$, each point is evaluated against a total of $(N + 1)^2$ Gaussian mixtures. However, each cell in our map is quite spatially similar, meaning that inference into (T_x, T_y) yields a similar log-likelihood as $(T_x + q_{xy}, T_y)$, so the exhaustive search can often spend unnecessary time in low-likelihood regions that can ideally be ruled out quicker.

We formulate a branch-and-bound search that exhaustively searches over the coarsest resolution providing upper-bounds over a range of transformations. These coarse search results are then added to a priority queue, ranked by upper-bound likelihoods. We then iterate through this priority queue, branch to evaluate the next finer resolution, and add back to the priority queue. The search is then complete once the finest resolution is returned from the priority queue.

We propose a slightly different multiresolution map structure than is traditionally considered. In many domains, multiresolution searches imply building coarser versions of your target data and making evaluations on that (e.g., the image pyramid). However, our approach creates many overlapping coarse blocks (as depicted in Fig. 4) to better compute tight upper-bounds. We could use the traditional multiresolution scheme, though our technique trades off for better bounds as opposed to a smaller memory footprint.

Because our maps are the same resolution throughout each multiresolution layer, this results in us taking larger *strides* through the coarser resolutions, where $stride = 2^{layer} \cdot q_{xy}$. Branching factor and number of multiresolution maps is completely user-defined. In our experiments, we opted for a branching factor of 2; that is, (T_x, T_y) branches into (T_x, T_y), $(T_x + stride, T_y)$, $(T_x, T_y + stride)$, and $(T_x + stride, T_y + stride)$ of the finer resolution map. Refer to Algorithm 3 and Fig. 4 for a more detailed overview.

5.2 Rasterized Gaussian Mixture Maps

Finding tight, *parametric* bounds for a collection of Gaussians is a rather difficult task, so we instead opt for a non-parametric solution in the form of rasterized lookup tables. We take our parametric Gaussian mixture map and compute a rasterized version by evaluating the log-likelihood at a fixed discretization, generating a rasterization for each grid cell. Upper bounds can then be *exactly* computed between neighboring grid cells by tak-

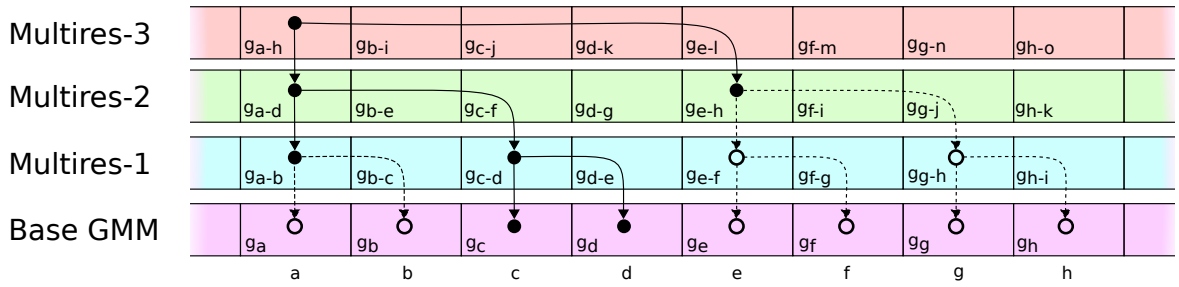


Fig. 4: A one-dimensional example of our multiresolution search formulation, where we demonstrate how a *single* 3D point would traverse through the multiresolution tree. Given some knowledge that the best transformation aligns the point somewhere within a-h, we begin the search at the coarsest resolution in cell a. Using branch-and-bound and computing upper-bounds over the Base GMM distribution in the multiresolution layers, we can efficiently search large spaces by avoiding low likelihood registrations (as depicted by dashed lines and open circles). In this figure, the notation $g_{a\text{-}h}$ refers to the fact that inference in that cell is an upper-bound over the distributions $g_a - g_h$, where g_x is the Gaussian mixture in cell x of the Base GMM. Note that contrary to several other multiresolution approaches, coarser resolutions in our framework *do not* imply a coarser resolution map. We maintain uniform resolution by using many overlapping coarse blocks to facilitate tighter upper-bounds; a truly coarser map in the Multires-1 layer would combine the first two cell into a single $g_{a\text{-}c}$ cell.

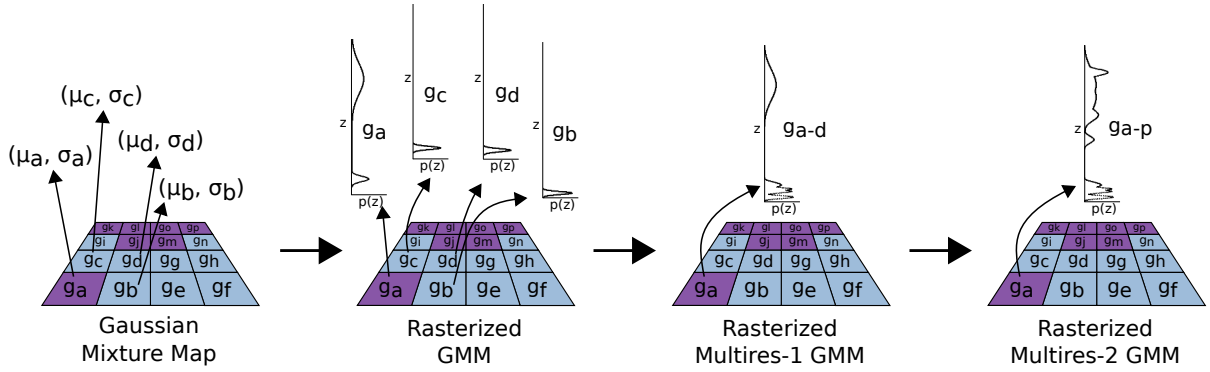


Fig. 5: Demonstration of the rasterization performed on the original Gaussian mixture map to facilitate exact upper-bounds. We begin with a parametric 2D map that encodes a Gaussian mixture in each cell, where the grid is colored by the difference between the two Gaussian modes in the cell, *blue* indicates 2 overlapping mixture components around the ground-plane and *purple* indicates two distinct modes captured including ground-plane and superstructure. We then rasterize each cell (note we display the likelihood, not log-likelihood for clarity); these rasterized representations can then be used to create rasterized upper-bounds for multiresolution search. The first step of this evaluates the upper-bound at each discretization by taking the `max` of the underlying cell rasterizations. Note that as you continue to move to coarser resolutions the distribution generalizes quite well—data for this figure was generated from looking at the edge of a tree, where the multiresolution map can capture the two common modes of tree limbs and ground-plane. In this figure, the notation $g_{a\text{-}d}$ means the rasterization is an upper-bound over the $g_a - g_d$ rasterizations.

ing the `max` across each discretization in the rasterized lookup table. While localizing, likelihoods at the finest resolution are computed using the original Gaussian mixture maps and the rasterized maps only facilitate fast traversal through the search tree. See Fig. 5 for a visual representation of these maps.

For a pure localization task such as ours, lookup tables can be pre-computed offline. However, we decided to store *only* the parametrized Gaussian mixture maps on disk to avoid storing extremely large maps. We are then able to efficiently compute rasterized multiresolution maps online from our parameterized Gaussian mixture map as a background job. This is done incrementally using each successive multiresolution layer to build the next.

Note that our rasterized multiresolution maps are a generic representation that can also be used with other map types including standard NDT, MLS maps, occupancy voxels, etc. After converting one of these maps to a rasterized multiresolution map, the remainder of our proposed pipeline can be used for fast registration of a point cloud.

A sample search through our multiresolution search space can be seen in Fig. 6. The shown example explores a $25\ \mathrm{m} \times 25\ \mathrm{m}$ area at $16\ \mathrm{cm}$ resolution in approximately 2 seconds, while only needing to evaluate 1% of the transformations necessary in the exhaustive search.

6 Localization Filter

Our localization task is framed as an estimation problem over the full 6-DOF dynamics of our vehicle, where our state vector is $\boldsymbol{\mu}_k = \{x_k, y_k, z_k, \phi_k, \theta_k, \psi_k\}$. We propose to use an extended Kalman filter (EKF) to estimate our

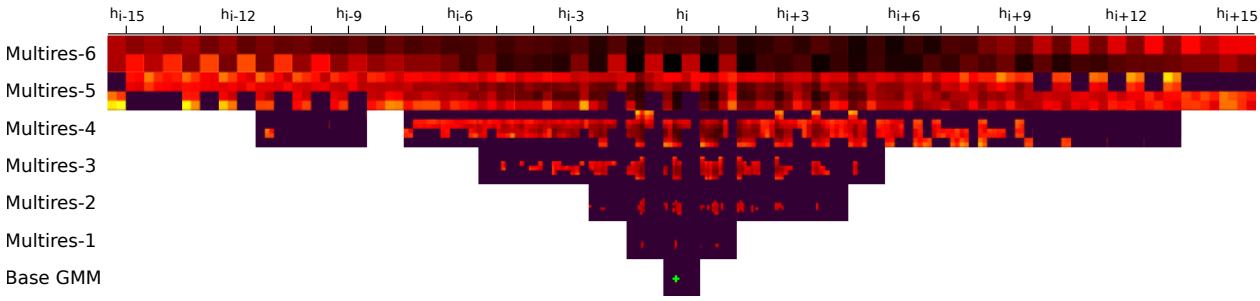


Fig. 6: Sample multiresolution search space traversal. Top-bottom represents coarse-to-fine searching, left-right represents different slices through our *heading* search, and each pixel depicts an xy translation searched. Log-likehoods are colored increasingly yellow-black, purple and non-existent cells are areas not needed to be explored by the multiresolution search, and the optimal is indicated in green. We exhaustively search the coarsest resolution, then use branch-and-bound to direct our traversal through the tree. For typical scan alignments, we only have to search approximately 1% of the transformations in the finer resolutions, doing a majority of the work in the coarser resolutions.

vehicle state from various measurement sources. An EKF provides a simple way to fuse information from multiple sensing modalities; though in this section we only consider integrating our vehicle odometry and multiresolution registrations into our EKF estimation.

We define a discrete time process model and incorporate our registration corrections into our state filter:

$$\begin{aligned}
\text{Predict} \quad & \bar{\boldsymbol{\mu}}_k = f(\boldsymbol{\mu}_{k-1}, \mathbf{u}_k) \\
& \bar{\Sigma}_k = \mathrm{F}_k \Sigma_{k-1} \mathrm{F}_k^\top + \mathrm{Q}_k \\
\\
\text{Update} \quad & \mathrm{K}_k = \bar{\Sigma}_k \mathrm{H}_k^\top (\mathrm{H}_k \bar{\Sigma}_k \mathrm{H}_k^\top + \mathrm{R}_k)^{-1} \\
& \boldsymbol{\mu}_k = \bar{\boldsymbol{\mu}}_k + \mathrm{K}_k \big(\mathbf{z}_k - \mathbf{h}_k(\bar{\boldsymbol{\mu}}_k)\big) \\
& \Sigma_k = (\mathrm{I} - \mathrm{K}_k \mathrm{H}_k) \bar{\Sigma}_k (\mathrm{I} - \mathrm{K}_k \mathrm{H}_k)^\top + \mathrm{K}_k \mathrm{R}_k \mathrm{K}_k^\top
\end{aligned}$$

Here, $f(\cdot)$ is our nonlinear plant model that integrates odometry measurements from an Applanix IMU, $\mathbf{u}_k$, with uncertainty Q_k and linearized Jacobian F_k. H_k is a linear observation model (identity matrix) and K_k is the corrective Kalman gain induced by our registration measurement with uncertainty R_k. The measurement $\mathbf{z}_k$ is exactly the output of our multiresolution registration detailed in Section 5, $\mathbf{z}_k = T^*$. We use fixed measurement uncertainties, R_k, that were empirically determined (discussed in Section 7.4); however, one could fit a conservative covariance using the explored search space as shown by Olson (2009).

One issue with this formulation is that when measurements arrive there is a non-zero latency associated with *registering* the measurements. This presents a problem when we attempt to add the measurement to our EKF because the measurement would be applied to the future state of the robot. Thus, as soon as the measurement is received (near zero latency), we augment our EKF with a delayed-state (Leonard and Rikoski, 2000). In our discrete model, this leads to an expanded state belief as:

$$\mathbf{x}_k = \left[\boldsymbol{\mu}_k^\top, \boldsymbol{\mu}_{k-1}^\top\right]^\top.$$

Thus, we can continue to apply odometry prediction directly to the most recent state, but apply the registration correction on the associated delayed-state. The correlation inherent between temporal poses will allow the effect of this measurement to then propagate to the current state belief. We then marginalize this delayed-state as it is no longer necessary to maintain in our state vector. This formulation also facilitates the integration of more measurements from various sources to increase robustness, despite the fact that these sources can have varying latencies associated with them.

Our filter is initialized in a global frame from a single dual-antenna GPS measurement with high uncertainty, which provides a rough initial guess of global pose with orientation. We adaptively update our multiresolution search bounds to ensure that we explore a 4σ window around our posterior distribution. This dynamic approach allows us to improve performance as our posterior confidence increases, while leaving room to statistically eliminate outlier measurements by evaluating the corresponding measurement normalized innovation squared (NIS). Note that aside from using GPS for initializing the filter, our proposed localization method *only* uses input from inertial sensors, a wheel encoder, and 3D LIDAR scanners.

7 Evaluation

In this section, we present a thorough evaluation of our proposed theory covering a diverse set of real-world experiments. All algorithms were implemented in C/C++ using CUDA and, unless otherwise specified, experiments were run on a workstation computer equipped with an Intel Xeon E5-2670 CPU and an NVIDIA GeForce GTX TITAN X GPU. For parallelization in CUDA, we parallelized the inner-loop calculation of registration likelihood (implemented as a parallel sum reduction).

