2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Asynchronous Multi-Sensor Fusion for 3D Mapping and Localization

Patrick Geneva*, Kevin Eckenhoff†, and Guoquan Huang†

Abstract— **In this paper, we address the problem of optimally fusing multiple heterogeneous and asynchronous sensors for use in 3D mapping and localization of autonomous vehicles. To this end, based on the factor graph-based optimization framework, we design a modular sensor-fusion system that allows for efficient and accurate incorporation of multiple navigation sensors operating at different sampling rates. In particular, we develop a general method of out-of-sequence (asynchronous) measurement alignment to incorporate heterogeneous sensors into a factor graph for mapping and localization in 3D, without requiring the addition of new graph nodes, thus allowing the graph to have an overall reduced complexity. The proposed sensor-fusion system is validated on a real-world experimental dataset, in which the asynchronous-measurement alignment is shown to have an improved performance when compared to a naive approach without alignment.**

I. INTRODUCTION

Autonomous driving is an emerging technology that enables the reduction of traffic accidents and allows for those who are unable to drive for various medical conditions to regain their independence, by performing intelligent perception and planning based on multimodal sensors such as LIDARs, cameras, IMUs and GPS. It is critical for an autonomous vehicle to perform precise, robust localization for decision making as it is the sub-system that almost all online autonomous operations depend upon. There has been a large amount of research efforts focused on multi-sensor fusion for localization [1], which has reached a certain level of maturity, yielding a bounded problem given the well structured environment a vehicle operates in. In particular, graph-based optimization has recently prevailed for robot mapping and localization [2]. Due to the different sampling rates of the heterogeneous sensors, measurements arrive at different times. Accurate alignment of such out-of-sequence (i.e., asynchronous) measurements before optimally fusing them through graph optimization, while essential, has not been sufficiently investigated in the literature.[1]

Factor graph-based formulation [4] is desirable due to its ability to allow for delayed incorporation of asynchronous measurements. Indelman et al. [5] addressed the problem of the inclusion of asynchronous measurements by taking advantage of IMU preintegrated terms. This allowed them to incorporate any set of asynchronous sensors whose rates are slower than that of the IMU. Sünderhauf et al. [6] looked to address the incorporation of measurements with unknown time delays. Using high frequency odometry measurements, they created states for *each* incoming odometry measurement so that delayed factors could be directly connected to the closest high frequency state. While both of these can be used to address arbitrary amounts of delay between sensors, they add large amounts of additional factors and edges to the graph. In contrast, the proposed approach incorporates measurements of different frequencies *without* increasing of the overall graph complexity. It should be noted that while the proposed method does reduce the computational cost of optimization, graph size reduction also positively impacts memory constrained devices.

From the theoretical perspective, as the main contribution of this paper, we accurately align both asynchronous unary and binary graph factors to existing states based on our *analytically* derived linear 3D pose interpolation. This interpolation allows for the direct addition of asynchronous measurements into the graph, without the need for extra nodes to be added or for the naive ignoring of the measurement delay. Patron-Perez et al. [7] first proposed a spline-based trajectory method that allows for the fusion of delayed measurements with the consequence of an increase of overall system complexity and deviation from a pure pose graph. Outside of graph-based optimization, interpolation has been used to correct time offsets of continuous measurements such as rolling shutter camera features and LIDAR point clouds. In particular, Guo et al. [8] introduced the idea of linear interpolation between past camera poses, which allowed for the use of extracted features from rolling shutter cameras. Ceriani et al. [9] used a linear interpolation between two poses in $SE(3)$ to unwarp LIDAR point measurements. In this work, however, we focus on the *analytical* derivation of the $SO(3) \times \mathbb{R}^3$ interpolation and its application inside of a graph-based optimization framework to allow for the efficient alignment of asynchronous measurements.

From the system perspective, we design and implement a modular framework for fusing a variety of sensors, where we decouple the sensor fusion and pose estimation to allow for any sensor to be incorporated. This decoupled framework allows for an arbitrary sensor odometry (ego-motion) module, that emits 3D poses, to be included and leveraged to improve the state estimate. Special care is also taken to

This work was partially supported by the University of Delaware College of Engineering, UD Cybersecurity Initiative, the NSF (IIS-1566129), and the DTRA (HDTRA1-16-1-0039).