7.1 Platforms and Datasets

We evaluate our proposed methods using data collected on our autonomous platforms, a TORC ByWire XGV

(a) TORC ByWire XGV

(b) Ford Fusion Hybrid Autonomous Research Vehicle

Fig. 7: Test platforms used for evaluation of multiresolution Gaussian mixture map localization: a TORC ByWire XGV and a Ford Fusion Hybrid Autonomous Research Vehicle. Both platforms are equipped with 4 Velodyne HDL-32E LIDAR scanners and an Applanix POS-LV 420 INS.

(Fig. 7(a)) and a Ford Fusion Hybrid Autonomous Research Vehicle (Fig. 7(b)). These automated vehicles are equipped with four Velodyne HDL-32E 3D LIDAR scanners and an Applanix POS-LV 420 IMU. Given the use of four independent LIDAR scanners, it is crucial to perform extrinsic calibration between these to establish a rigid body transformation between the IMU and each sensor. This is achieved by formulating a pose-graph as in Section 3 and treating the calibration parameters as unknowns in the optimization.

Moreover, reflectivity measurements need to be calibrated against a known reference map such that measurements are consistent (i) within each LIDAR scanner, (ii) between LIDAR scanners on the same platform, and (iii) between platforms, as mapping data must be consistent with data observed online. To achieve this, we use a method similar to that proposed by Levinson and Thrun (2014), which derives a map from observed reflectivity to true reflectivity. In their work, each beam rotates about an axis perpendicular to the ground-plane, which allows for learning a mapping $observed\ reflectivity \rightarrow real\ reflectivity$ to implicitly account for angle of incidence. However, our rotation axes are *not* perpendicular to the ground-plane, so we must alter their approach by learning the mapping $\{observed\ reflectivity, rotation\} \rightarrow real\ reflectivity$.

Experiments are presented on two primary datasets collected with our platforms:

- *PG14 Dataset*: Set of 14 logs collected over 3 months with the TORC and Fusion platforms, each log approximately $38\ \mathrm{km}$ in length covering a loop near Ann Arbor, Michigan, over **P**lymouth Road, **G**otfredson Road, and M-**14** highway. This dataset totals $525.72\ \mathrm{km}$ in length, covering common use cases including highway, rural, and residential areas at various times of day, including rush hour. Furthermore, there were 3 construction zones that evolved over the data collection; 2 of which resulted *full repavings* approximately $0.5\ \mathrm{km}$ each and the third was for sewage repairs that did not radically alter appearance aside from large amounts of construction signage. See Fig. 8 for a visual depiction of the route and an overview of each log.
- *Downtown Dataset*: Set of 5 logs collected with the TORC platform, each spanning a $3\ \mathrm{km}$ loop through downtown Ann Arbor, Michigan. This dataset totals $14.92\ \mathrm{km}$ of urban roadways and one of these logs was collected on a snowy day with snow actively falling and covering the ground, as depicted in Fig. 18(a). See Fig. 9 for a visual depiction of the route and an overview of each log.

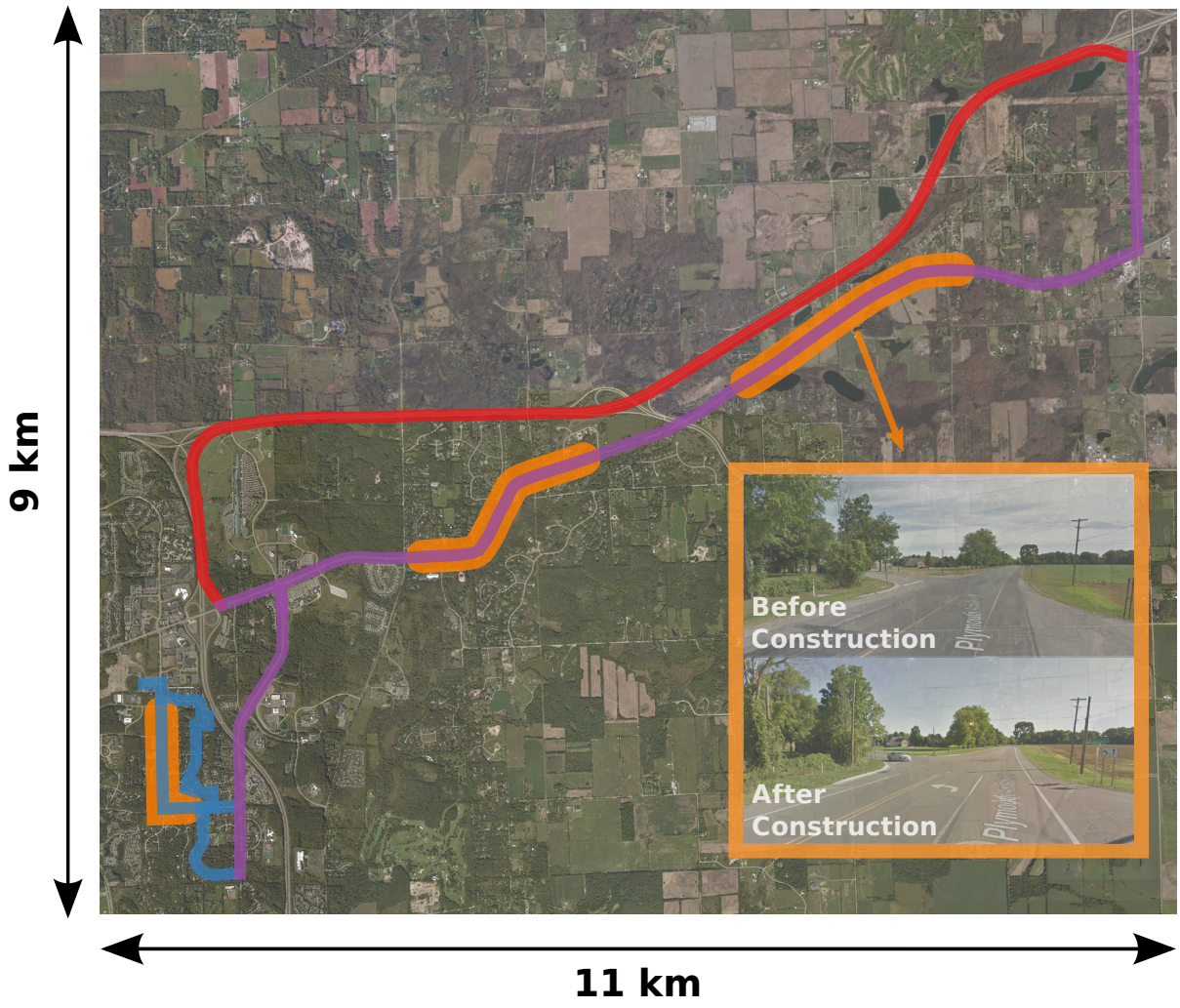
Collectively, these datasets cover samplings through various environments including highway, urban, rural, and residential roadways during various conditions including heavy traffic, heavy snowfall, and construction zones. Further, these datasets traverse road grades ranging from -5% to 5%.

In each of these datasets, the first 2 logs are used for map construction; this helps mitigate the effect of stationary dynamic obstacles (e.g., parked cars), which we do not explicitly remove, but allow the EM algorithm to blur away their impact. These two mapping logs are stitched together into a single pose-graph, then all data for map construction is compiled at the sparse histogram level, as detailed in Section 4.1. Each subsequent log was merged into this pose-graph to provide experimental ground-truth for each—we held the poses from the mapping logs fixed such that scan registrations in the SLAM pipeline are implemented as measurement *priors*. We assume the accuracy of this ground-truth is an order of magnitude better than our localization errors; this was verified by reprojecting LIDAR data into this common *map* reference frame and visually comparing the reconstructed surface-reflectivity for consistency.

7.2 Map Parameter Selection

This section intends to analyze the impacts of varying map parameters, both in terms of resulting localization errors as well as disk space requirements for storing maps. The primary user configurable map settings include (i) grid resolution, (ii) number of Gaussians per cell, and (iii) reflectivity maps that contain full 3D appearance ($\mathcal{G}_r$) or appearance of the ground-plane only ($\mathcal{G}_{r,\mathrm{grd}}$). Moreover, these first two settings can be tuned for each map type (structure vs. appearance). All of these variabilities lead to a wide search space of possible map combinations.

To fully experiment and determine the optimal map parameters, we constructed 360 maps to run evaluation over (180 using the *PG14 Dataset* and 180 using the *Downtown Dataset*). Each of these sets of 180 maps were made by varying 12 grid resolutions ($6.4\ \mathrm{cm}$, $8.0\ \mathrm{cm}$, $12.8\ \mathrm{cm}$, $16.0\ \mathrm{cm}$, $25.6\ \mathrm{cm}$, $32.0\ \mathrm{cm}$, $51.2\ \mathrm{cm}$, $64.0\ \mathrm{cm}$, $80.0\ \mathrm{cm}$, $128.0\ \mathrm{cm}$, $160.0\ \mathrm{cm}$, and $256.0\ \mathrm{cm}$), 5 different number



Ann Arbor—PG14							
ID	Date	Platform	Length	ID	Date	Platform	Length
P-M1	May 1, 2015	TORC	38.0 km	P-M2	May 1, 2015	Fusion	37.9 km
P-1	May 8, 2015	TORC	37.9 km	P-2	May 8, 2015	Fusion	33.4 km
P-3	May 15, 2015	TORC	37.8 km	P-4	May 15, 2015	Fusion	37.9 km
P-5	May 21, 2015	TORC	37.8 km	P-6	May 21, 2015	Fusion	37.9 km
P-7	June 5, 2015	TORC	37.9 km	P-8	June 5, 2015	Fusion	37.9 km
P-9	June 12, 2015	TORC	37.8 km	P-10	June 12, 2015	Fusion	37.9 km
P-11	July 24, 2015	TORC	37.9 km	P-12	July 24, 2015	Fusion	37.9 km
							Total: 525.72 km

Fig. 8: *PG14 Dataset*: a dataset of 14 manually driven loops near Ann Arbor, Michigan, covering **P**lymouth Road, **G**otfredson Road, and M-**14** highway. This dataset was collected over a span of 3 months and covers residential roads (*blue*), rural roads (*purple*), and a highway (*red*). Moreover, we observed 3 construction zones over data acquisition (*orange*). The two zones on the right resulted in full repavings that included the addition of left turn lanes that were not completed until P-11 and P-12, as can be seen in the lower-right corner (images courtesy of Google Maps). In our evaluation with this dataset, P-M1 and P-M2 were used for map construction.

Ann Arbor—Downtown			
ID	Date	Platform	Length
D-M1	Nov. 19, 2013	TORC	3.0 km
D-M2	Nov. 19, 2013	TORC	3.0 km
D-1	Nov. 20, 2013	TORC	3.0 km
D-2	Nov. 20, 2013	TORC	3.0 km
D-3	Dec. 17, 2013	TORC	3.0 km
			Total: 14.92 km

Fig. 9: *Downtown Dataset*: a dataset of 5 manually driven loops through downtown Ann Arbor, Michigan, covering urban driving scenarios. D-3 was collected while heavy snow was falling and covered significant portions of the roadway. In our evaluation with this dataset, D-M1 and D-M2 were used for map construction.

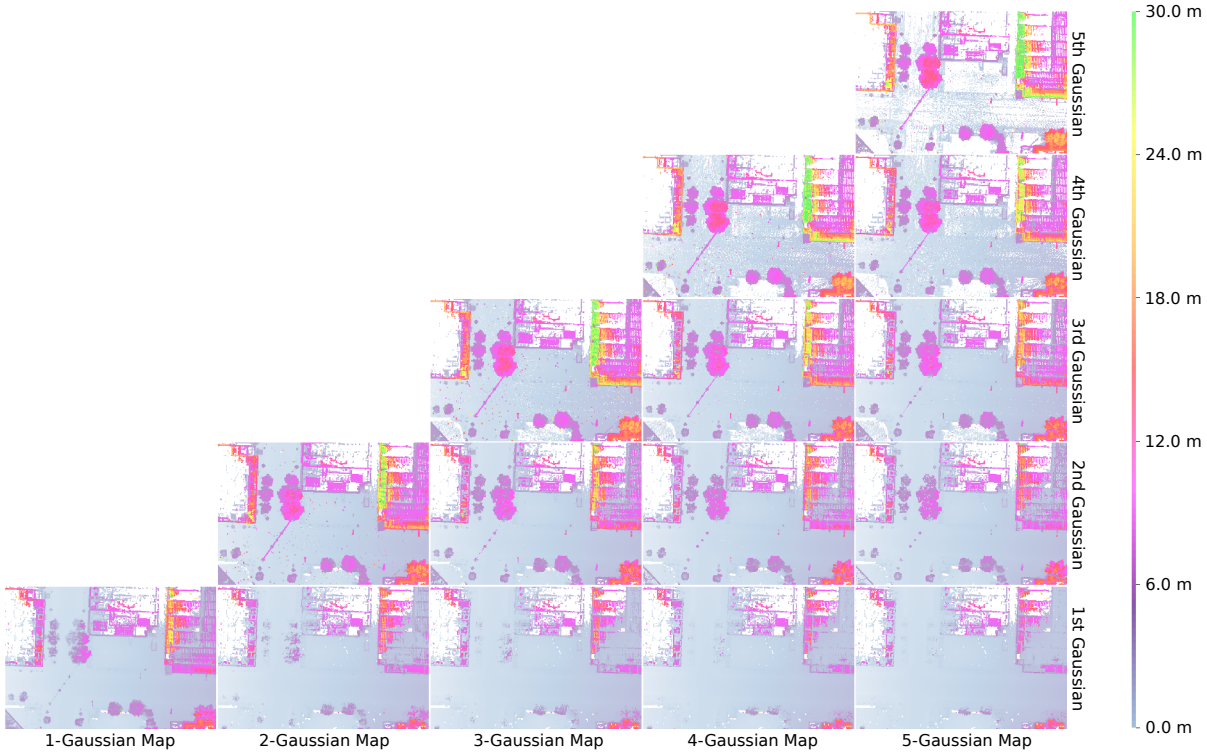


Fig. 10: Gaussian mixture map components contained within the z-height map, $\mathcal{G}_z$, capturing ground-plane, buildings, trees, lightposts, and traffic lights. Left-to-right, each column in this figure represents a different map, ranging from a Gaussian mixture map containing only 1 component per cell to one containing 5 components per cell. Bottom-to-top, we display the mean of the i^{th} Gaussian component, ordered by increasing component mean (z-height); white cells indicate no Gaussian mixture component exists because no data was available during mapping or the Gaussian mixture resulting weight was less than 0.001.

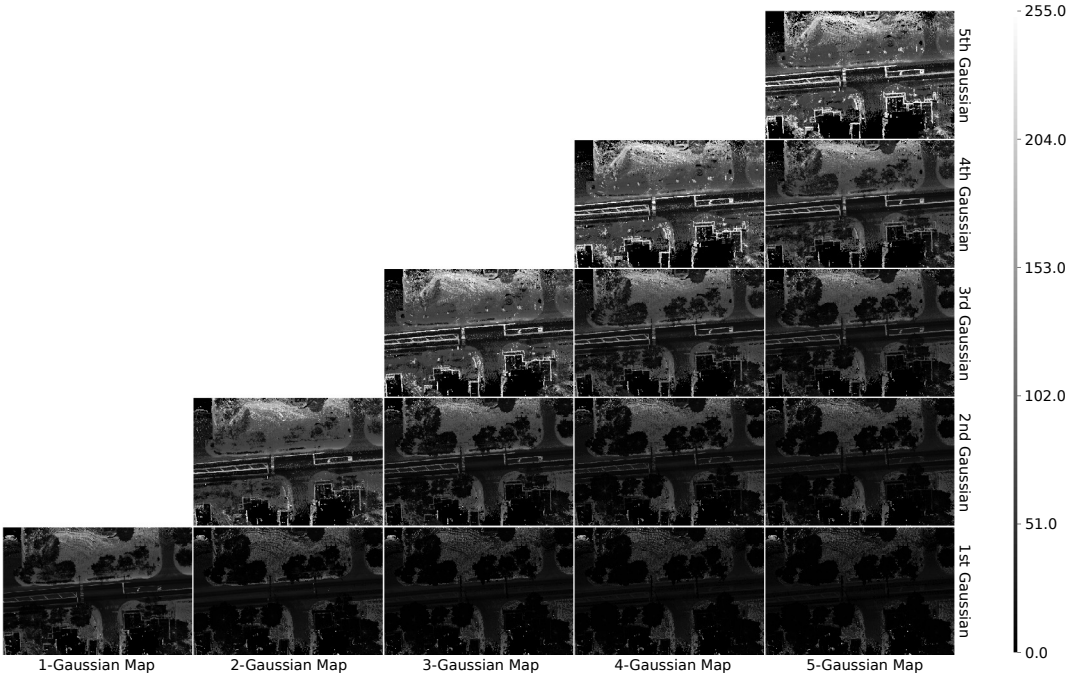


Fig. 11: Gaussian mixture map components contained within the reflectivity map, $\mathcal{G}_r$, capturing appearance of ground-plane, foliage, etc. Left-to-right, each column in this figure represents a different map, ranging from a Gaussian mixture map containing only 1 component per cell to one containing 5 components per cell. Bottom-to-top, we display the mean of the i^{th} Gaussian component, ordered by increasing component mean (reflectivity).

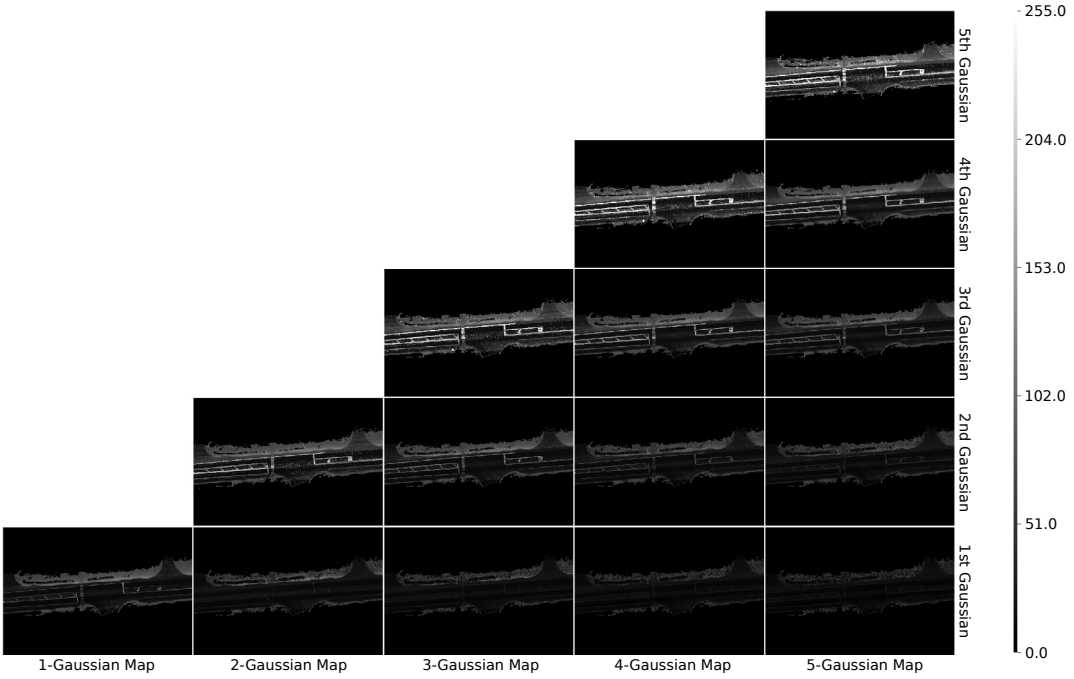


Fig. 12: Gaussian mixture map components contained within the reflectivity map, $\mathcal{G}_{r,\texttt{grd}}$, capturing appearance of ground-plane *only*. See Fig. 11 for more details.

of Gaussians per grid cell (1–5), and were generated for our 3 map types, ($\mathcal{G}_z$, $\mathcal{G}_r$, and $\mathcal{G}_{r,\mathtt{grd}}$).

In our implementation, we store our maps on disk in $64\ \mathrm{m} \times 64\ \mathrm{m}$ tiles. Thus, grid resolutions considered here were chosen to evenly divide these tiles. Moreover, to efficiently generate hundreds of maps, we construct our sparse histogram map representations at the $6.4\ \mathrm{cm}$ and $8.0\ \mathrm{cm}$ resolutions only, then build the remaining maps using these histograms (e.g., the $25.6\ \mathrm{cm}$ map is constructed by pooling a 4×4 window of the $6.4\ \mathrm{cm}$ histogram).

As a first reference to qualitatively demonstrate what is being captured in our maps, we looked at 5 versions of $\mathcal{G}_z$, $\mathcal{G}_r$, and $\mathcal{G}_{r,\mathtt{grd}}$, at a resolution of $25.6\ \mathrm{cm}$, in which we varied the number of Gaussians per cell from 1–5. Snapshots of these maps are visually depicted in Fig. 10, Fig. 11, and Fig. 12, respectively.

Within the z-height map, Fig. 10, we immediately see that increasing the number of Gaussians leads to overfitting. This is clear in the ground-plane and superstructure where many components share the same mean. There is a significant qualitative improvement from a 1-Gaussian map to a 2-Gaussian map. In the 1-Gaussian case, we see that there's a necessary blurring between ground and superstructure (trees, lightposts, etc.), while the 2-Gaussian case can easily capture a mode near the ground-plane and a mode covering superstructure. This trend continues through higher fidelity maps as we see the ground is captured in the lowest mean component and the increase in number of Gaussians allows for more overfitting to building facades and other superstructure. Keeping these modes separate and distinct is important for discarding obstacles that may appear in the void between ground and structure, thus we expect there to be a noticeable localization improvement between the 1-Gaussian and higher number maps.

Looking at the reflectivity maps, Fig. 11 and Fig. 12, we again notice overfitting beyond 2 Gaussians. Using 2 or more Gaussians is necessary for $\mathcal{G}_r$, as there appears to be two distinct modes per cell: the appearance of the ground and the appearance of above ground features. In both maps, multiple Gaussian components allows us to better capture edge effects (transitions from asphalt to road paint), where ground-plane road paint is typically much smaller than the $25.6\ \mathrm{cm}$ grid resolution. The state-of-the-art method (Levinson and Thrun, 2010), depicted as the 1-Gaussian ground only map in Fig. 12, leads to a more washed out image as these edges blur between high and low reflectivity—thus capturing a large Gaussian variance in these cells.

7.2.1 Map Parameter Sweep

We performed a series of evaluation over these 360 maps constructed in which we hold the experimental log fixed while evaluating against each map. Thus, we used P-3 to test against *PG14 Dataset* maps and D-2 to evaluate against *Downtown Dataset* maps. Further, we test using each map type independently ($\mathcal{G}_z$, $\mathcal{G}_r$, $\mathcal{G}_{r,\mathtt{grd}}$), assuming that the resulting combination of the structure and appearance maps will yield more robust measurements, without directly optimizing over the exorbitant number of cross possibilities between all map types.

To benchmark our registration quality we took known ground-truth for our evaluation log and generated a randomized offset every $40\ \mathrm{m}$ of road travel. This random offset was sampled uniformly within $2.5\ \mathrm{m} \times 2.5\ \mathrm{m}$ of the ground-truth pose. This randomly sampled point can then be viewed as the initial guess, T_0, into our Gaussian mixture map registration framework. The expectation is that the resulting registration event will converge on the ground-truth pose.

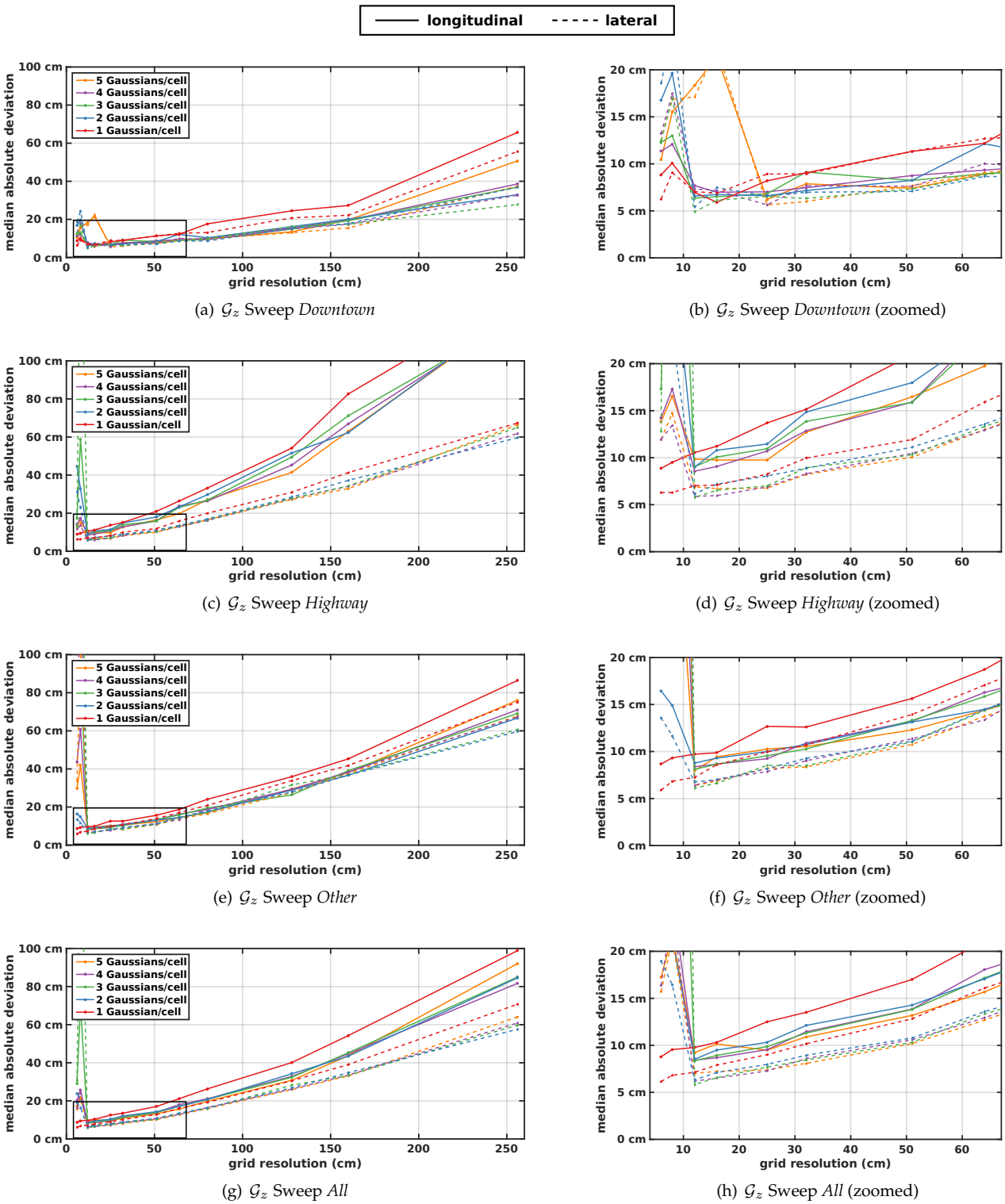
Results for the parameter sweep over $\mathcal{G}_z$, $\mathcal{G}_r$, and $\mathcal{G}_{r,\mathtt{grd}}$ are presented in Fig. 13, Fig. 14, and Fig. 15, respectively. These figures show the longitudinal and lateral median absolute deviation with respect to ground-truth, and results are divided between *downtown*, *highway*, and *other* (encompassing rural and residential roads) portions, along with a summary over *all* roadways. We chose to evaluate in terms of longitudinal and lateral errors as these are most relatable given the context of on-road navigation, and median absolute deviation was chose over other statistics as it is a robust, outlier-proof measure of variability.