*Author is with the Department of Computer & Information Sciences, University of Delaware, Newark, DE 19716, USA. Email: pgeneva@udel.edu

†Authors are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: {keck,ghuang}@udel.edu

[1]It should be noted that the asynchronous measurement alignment under consideration is different from the time synchronization (or temporal sensor calibration) [3]; that is, even if sensors are well synchronized, their observations are still collected asynchronously.

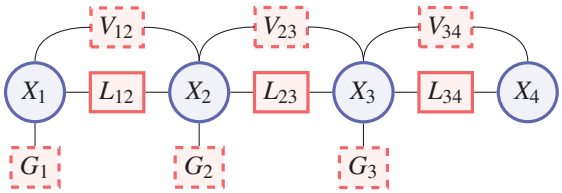

Fig. 1: Example of a factor graph that our system created. States that will be estimated are denoted in circles and measurements are denoted in squares. Note that we differentiate factors with interpolated measurements with dashed outlines.

ensure that odometry modules emitting 3D poses in unknown frame of references are converted into local robot-centric measurements, allowing for their direct incorporation into the global estimation of the robot's pose.

A. Notations

In this work, we parameterize the pose of each timestep as $\{{}^i_G\mathbf{R}, {}^G\mathbf{p}_i\} \in SO(3) \times \mathbb{R}^3$, which describes the rotation from the global frame $\{G\}$ to the local frame $\{i\}$ and the position of the frame $\{i\}$ seen from the global frame $\{G\}$ of reference. We denote the vector form of the matrix exponential as $\text{Exp}(\cdot)$ which maps $\mathbb{R}^3 \rightarrow SO(3)$, (e.g., $\text{Exp}({}^G_I\boldsymbol{\theta}) = {}^G_I\boldsymbol{R}$). Similarly, we define the matrix logarithm $\text{Log}(\cdot)$ to map between $SO(3) \rightarrow \mathbb{R}^3$, (e.g., $\text{Log}({}^G_I\boldsymbol{R}) = {}^G_I\boldsymbol{\theta}$). Other notation and identities can be found in Appendix A.

II. GRAPH-BASED ESTIMATION

As the vehicle moves through the environment, a set of measurements, $\mathbf{z}$, is collected from its sensors, such as LIDAR scans, images, GPS, etc. These measurements relate to the underlying state to be estimated, $\mathbf{x}$. This process can be represented by a graph, where nodes correspond to parameters to be estimated (i.e., historical vehicle poses). Incoming measurements are represented as edges connecting their involved nodes (see Figure 1). Under the assumption of independent Gaussian noise corruption of our measurements, we formulate the Maximum Likelihood Estimation (MLE) problem as the following nonlinear least-squares problem [10, 11]:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \sum_i ||\mathbf{r}_i(\mathbf{x})||^2_{\mathbf{P}_i} \tag{1}$$

where $\mathbf{r}_i$ is the zero-mean residual associated with measurement $\mathbf{z}_i$, $\mathbf{P}_i$ is the measurement covariance, and $||\mathbf{v}||^2_{\mathbf{P}} = \mathbf{v}^\top \mathbf{P}^{-1}\mathbf{v}$ is the energy norm. This problem can be solved iteratively by linearizing about the current estimate, $\hat{\mathbf{x}}^-$, and defining a new optimization problem in terms of the *error state*, $\Delta\mathbf{x}$:

$$\Delta\mathbf{x}^- = \underset{\Delta\mathbf{x}}{\text{argmin}} \sum_i \left|\left|\mathbf{r}_i\left(\hat{\mathbf{x}}^-\right) + \mathbf{H}_i\Delta\mathbf{x}\right|\right|^2_{\mathbf{P}_i} \tag{2}$$

where $\mathbf{H}_i = \frac{\partial \mathbf{r}_i(\hat{\mathbf{x}}^- \boxplus \Delta\mathbf{x})}{\partial \Delta\mathbf{x}}$ is the Jacobian of i-th residual with respect to the error state. We define the generalized update operation, $\boxplus$, which maps the error state to the full state. Given the error state $\{{}^i_G\tilde{\boldsymbol{\theta}}, {}^G\tilde{\mathbf{p}}_i\}$, this update operation can be written as $\{\text{Exp}\left(-{}^i_G\tilde{\boldsymbol{\theta}}\right){}^i_G\hat{\mathbf{R}}^-, {}^G\hat{\mathbf{p}}_i^- + {}^G\tilde{\mathbf{p}}_i\}$. After solving the linearized system, the current state estimate is updated as $\hat{\mathbf{x}}^+ = \hat{\mathbf{x}}^- \boxplus \Delta\mathbf{x}^-$. This linearization process is then repeated until convergence. While there are openly available solvers [10, 12, 13], the computational complexity of the graph-based optimization can reach $O(n^3)$ in the worst case, with n being the dimension of $\mathbf{x}$.