As expected, we see that error grows as a function of coarser grid resolution across all map types, and we are constrained far better laterally than we are longitudinally. Furthermore, we see that all map types perform markedly better on the *downtown* portions, as can be expected given the significant structure and well maintained road paint. Across most of these plots, we notice that there are significant error spikes at very fine resolutions. This is believed to be caused by overfitting as there is simply not enough training data to accurately learn the Gaussian mixture maps. Additionally, large grid sizes increase the basin of convergence for our registration optimization; features at the fine resolution may be skipped over through the grid search.

Looking at the evaluation over $\mathcal{G}_z$, we see a noticeable improvement from 1 to 2+ Gaussians in both lateral and longitudinal error. As predicted in the previous section, the ability to rule out obstacles between superstructure and ground-plane plays an important role here.

Considering the evaluation over $\mathcal{G}_r$, it is clear that 2+ Gaussians is necessary as anticipated—given that 2 modes are needed to capture ground-plane appearance and above-ground appearance. On the contrary, $\mathcal{G}_{r,\mathtt{grd}}$ performs best with a single Gaussian, though only at fine resolutions. As grid cell size is increased, it is necessary to use 2+ Gaussians so that features are not blurred away.

Throughout all of these sweeps, it is not immediately



(a) $\mathcal{G}_z$ Sweep *Downtown*

(b) $\mathcal{G}_z$ Sweep *Downtown* (zoomed)

(c) $\mathcal{G}_z$ Sweep *Highway*

(d) $\mathcal{G}_z$ Sweep *Highway* (zoomed)

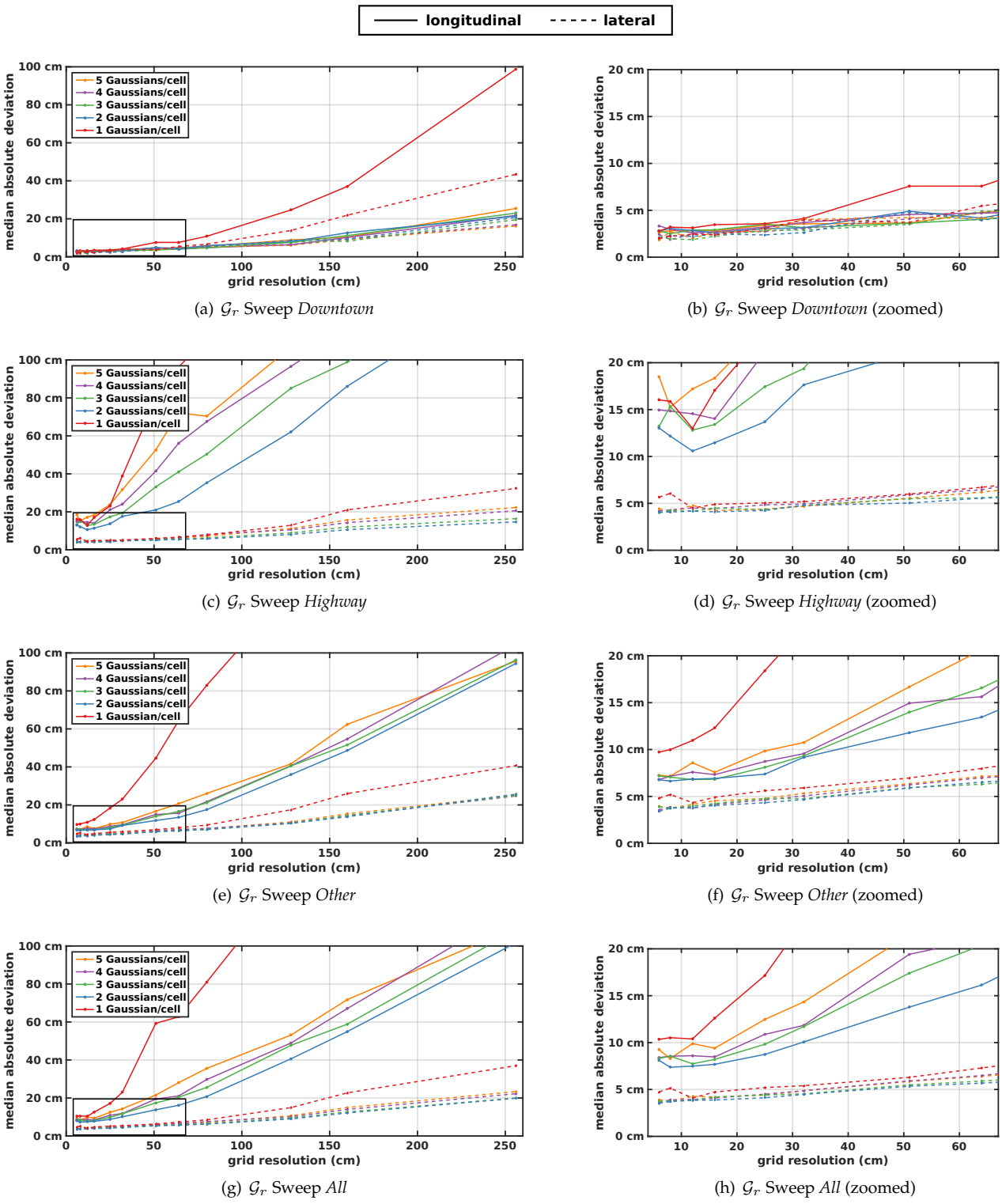
(e) $\mathcal{G}_z$ Sweep *Other*

(f) $\mathcal{G}_z$ Sweep *Other* (zoomed)

(g) $\mathcal{G}_z$ Sweep *All*

(h) $\mathcal{G}_z$ Sweep *All* (zoomed)

Fig. 13: Summary of parameter sweep over grid resolution and number Gaussians in z-height Gaussian mixture maps, $\mathcal{G}_z$, and the resulting longitudinal (solid lines) and lateral (dashed lines) median absolute deviation (MAD)—partitioned into *downtown*, *highway*, *other* (rural and residential), and a summary over *all* data.



(a) $\mathcal{G}_r$ Sweep *Downtown*

(b) $\mathcal{G}_r$ Sweep *Downtown* (zoomed)

(c) $\mathcal{G}_r$ Sweep *Highway*

(d) $\mathcal{G}_r$ Sweep *Highway* (zoomed)

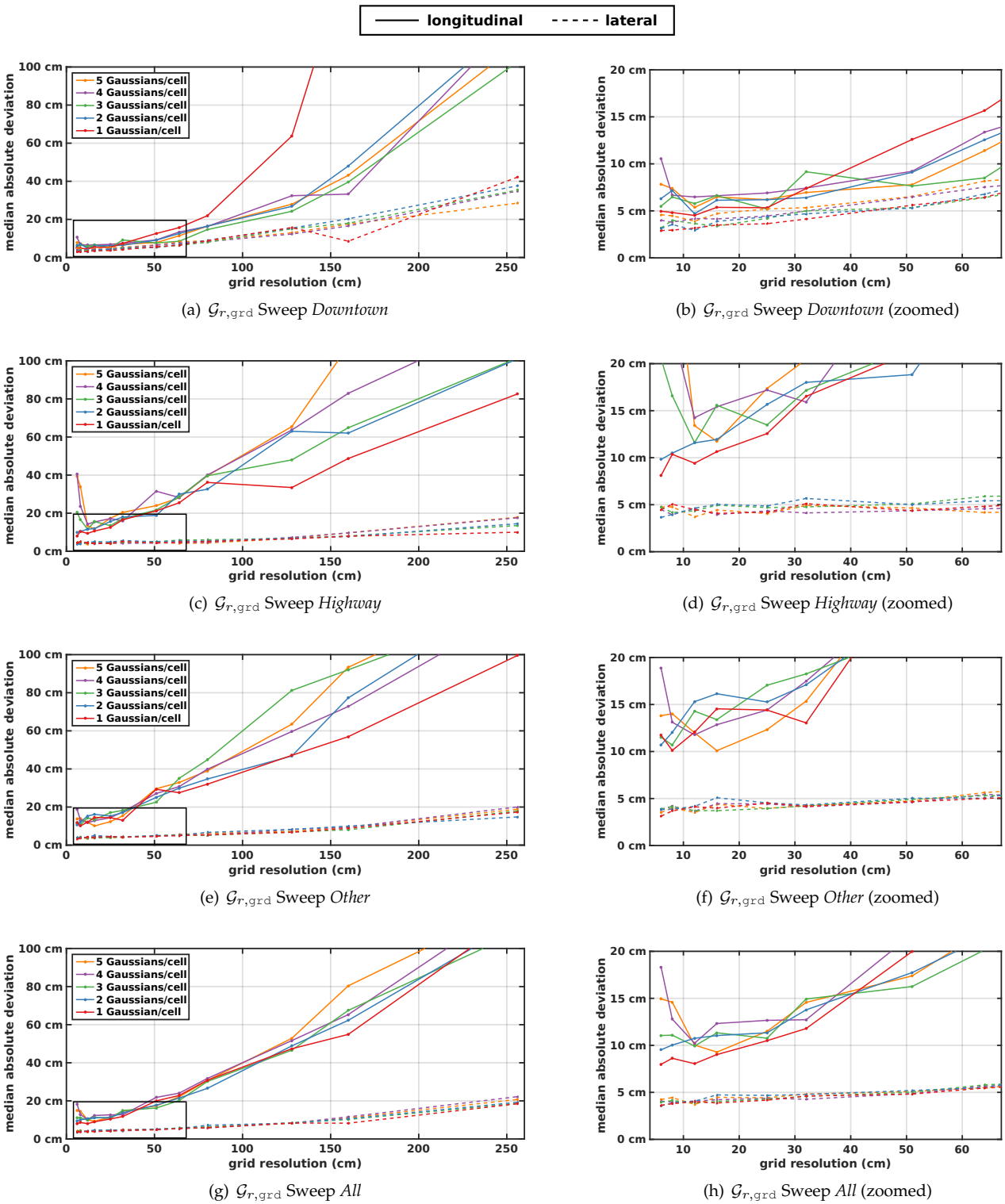
(e) $\mathcal{G}_r$ Sweep *Other*

(f) $\mathcal{G}_r$ Sweep *Other* (zoomed)

(g) $\mathcal{G}_r$ Sweep *All*

(h) $\mathcal{G}_r$ Sweep *All* (zoomed)

Fig. 14: Summary of parameter sweep over grid resolution and number Gaussians in reflectivity Gaussian mixture maps, $\mathcal{G}_r$, and the resulting longitudinal (solid lines) and lateral (dashed lines) median absolute deviation (MAD)—partitioned into *downtown*, *highway*, *other* (rural and residential), and a summary over *all* data.



(a) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *Downtown*

(b) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *Downtown* (zoomed)

(c) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *Highway*

(d) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *Highway* (zoomed)

(e) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *Other*

(f) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *Other* (zoomed)

(g) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *All*

(h) $\mathcal{G}_{r,\texttt{grd}}$ Sweep *All* (zoomed)

Fig. 15: Summary of parameter sweep over grid resolution and number Gaussians in ground-plane only, reflectivity Gaussian mixture maps, $\mathcal{G}_{r,\texttt{grd}}$, and the resulting longitudinal (solid lines) and lateral (dashed lines) median absolute deviation (MAD)—partitioned into *downtown*, *highway*, *other* (rural and residential), and a summary over *all* data.

clear that more than 2 Gaussians is necessary as there is not a significant performance improvement by doing so.

Map Size: In addition to performance metrics, in many cases map parameter selection must also consider the required disk space for map storage. In Fig. 16, we look at the corresponding disk space required per km of map data. As expected, finer resolution maps get exponentially larger relative to coarser grid resolutions. Additionally, ground-only reflectivity maps are significantly smaller than those constructed using all points—this is because ground-only maps are restricted to areas within a few meters of the roadway, while full maps can include points over 50 meters away.

For our maps, all Gaussian mixture components (weights, means, variances) are stored as single-precision floating point values, and the maps are compressed using gzip. Therefore, more intricate compression schemes can be used and map sizes presented here should be viewed as a worst case scenario.

We envision that our maps can be streamed to an autonomous car, where our $64\ \mathrm{m} \times 64\ \mathrm{m}$ tiles can be continuously downloaded over a 4G connection. We assume a network bandwidth of nominally $2\ \mathrm{MBps}$ and a vehicle certainly traveling less than $150\ \mathrm{kph}$. Thus, we have an available streaming budget of roughly $48.0\ \mathrm{MB/km}$.

Considering this budget, we decided on a 2-Gaussian, $25.6\ \mathrm{cm}$ z-height map ($\mathcal{G}_z$), and a 1-Gaussian, $6.4\ \mathrm{cm}$ ground-only reflectivity map ($\mathcal{G}_{r,\mathtt{grd}}$). Note that our reflectivity map selection is roughly the same as Levinson's probabilistic appearance maps. The combination of these two maps found the best balance between performance, while falling under our required budget at roughly $44.3\ \mathrm{MB/km}$. We found that the superior performance of the ground-only reflectivity maps on the highway are an added benefit as z-height is least effective there.

The remainder of our evaluation will perform experiments over this map configuration alone.