III. ASYNCHRONOUS MEASUREMENT ALIGNMENT

A. Unary Factors

It is clear that a reduction in the number of states being estimated can both help with the overall computational complexity and the physical size of a graph during long-term SLAM. Naively, if a new node is created for each incoming measurement, the overall graph optimization frequency can suffer. To prolong high frequency graph optimization, we present our novel method of measurement alignment which allows for the estimation of a *single set* of poses that all other incoming measurements are aligned to.

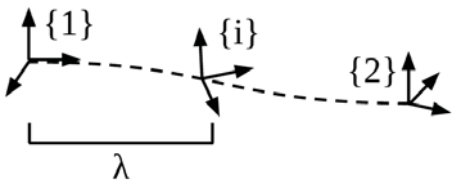

Fig. 2: Given two *measurements* in the global frame of reference $\{1\}$ and $\{2\}$, we interpolate to a new pose $\{i\}$. The above λ is the time-distance fraction that defines how much to interpolate the pose.

Unary factors can appear when sensors measure information with respect to a single time instance. For example, GPS can provide global position measurements indirectly through latitude, longitude, and altitude readings, while LIDAR scan-matching to known point cloud maps can provide a direct reading of the global pose. To limit the addition of new graph nodes when receiving asynchronous data, we interpolate between two sequential 3D pose *measurements* to a given state timestamp. This new interpolated measurement can then be directly added as a prior factor on the given state.

We define a time-distance fraction corresponding to the time-instance between two consecutive measurements as follows (see Figure 2):

$$\lambda = \frac{(t_i - t_1)}{(t_2 - t_1)} \tag{3}$$

where t_1 and t_2 are the timestamps of the bounding measurements, and t_i is the desired interpolation time (i.e., the timestamp of the given state). Under the assumption of a constant velocity (both translational and angular) motion model, we interpolate between the two 3D pose measurements:

$${}^i_G\mathbf{R} = \text{Exp}\left(\lambda\ \text{Log}({}^2_G\mathbf{R}{}^1_G\mathbf{R}^\top)\right){}^1_G\mathbf{R} \tag{4}$$

$${}^G\mathbf{p}_i = (1-\lambda){}^G\mathbf{p}_1 + \lambda\,{}^G\mathbf{p}_2 \tag{5}$$

where $\{{}^1_G\boldsymbol{R}, {}^G\boldsymbol{p}_1\}$ and $\{{}^2_G\boldsymbol{R}, {}^G\boldsymbol{p}_2\}$ are the bounding measurement poses and $\{{}^i_G\boldsymbol{R}, {}^G\boldsymbol{p}_i\}$ is the aligned pose measurement to the given state timestamp. The last step before directly adding this 3D pose as a prior measurement is to correctly compute the corresponding covariance needed in graph-based optimization. Hence, we perform the following covariance propagation:

$$\boldsymbol{P}_i = \boldsymbol{H}_u \boldsymbol{P}_{1,2} \boldsymbol{H}_u{}^T \tag{6}$$

$$\boldsymbol{H}_u = \begin{bmatrix} \frac{\partial^i_G \tilde{\boldsymbol{\theta}}}{\partial^1_G \tilde{\boldsymbol{\theta}}} & \mathbf{0}_{3\times3} & \frac{\partial^i_G \tilde{\boldsymbol{\theta}}}{\partial^2_G \tilde{\boldsymbol{\theta}}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \frac{\partial^G \tilde{\boldsymbol{p}}_i}{\partial^G \tilde{\boldsymbol{p}}_1} & \mathbf{0}_{3\times3} & \frac{\partial^G \tilde{\boldsymbol{p}}_i}{\partial^G \tilde{\boldsymbol{p}}_2} \end{bmatrix} \tag{7}$$

where $\boldsymbol{P}_{1,2}$ is the joint covariance matrix from the bounding measurements, and $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{p}}$ are the error states of each angle and position measurement, respectively. For detailed calculations of all Jacobians derived in this paper, we refer the reader to the companion technical report [14]. The resulting non-zero Jacobian matrix entries are analytically derived as:

$$\frac{\partial^i_G \tilde{\boldsymbol{\theta}}}{\partial^1_G \tilde{\boldsymbol{\theta}}} = -{}^i_1\boldsymbol{R} \left(J_r\Big(\lambda \ \mathrm{Log}({}^2_1\boldsymbol{R})\Big) \ \lambda \ J_r^{-1}\Big(\mathrm{Log}({}^2_1\boldsymbol{R})\Big) - I \right) \tag{8}$$

$$\frac{\partial^i_G \tilde{\boldsymbol{\theta}}}{\partial^2_G \tilde{\boldsymbol{\theta}}} = {}^i_1\boldsymbol{R} \ J_r\Big(-\lambda \ \mathrm{Log}({}^2_1\mathbf{R}^\top)\Big) \ \lambda \ J_r^{-1}\Big(\mathrm{Log}({}^2_1\mathbf{R}^\top)\Big) \tag{9}$$

$$\frac{\partial^G \tilde{\boldsymbol{p}}_i}{\partial^G \tilde{\boldsymbol{p}}_1} = (1-\lambda)\ \mathbf{I}\ , \quad \frac{\partial^G \tilde{\boldsymbol{p}}_i}{\partial^G \tilde{\boldsymbol{p}}_2} = \lambda\ \mathbf{I} \tag{10}$$

where the Right Jacobian of $SO(3)$ denoted as $J_r(\phi)$ and its inverse $J_r^{-1}(\phi)$ (see Appendix A).

B. Binary Factors

Designing multi-sensor systems for state estimation often requires fusing asynchronous motion information from different sensor modules (e.g., ORB-SLAM2 [15] or LOAM [16]). A difficulty that arises is the unknown transformation between the different global frame of references of each module. This unknown comes from both the rigid transformation between sensors (which can be found through extrinsic calibration) and the fact that each module *initializes* its *own* global frame of reference independently. Rather than directly modifying the codebase of each module or performing sophisticated initialization, we combine sequential 3D pose measurements into *relative* transforms; thereby, we remove the ambiguity of each module-to-module transformation.

In particular, given two pose measurements in the second sensor's global frame, $\{{}^1_o\boldsymbol{R}, {}^o\boldsymbol{p}_1\}$ and $\{{}^2_o\boldsymbol{R}, {}^o\boldsymbol{p}_2\}$ with the joint covariance $\boldsymbol{P}_{1,2}$, we calculate a new relative measurement as follows:

$${}^2_1\boldsymbol{R} = {}^2_o\boldsymbol{R}\,{}^1_o\boldsymbol{R}^\top \tag{11}$$

$${}^1\boldsymbol{p}_2 = {}^1_o\boldsymbol{R}\,({}^o\boldsymbol{p}_2 - {}^o\,\boldsymbol{p}_1) \tag{12}$$

where we define the unknown global frame of these 3D pose measurements as $\{o\}$ and their corresponding reference frames as $\{1\}$ and $\{2\}$. To find the covariance of this new relative measurement, we perform the following covariance propagation based on the above measurement transformation:

$$\boldsymbol{P}_{12} = \boldsymbol{H}_r \boldsymbol{P}_{1,2} \boldsymbol{H}_r{}^T \tag{13}$$

The resulting Jacobian matrix $\boldsymbol{H}_r$ is defined as the following:

$$\boldsymbol{H}_r = \begin{bmatrix} -{}^2_1\boldsymbol{R} & \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \lfloor {}^1_o\boldsymbol{R}({}^o\boldsymbol{p}_2 - {}^o\,\boldsymbol{p}_1)\times \rfloor & -{}^1_o\boldsymbol{R} & \mathbf{0}_{3\times3} & {}^1_o\boldsymbol{R} \end{bmatrix} \tag{14}$$

where $\lfloor \cdot \times \rfloor$ is the skew-symmetric matrix, see Appendix A.

We now have the $\{{}^2_1\boldsymbol{R}, {}^1\boldsymbol{p}_2\}$ relative measurement between two poses and its corresponding covariance $\boldsymbol{P}_{12}$. If this transformation is not in the same sensor frame of reference (e.g., relative transform is between camera to camera and the state is LIDAR to LIDAR), one can use the method described in Appendix B to convert the measurement into the same sensor frame of reference as the state.