7.3 Registration Experiments

Since our odometry source has significantly low drift-rates, registration deficiencies can be masked by a well-tuned filtering framework. Thus, this section looks directly at evaluating the unfiltered registrations that exploit structure and appearance within the vicinity of ground-truth results.

Identical in setup to our parameter sweep discussed in the previous section, we now look at a sweep over all logs in our dataset while holding map settings fixed—showing that our map is robust for localizing over time. Again we randomly sample a point uniformly within $2.5\ \mathrm{m} \times 2.5\ \mathrm{m}$ of ground-truth every $40\ \mathrm{m}$, and evaluated the resulting transformation from our multiresolution registration framework relative to this ground-truth. To fully understand the contributions of each map type, we perform 3 registrations per ground-truth sample: (i) using structure alone ($\mathcal{G}_z$), (ii) using appearance alone ($\mathcal{G}_{r,\mathtt{grd}}$), and (iii) using structure and appearance jointly ($\mathcal{G}_z, \mathcal{G}_{r,\mathtt{grd}}$).

Errors are summarized per data log in Fig. 17, where we show median absolute deviation bars for longitudinal and lateral errors, along with first and third quartile error whiskers. Over most of *PG14 Dataset*, we see high longitudinal errors when using reflectivity alone that becomes well constrained with the addition of 3D structure. In most logs, we see that the joint cost function yields an improvement in our registrations. However, in the case of the *Downtown Dataset*, it is not surprising that the joint cost function is heavily dictated by z (seeming to ignore the more accurate reflectivity measurements) because the significant number of point returns off of 3D structure that dominate the cost function.

Results are summarized for all datasets in Table 1.

7.3.1 Heavy Snowfall Registrations

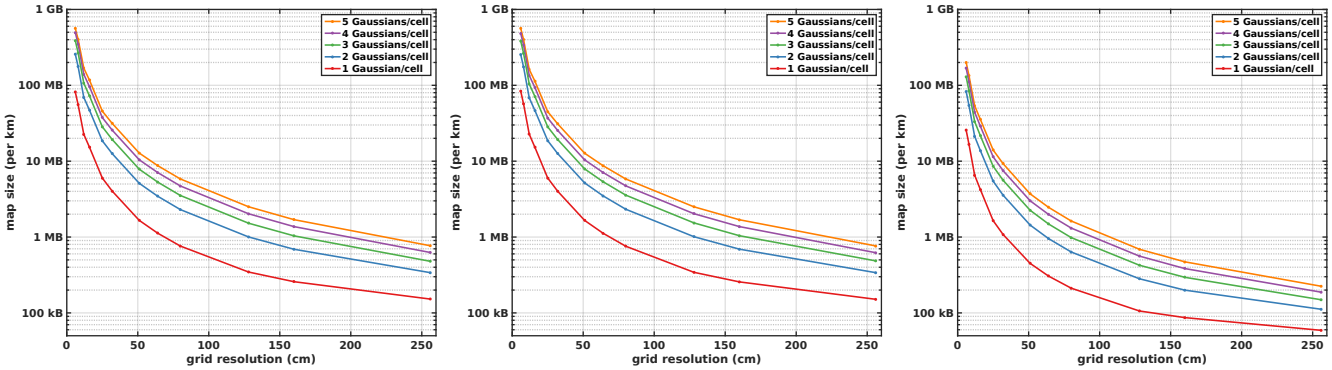
We more thoroughly looked at z-height registrations alone in the snow-filled *Downtown Dataset*, D-3, by randomly sampling within $10\ \mathrm{m}$ of the ground-truth pose. We present these results in two ways. First, we compiled the results into a histogram, as shown in the top row of Fig. 18(b). Here we see that our proposed solution is able to return to within $25\ \mathrm{cm}$ of the ground-truth with minimal outliers. Additionally, we see that because our method exploits the 3D structure, it is not impacted by harsh weather and significant amounts of falling snow, as shown in Fig. 18(a).

Second, we display this same registration error as a function of initial offset input to the scan matcher, as displayed in the bottom row of Fig. 18(b). We show that our registration success is not dictated by distance from the optimum, as long as our search space is able to enclose the true transformation.

7.3.2 Construction Zone Registrations

We further look at registration errors through one of the three construction zones that was repaved during our dataset collection, see Fig. 19. In this figure, we display our single map containing $\mathcal{G}_z$ and $\mathcal{G}_{r,\mathtt{grd}}$ built using P-M1 and P-M2 data in Fig. 19(a). Figured (b)-(d) shows registration results evaluated against $\mathcal{G}_z$, $\mathcal{G}_{r,\mathtt{grd}}$, and the joint measurement over both. Further, each of these figures are drawn over a reference reflectivity map that was built using data on each day to demonstrate the changes over time—note that the map in (a) was still used for all experiments.

Even before construction began in Fig. 19(b), reflectivity poorly constrains longitudinally due to limited features—the method relies on cutouts into driveways



(a) $\mathcal{G}_z$ (b) $\mathcal{G}_r$ (c) $\mathcal{G}_{r,\mathtt{grd}}$

Fig. 16: Map size for Gaussian mixture maps over z-height ($\mathcal{G}_z$), reflectivity ($\mathcal{G}_r$), and ground-plane reflectivity ($\mathcal{G}_{r,\mathtt{grd}}$). Sizes are listed as a function of Gaussian mixture map grid resolution and number of Gaussians per cell. All sizes are per kilometer of road travel.

	Longitudinal		Lateral	
Map	Median	$1^{\text{st}}/3^{\text{rd}}$ Qtr.	Median	$1^{\text{st}}/3^{\text{rd}}$ Qtr.
$\mathcal{G}_z$	8.5 cm	(3.8 cm, 16.7 cm)	8.0 cm	(3.4 cm, 15.9 cm)
$\mathcal{G}_{r,\mathtt{grd}}$	10.9 cm	(3.8 cm, 35.0 cm)	4.6 cm	(2.0 cm, 8.6 cm)
$\mathcal{G}_z$,$\mathcal{G}_{r,\mathtt{grd}}$	7.7 cm	(3.4 cm, 15.2 cm)	5.3 cm	(2.4 cm, 9.8 cm)

Table 1: Comparison of errors between Gaussian mixture map types, showing the median, first quartile, and third quartiles of absolute deviation (longitudinally and laterally).

and roads for success. Over time, the ability to localize using reflectivity alone becomes impossible because the appearance is fundamentally different; however, localization using $\mathcal{G}_z$ remains effective and the joint measurement is not distracted by erroneous reflectivity measurements.

7.4 Filtered Experiments

We integrated our registration algorithm into the EKF localization framework described in Section 6. The only measurements used were those from a GPS unit for initialization, our IMU for vehicle odometry, and our multiresolution scan matches considering structure and appearance initialized around our 4σ posterior belief. Standard deviation for these scan registrations was set relative to our median absolute deviation derived in the previous section, $1.4826 \cdot \text{MAD}$; this scaling is so that MAD can be viewed as a consistent estimator of normally distributed variance (Rousseeuw and Croux, 1993). Further, we evaluate the measurement NIS and only include measurements that are 99% likely to be consistent with our filter—this allows our filtering to be robust to outliers.

Results are tabulated in Table 2, where we present longitudinal, lateral, and heading errors relative to ground-truth. Errors are shown in terms of RMS errors as well as percentages of filtered poses that fall within 5 cm, 25 cm, and 1 m (longitudinally and laterally). Overall, we see that our measurements result in better constraints laterally than longitudinal; lateral RMS errors are typically less than 10 cm and longitudinal RMS errors are within the range of 10 cm-13 cm.

Further, we demonstrate each log graphically over satellite imagery in Fig. 20, Fig. 21, and Fig. 22, where each log's trajectory is colored by L_2 error. We see that errors are frequently along highway and rural roads where longitudinal constraints become dependent on visible 3D structure (unconstrained via reflectivity as showin in Fig. 19). A common problem occurs when passing large semi-trailer trucks that fully occlude field of view of informative 3D structure beside the road, often leading to noisy measurements.

Additionally, we see our method is robust to radical appearance changes. This can be seen in Fig. 20(g) and Fig. 21(g) where our method can remain localized through repavings that completely altered the appearance of 0.5-1.0 km stretches of road. There are occasional spikes in accuracy through these regions, though we still maintain localization through these periods of drastic appearance changes. Moreover, we demonstrate in Fig. 22(e) that we are able to remain localized through heavy snowfall that was present during the D-3 log.

Session	RMS Error Long.	RMS Error Lat.	RMS Error Hdg.	p(err<5 cm) Long.	p(err<5 cm) Lat.	p (err<25 cm) Long.	p (err<25 cm) Lat.	p(err<1 m) Long.	p(err<1 m) Lat.
P-M1	10.1 cm	6.5 cm	0.09 °	60.9 %	58.3 %	96.8 %	99.4 %	100.0 %	100.0 %
P-M2	6.6 cm	5.4 cm	0.09 °	68.6 %	69.7 %	99.1 %	99.8 %	100.0 %	100.0 %
P-1	13.3 cm	8.4 cm	0.13 °	33.2 %	56.9 %	94.3 %	98.4 %	100.0 %	100.0 %
P-2	10.3 cm	7.6 cm	0.11 °	40.3 %	56.1 %	98.0 %	99.3 %	100.0 %	100.0 %
P-3	10.7 cm	8.2 cm	0.13 °	41.0 %	50.4 %	97.7 %	98.9 %	100.0 %	100.0 %
P-4	10.8 cm	8.3 cm	0.16 °	38.5 %	46.0 %	98.5 %	99.4 %	100.0 %	100.0 %
P-5	11.9 cm	9.3 cm	0.20 °	36.7 %	44.4 %	96.2 %	98.4 %	100.0 %	100.0 %
P-6	8.4 cm	8.2 cm	0.15 °	46.6 %	45.3 %	99.1 %	99.4 %	100.0 %	100.0 %
P-7	12.7 cm	9.1 cm	0.12 °	34.3 %	41.5 %	94.4 %	99.0 %	100.0 %	100.0 %
P-8	12.2 cm	9.5 cm	0.12 °	45.1 %	40.6 %	94.4 %	98.5 %	100.0 %	100.0 %
P-9	16.2 cm	9.8 cm	0.13 °	34.8 %	47.3 %	91.1 %	98.1 %	100.0 %	100.0 %
P-10	12.2 cm	10.6 cm	0.12 °	41.5 %	43.0 %	95.7 %	96.6 %	99.9 %	100.0 %
P-11	15.0 cm	12.3 cm	0.15 °	30.3 %	41.4 %	92.2 %	96.8 %	99.9 %	99.9 %
P-12	11.0 cm	12.7 cm	0.12 °	40.3 %	34.3 %	97.4 %	95.6 %	100.0 %	100.0 %
D-M1	10.7 cm	8.9 cm	0.17 °	30.0 %	43.1 %	98.3 %	99.3 %	100.0 %	100.0 %
D-M2	12.5 cm	10.3 cm	0.17 °	30.4 %	50.9 %	96.4 %	97.1 %	100.0 %	100.0 %
D-1	10.8 cm	9.8 cm	0.18 °	39.7 %	39.4 %	98.7 %	97.3 %	100.0 %	100.0 %
D-2	11.6 cm	9.1 cm	0.16 °	26.8 %	45.6 %	95.4 %	98.2 %	100.0 %	100.0 %
D-3	11.8 cm	10.9 cm	0.15 °	42.5 %	36.4 %	96.8 %	97.7 %	100.0 %	100.0 %

Table 2: Filtered results using joint measurements over structure and appearance for all datasets. We show longitudinal, lateral and heading RMS errors in addition to percentage of filtered poses that are within 5 cm, 25 cm, and 1 m of ground-truth, longitudinally and laterally. Top half of table tabulates results over the *PG14 Dataset*, and the bottom half covers the *Downtown Dataset*.

7.5 Run-time Analysis

Given that our registration approach is a function of desired search space, we analyze the run-time performance at several search windows of: $1\text{ m} \times 1\text{ m}$, $2\text{ m} \times 2\text{ m}$, $4\text{ m}\times 4\text{ m}$, $8\text{ m}\times 8\text{ m}$, and $16\text{ m}\times 16\text{ m}$. Fig. 23 shows framerate for each of these search spaces at a *single* rotational search, in which each registration likelihood is evaluated using approximately $300,000$ points. Results are presented for the CPU implementation and we show a $30--40\times$ speedup for both the exhaustive and multiresolution branch-and bound search when implemented on a GPU. Relative to our work presented in (Wolcott and Eustice, 2015), we no longer have to significantly downsample our point cloud to achieve real-time localization. Note the time speedup is more pronounced when more than one rotational offset is considered because the multiresolution search can short-circuit quicker.

During online performance, we initially need to perform a dense search over a window of $\sim 10\text{ m} \times 10\text{ m}$, which can be achieved in a little over a second. Over time, these search windows gradually shrink according to our posterior pose belief such that we can perform online localization using *all* measured 3D points at roughly $5-10$ Hz

Furthermore, rasterizing the Gaussian mixture maps into multiresolution lookup tables must be carefully managed when implemented on the CPU as these take 4562 ms to construct. However, this is dramatically improved to 114 ms implemented on a GPU ($40\times$ speedup).

7.6 Alternative Uses of GMM

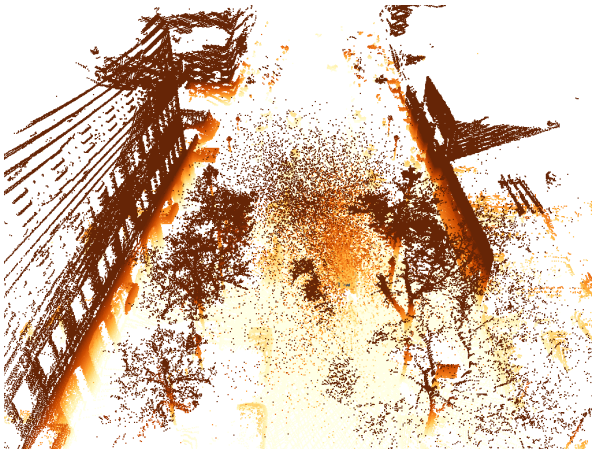
In addition to localization, Gaussian mixture maps over z-height can be used for other purposes. In this section, we briefly present two possible use cases: point cloud compression and obstacle background subtraction.

7.6.1 Point Cloud Compression

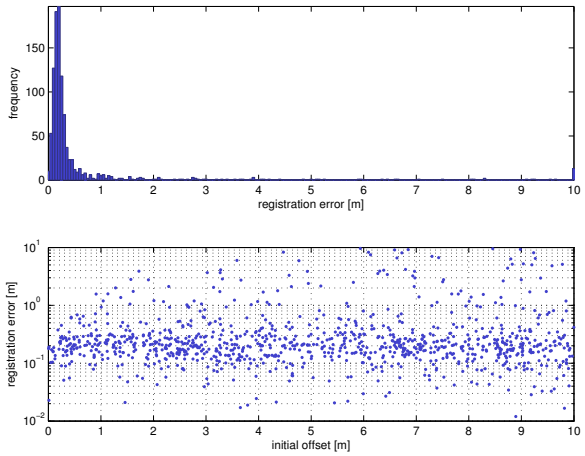
Storing raw point clouds can require more than 500 MB per km of road. As an alternative, Gaussian mixture maps over z-height can be a parametric method for compactly storing terrestrial maps. In Fig. 24, we show the efficacy of our method for retaining the true point cloud distribution using 1, 2, 3, 4, and 10 Gaussians per grid cell—this figure shows points that are within 2.5 standard deviations of each mixture component. It is clear that a single Gaussian per cell would be insufficient, though as few as 2 appears to well capture the building facade and foliage. These maps require roughly 10 MB, 20 MB, 30 MB, 40 MB, and 100 MB, respectively, per km of road.

7.6.2 Obstacle Background Subtraction

Another benefit of using structure in our automated vehicle's localization pipeline is that it provides a probabilistic method to classify 3D points as dynamic obstacles or belonging to the background environment. In generating the likelihood for a registration, we evaluate the likeli-



(a) Heavy Snowfall Point Cloud



(b) *Downtown Dataset*—D-3

Fig. 18: In (a), we show a point cloud rendering of typical snowfall (with ground-plane removed) during the D-3 dataset. Orange and brown points located at the center of the figure shows the dense snow returns. In (b), we demonstrate registration error using z-height alone on the snow-filled dataset, D-3. The top row shows a histogram of our L^2 error, demonstrating good registration performance. The bottom row shows a plot of initial offset versus registration error, where we show that our scan matching errors are independent of initial guess.

hood of each scan point against the prior map, which tells us how likely each scan point is to be part of the map. By looking at points that *poorly* align to the prior map (i.e., those with low likelihoods), we can perform a classification. We do this by setting a Mahalanobis distance threshold and labeling points that exceed this threshold as obstacles—this selection is precisely the outlier thresholds set to minimize effect of outliers in our robust cost formulation of (13). Our formulation allows us to do this classification on a frame-by-frame basis and extend our sensing range of obstacles. Visualization of point cloud classification can be seen in Fig. 25.

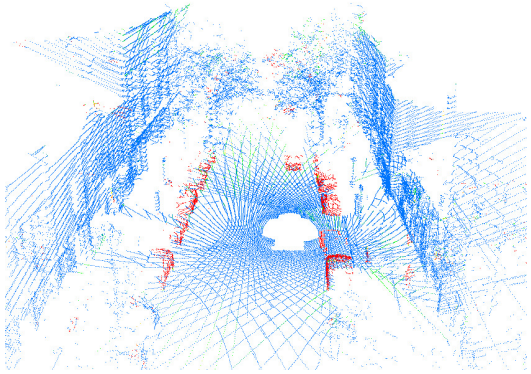


Fig. 25: Sample point cloud colored by Mahalanobis distance from the underlying map's Gaussian mixture. Note the parked cars in *red* and agreeing prior map in *blue* (including ground-plane, building facades, trees, and lightposts). Our method allows us to expand our obstacle sensing horizon, as we can not sense the ground-plane beyond 40 m.

8 Conclusion

In this paper, we demonstrated Gaussian mixture maps that reduce large point clouds into a compact, parametric representation that maintains expressibility over structure and appearance. Through the use of multiresolution rasterized maps that can be computed online, we can efficiently traverse these maps to find the guaranteed optimal registration using branch-and-bound search, rather than finding local optima as with modern scan matchers. Finally, we integrated this into an EKF to demonstrate that our autonomous platform can remain well localized in a prior map over more than 500 km of road data. Our proposed system is able to handle harsh weather and poorly textured roadways, which is a significant advantage over the current state-of-the-art methodologies for automated vehicle localization. We further demonstrated that localization can be done through construction zones undergoing drastic appearance changes, allowing us in future work to consider maintaining and updating these maps as they change. In addition, we expect to fully explore the impact of extreme road grades that may complicate our 2D registration and require more explicit search over z.

Acknowledgments

This work was supported in part by Ford Motor Company under the Ford-UM Alliance, and in part by The SMART Scholarship for Service Program by the Department of Defense under which R. Wolcott was supported for some of this work.

References

Aghamohammadi, A. A., Taghirad, H. D., Tamjidi, A. H., and Mihankhah, E. (2007). Feature-based laser scan matching for

accurate and high speed mobile robot localization. In *Proc. European Conf. on Mobile Robots*, Freiburg, Germany.

Aiger, D., Mitra, N. J., and Cohen-Or, D. (2008). 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics (TOG)*, 27(3):85.

Baldwin, I. and Newman, P. (2012). Road vehicle localization with 2d push-broom lidar and 3d priors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2611–2617, Saint Paul, MN, USA.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Biber, P. (2003). The normal distribution transform: A new approach to laser scan matching.

Chong, Z. J., Qin, B., Bandyopadhyay, T., Ang Jr., M. H., Frazzoli, E., and Rus, D. (2013). Synthetic 2d lidar for precise vehicle localization in 3d urban environment. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1554–1559, Karlsruhe, Germany.

Granger, S. and Pennec, X. (2002). Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Proceedings of the European Conference on Computer Vision*, pages 418–432, Copenhagen, Denmark.

Jian, B. and Vemuri, B. C. (2011). Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645.

Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Trans. on Robotics (TRO)*, 24(6):1365–1378.

Krishnan, A. K. and Saripalli, S. (2014). Point cloud registration using congruent pyramids. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1812–1817, Chicago, IL, USA.

Kümmerle, R., Triebel, R., Pfaff, P., and Burgard, W. (2007). Monte carlo localization in outdoor terrains using multi-level surface maps. In *Proceedings of the International Conference on Field and Service Robotics*, Chamonix, France.

Leonard, J. J. and Rikoski, R. J. (2000). Incorporation of delayed decision making into stochastic mapping. In *Experimental Robotics VII, D. Rus and S. Singh, Eds., Lecture Notes in Control and Information Sciences*, pages 533–542. Springer-Verlag.

Levinson, J., Montemerlo, M., and Thrun, S. (2007). Map-based precision vehicle localization in urban environments. In *Proceedings of the Robotics: Science & Systems Conference*, Atlanta, GA.

Levinson, J. and Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4372–4378, Anchorage, AK.

Levinson, J. and Thrun, S. (2014). Unsupervised calibration for multi-beam lasers. In *Experimental Robotics*, pages 179–193. Springer.

Maddern, W., Pascoe, G., and Newman, P. (2015). Leveraging Experience for Large-Scale LIDAR Localisation in Changing Cities. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA.

Magnusson, M. (2009). *The Three-Dimensional Normal-Distributions Transform—An Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, Örebro University.

Maier, D., Hornung, A., and Bennewitz, M. (2012). Real-time navigation in 3D environments based on depth camera data. In *Proc. of the IEEE/RAS Int. Conf. Humanoid Robots*, pages 692–697, Osaka, Japan.

Mellado, N., Aiger, D., and Mitra, N. J. (2014). Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33(5):205–215.

Olson, E. (2009). Real-time correlative scan matching. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4387–4393, Kobe, Japan.

Olson, E. (2015). M3rsm: Many-to-many multi-resolution scan matching. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA.

Pandey, G., McBride, J. R., Savarese, S., and Eustice, R. M. (2011). Visually bootstrapped generalized ICP. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2660–2667, Shanghai, China.

Ripperda, N. and Brenner, C. (2005). Marker-free registration of terrestrial laser scans using the normal distribution transform. volume 4, Mestre-Venice, Italy.

Rousseeuw, P. J. and Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283.

Rusu, R. B. (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany.

Ryde, J. and Hu, H. (2010). 3d mapping with multi-resolution occupied voxel lists. *Auton. Robots*, 28(2):169–185.

Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Proceedings of the Robotics: Science & Systems Conference*, Seattle, WA.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 1:167–193.

Triebel, R., Pfaff, P., and Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2276–2282, Beijing, China.

Ulaş, C. and Temelta, H. (2013). 3d multi-layered normal distribution transform for fast and long range scan matching. *J. Intell. and Robotic Syst.*, 71(1):85–108.

Wolcott, R. W. and Eustice, R. M. (2014). Visual localization within LIDAR maps for automated urban driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, Chicago, IL, USA.

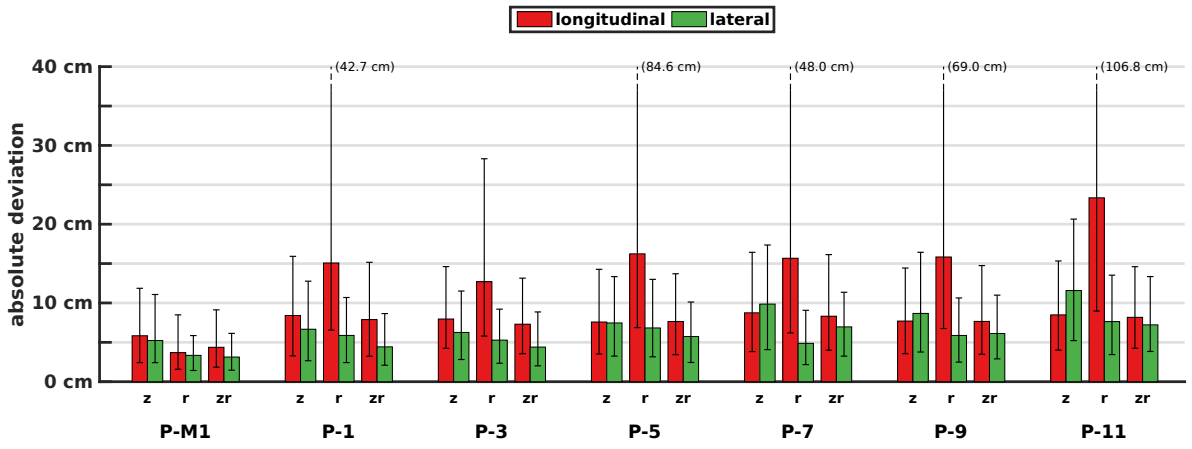
Wolcott, R. W. and Eustice, R. M. (2015). Fast LIDAR localization using multiresolution Gaussian mixture maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2814–2821, Seattle, WA, USA.

A Index to Multimedia Extensions

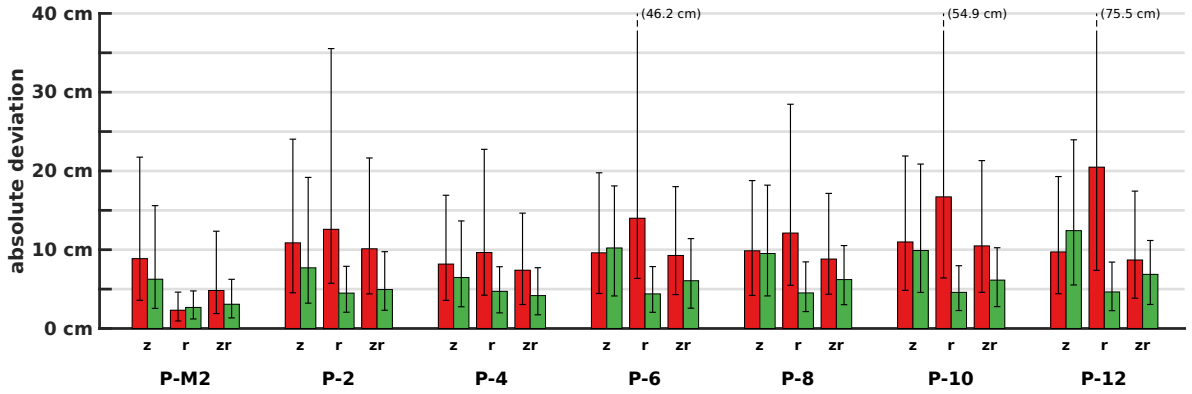
The multimedia extensions to this article are at: http://www.ijrr.org

Table of Multimedia Extensions

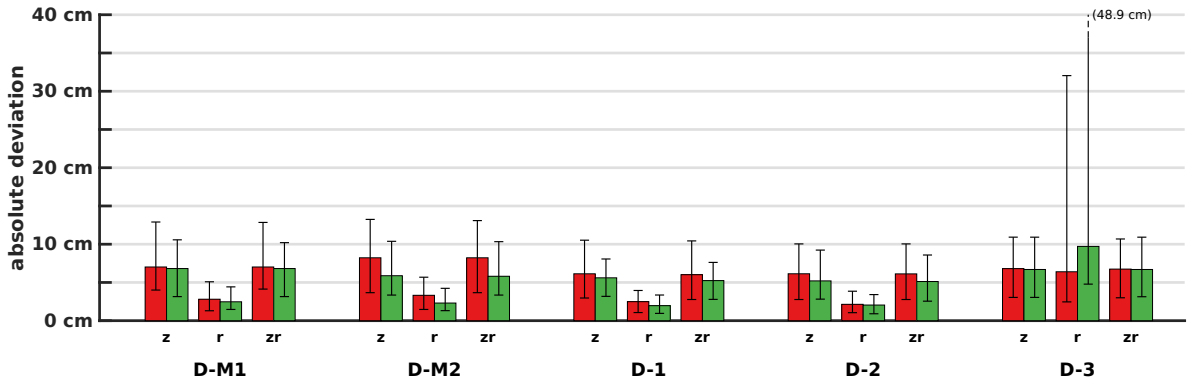
Extension	Type	Description
1	Video	General overview of Gaussian mixture map formulation



(a) *PG14 Dataset*: TORC Logs

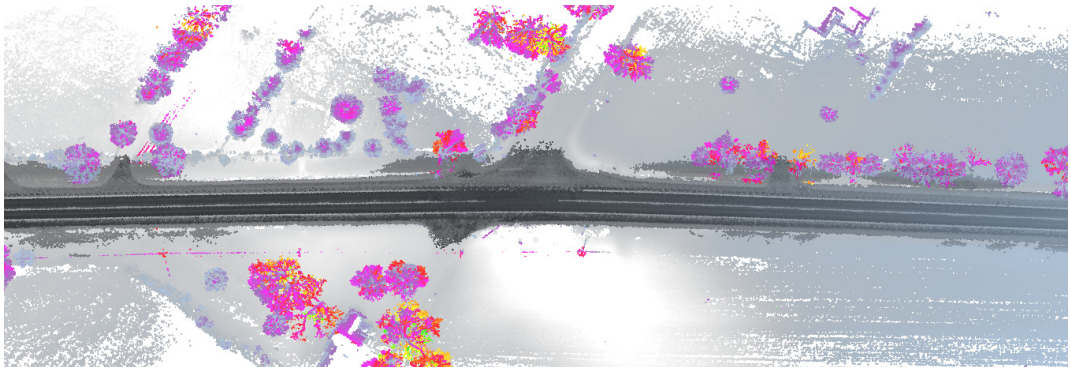


(b) *PG14 Dataset*: Fusion Logs

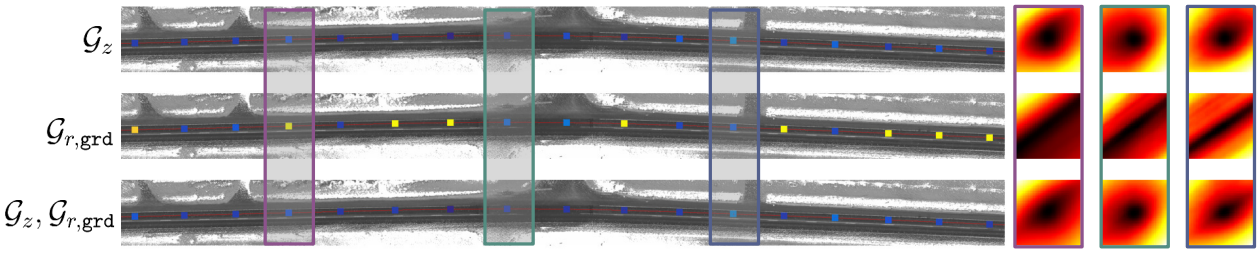


(c) *Downtown Dataset*

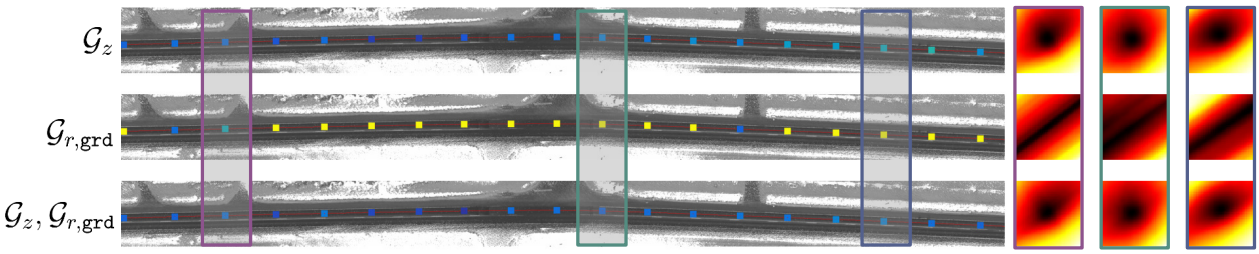
Fig. 17: This figure shows the registration errors from randomly sampled points along each log using $\mathcal{G}_z$, $\mathcal{G}_{r,\mathtt{grd}}$, and $\{\mathcal{G}_z, \mathcal{G}_{r,\mathtt{grd}}\}$, marked by **z**, **r**, and **zr**, respectively. Bars indicate the longitudinal (*red*) and lateral (*green*) median absolute deviation (MAD) and the error whiskers mark the first and third quartiles of absolute deviation. In most logs, the joint likelihood measure over structure and appearance yields improved performance relative to the likelihood measure over structure or appearance alone. Moreover, the use of structure prevents large longitudinal errors during the *PG14 Dataset* and allows for consistent localization during the snow dataset, D-3. The reflectivity alone does quite well in some circumstances, such as the *Downtown Dataset* where road paint is well maintained, though the joint measure still results in errors less than 10 cm.



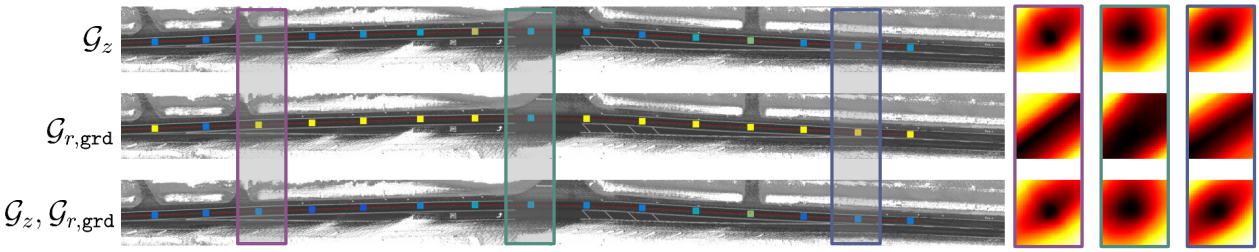
(a) $\mathcal{G}_z$ and $\mathcal{G}_{r,\texttt{grd}}$ Maps



(b) Fusion, May 8

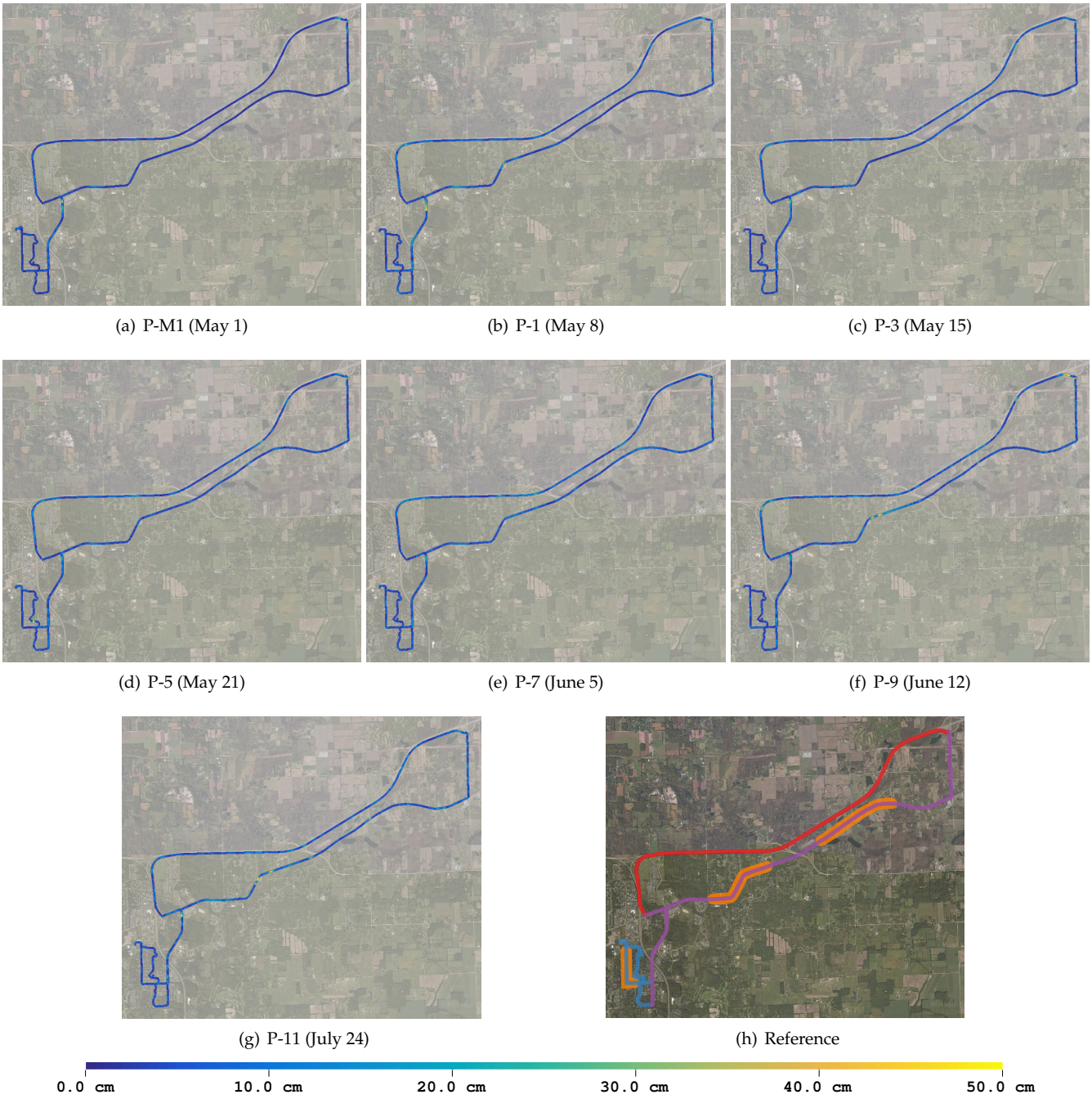


(c) Fusion, June 5



(d) Fusion, July 24

Fig. 19: This figure demonstrates the registration quality of our joint measurement function over a segment of road that is poorly constrained by appearance *and* undergoes significant construction. In (a), we show the map constructed for our experiments (showing the maximum mean component), visualizing the structure (*purple-green*) and appearance (*black/white*) together. In (b)-(d), we show registrations performed over each map type spanning different logs, rendering: a map that reflects the appearance *on that day*, registration L_2 error (colored dots), and sample cost function sweeps (right). Despite appearance cost functions that poorly constrain our pose and radically change, our z-height measurements and resulting joint measurements remain well constrained.

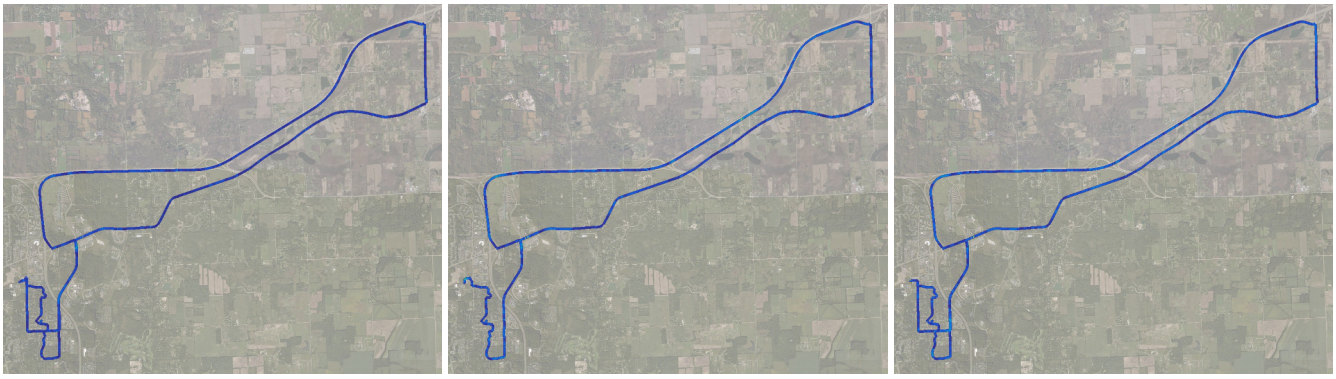


(a) P-M1 (May 1) (b) P-1 (May 8) (c) P-3 (May 15)

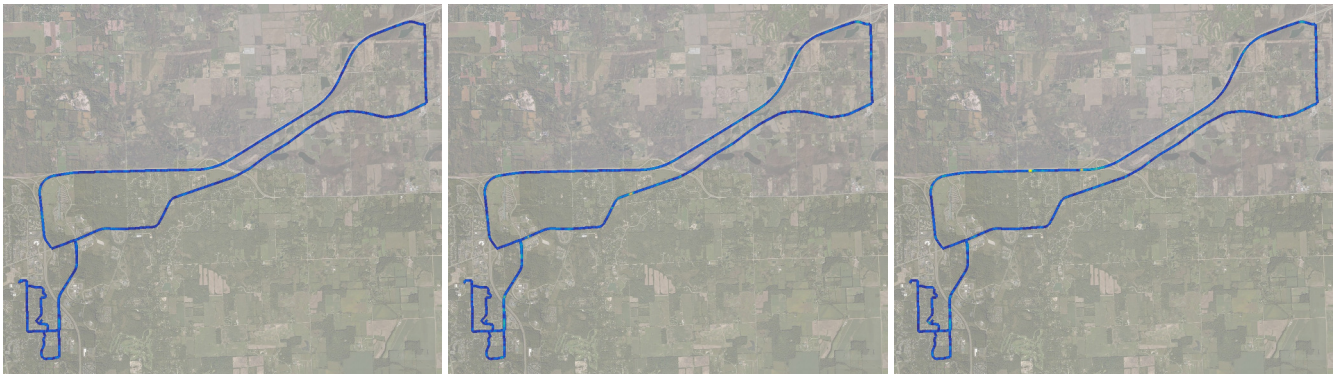
(d) P-5 (May 21) (e) P-7 (June 5) (f) P-9 (June 12)

(g) P-11 (July 24) (h) Reference

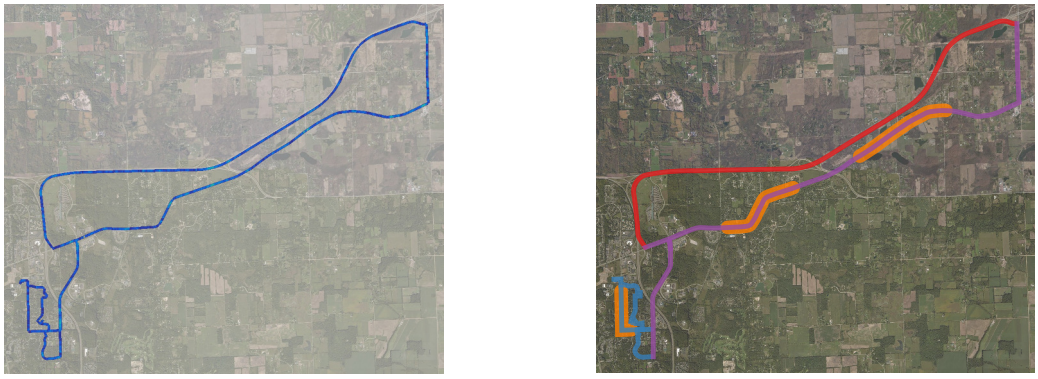
Fig. 20: Filtered results using joint measurements over structure and appearance for the TORC logs of the *PG14 Dataset*, where the trajectory is colored by L_2 error. Most noticeable errors are longitudinal—primarily on the highway or other long, straight stretches with little variation in that dimension. Further, note the slight increase in errors through construction zones. Despite these increases, our filter does not diverage and remains localized within acceptable tolerances. A route reference is provided in (h) highlighting residential roads (*blue*), rural roads (*purple*), highways (*red*), and construction zones (*orange*).



(a) P-M2 (May 1) (b) P-2 (May 8) (c) P-4 (May 15)

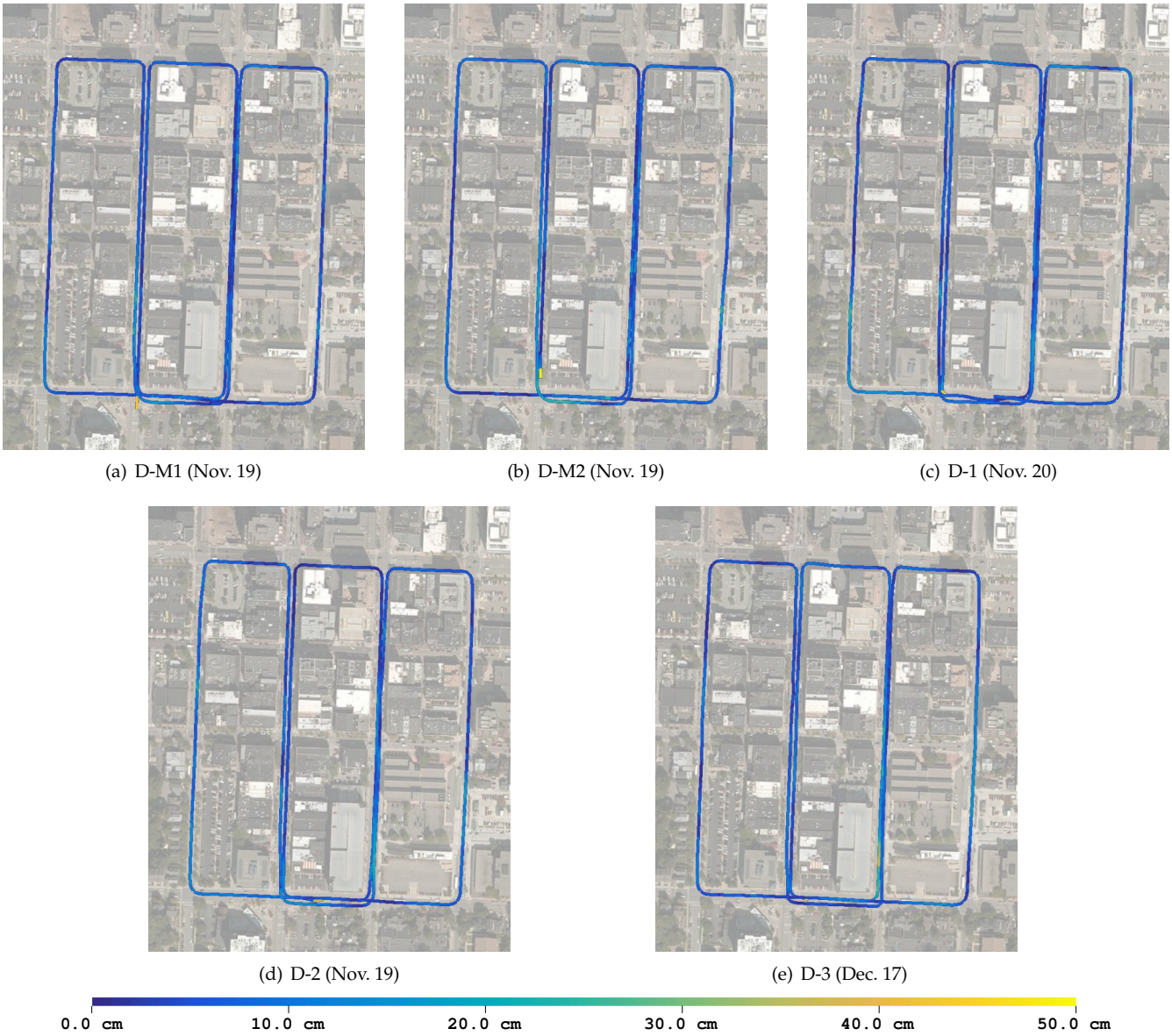


(d) P-6 (May 21) (e) P-8 (June 5) (f) P-10 (June 12)



(g) P-12 (July 24) (h) Reference

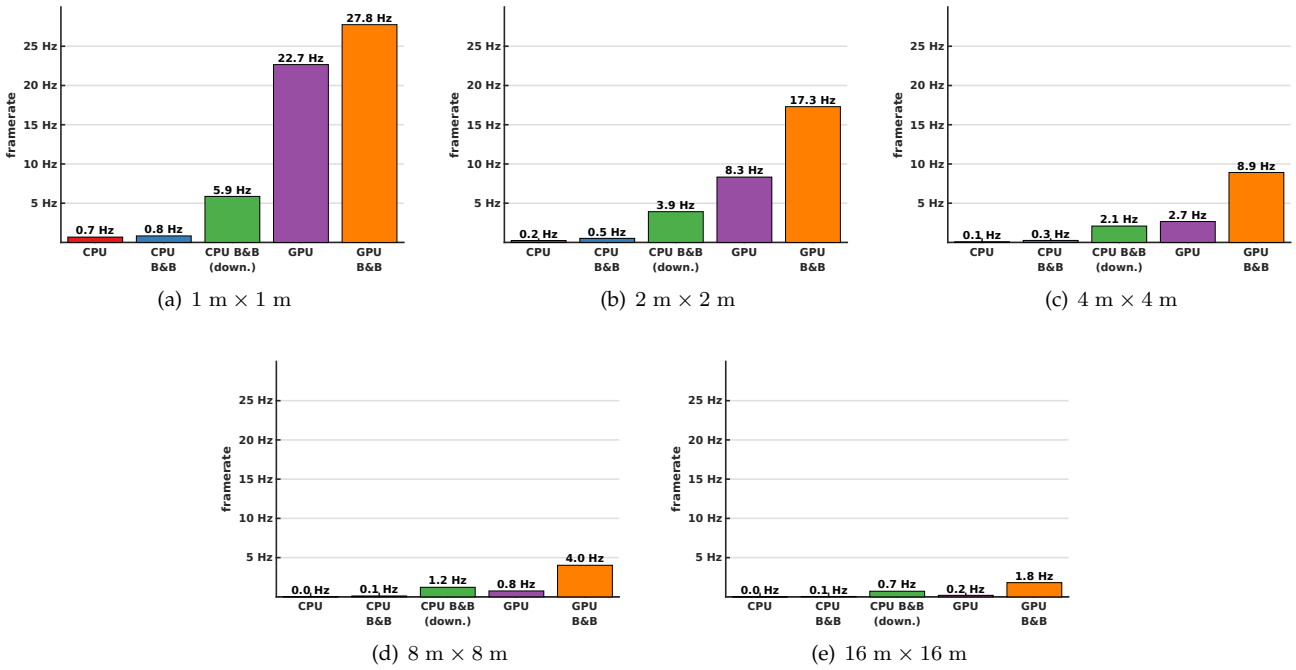
Fig. 21: Filtered results using joint measurements over structure and appearance for the Fusion logs of the *PG14 Dataset*, where the trajectory is colored by L_2 error. Most noticeable errors are longitudinal—primarily on the highway or other long, straight stretches with little variation in that dimension. Further, note the slight increase in errors through construction zones. Despite these increases, our filter does not diverge and remains localized within acceptable tolerances. A route reference is provided in (h) highlighting residential roads (*blue*), rural roads (*purple*), highways (*red*), and construction zones (*orange*).



(a) D-M1 (Nov. 19) (b) D-M2 (Nov. 19) (c) D-1 (Nov. 20)

(d) D-2 (Nov. 19) (e) D-3 (Dec. 17)

Fig. 22: Filtered results using joint measurements over structure and appearance for the *Downtown Dataset*, where the trajectory is colored by L_2 error. Aside from initial convergence time for the filter (the bright *yellow* segment on the bottom of each figure), our method does quite well in the urban environment. This includes staying well localized through heavy snowfall during D-3 log.



(a) $1\text{ m} \times 1\text{ m}$ (b) $2\text{ m} \times 2\text{ m}$ (c) $4\text{ m} \times 4\text{ m}$

(d) $8\text{ m} \times 8\text{ m}$ (e) $16\text{ m} \times 16\text{ m}$

Fig. 23: This figure shows registration framerates of our proposed localization method using Gaussian mixture maps over various search spaces. Left-to-right we show framerates using the CPU for exhaustive search (*red*), multiresolution branch-and-bound (*blue*), and again using a downsampled point cloud that was necessary in our previous work to meet acceptable localization framerates (Wolcott and Eustice, 2015) (*green*). We further show the exhaustive search (*purple*) and the branch-and-bound search (*orange*) when implemented on a GPU. Note, these figures were generated only searching over a *single* rotational offset.

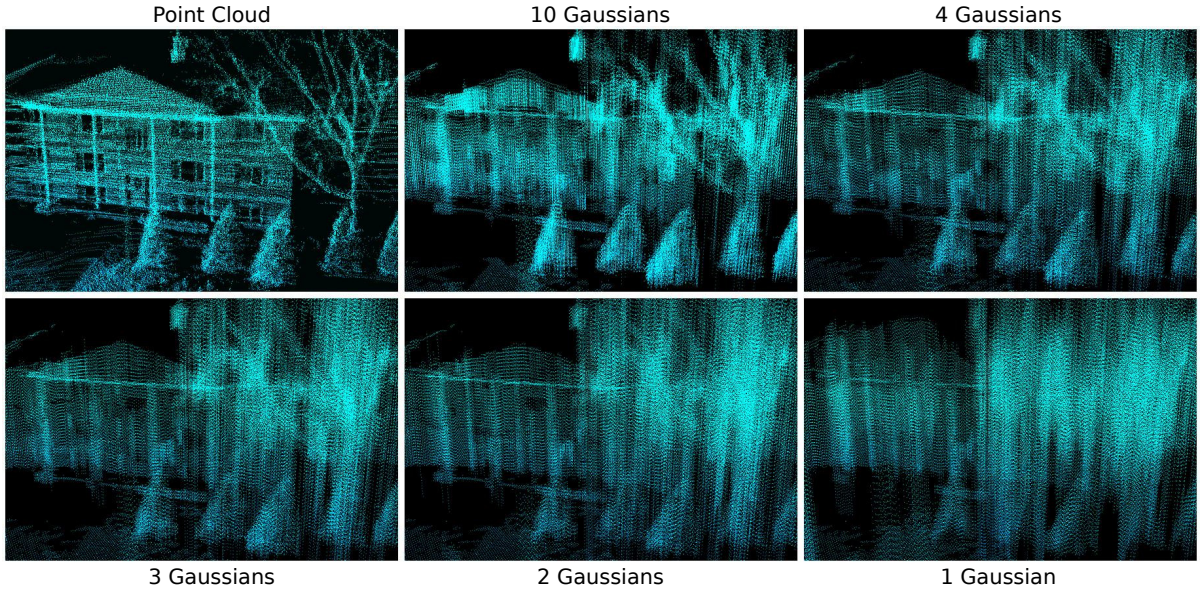


Fig. 24: Gaussian mixture maps over z-height can be used for point cloud compression. In this figure, we demonstrate the initial point cloud, seen in the top-left and corresponding reconstructions using various $\mathcal{G}_z$ maps.