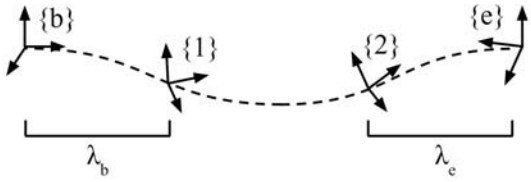

Fig. 3: Given a relative *measurement* between the $\{1\}$ and $\{2\}$ frame of reference, we extrapolate this relative transformation to the desired beginning $\{b\}$ and end $\{e\}$ timesteps. The above λs are the time-distance fractions that we use to extrapolate the relative measurement.

Due to the asynchronous nature of the measurements from two different sensors, the times corresponding to the start and end of incoming relative measurements will not align with existing state timestamps. Therefore, under the assumption of a constant velocity motion, we *extrapolate* the relative measurement to given beginning and end state timestamps. This intuitively corresponds to a "stretching" of the relative pose measurement in time. We define two time-distance fractions that determine how much the relative measurement needs to be extended (see Figure 3):

$$\lambda_b = \frac{t_1 - t_b}{t_2 - t_1} \qquad \lambda_e = \frac{t_e - t_2}{t_2 - t_1} \tag{15}$$

The λs describe the magnitude that the relative measurement is to be "stretched" in each direction, with the subscripts t_b and t_e denoting the beginning and end state timestamps. These time-distance fractions can also be negative, corresponding to the "shrinking" of the relative measurement. Formally, we define the following relative measurement extrapolation equations:

$${}^e_b\boldsymbol{R} = \mathrm{Exp}\left[(1+\lambda_b+\lambda_e)\mathrm{Log}\left({}^2_1\boldsymbol{R}\right)\right] \tag{16}$$

$${}^b\boldsymbol{p}_e = (1+\lambda_b+\lambda_e)\mathrm{Exp}\left[-\lambda_b \mathrm{Log}\left({}^2_1\boldsymbol{R}\right)\right] {}^1\boldsymbol{p}_2 \tag{17}$$

The measurement covariance propagation is then given by:

$$\boldsymbol{P}_{be} = \boldsymbol{H}_i \boldsymbol{P}_{12} \boldsymbol{H}_i{}^\top \ , \quad \boldsymbol{H}_i = \begin{bmatrix} \frac{\partial^e_b \tilde{\boldsymbol{\theta}}}{\partial^2_1 \tilde{\boldsymbol{\theta}}} & \mathbf{0}_{3\times3} \\ \frac{\partial^b \tilde{\boldsymbol{p}}_e}{\partial^2_1 \tilde{\boldsymbol{\theta}}} & \frac{\partial^b \tilde{\boldsymbol{p}}_e}{\partial^1 \tilde{\boldsymbol{p}}_2} \end{bmatrix} \tag{18}$$

where $\boldsymbol{P}_{12}$ is the covariance calculated from equation (13). For Jacobian derivation details, please see the companion

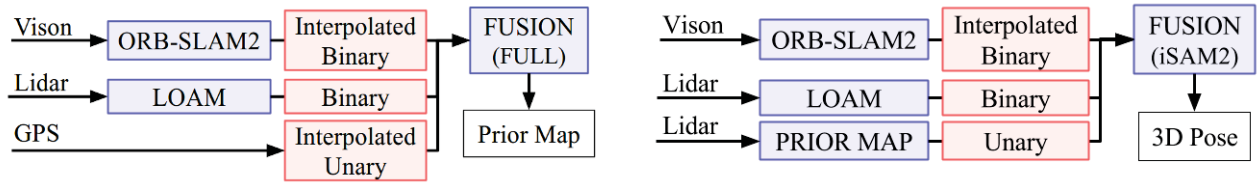

Fig. 4: Sensor measurements are first processed through an odometry module if needed (seen in blue) and then converted into factors (seen in red) that can be added to factor graph. The prior map system (left) leverages RTK GPS measurements to create a prior map in the GPS frame of reference. The GPS denied estimation system (right) uses the generated LIDAR maps to ensure that the pose estimation is in the GPS frame of reference.

tech report [14]. The resulting non-zero Jacobian matrix entries are analytically computed as:

$$\frac{\partial^e_b \tilde{\boldsymbol{\theta}}}{\partial^2_1 \tilde{\boldsymbol{\theta}}} = J_r\left[(1+\lambda_b+\lambda_e)\mathrm{Log}(^2_1\boldsymbol{R}^\top)\right](1+\lambda_b+\lambda_e)J_r^{-1}\left[\mathrm{Log}(^2_1\boldsymbol{R}^\top)\right] \quad (19)$$

$$\frac{\partial^b \tilde{\boldsymbol{p}}_e}{\partial^2_1 \tilde{\boldsymbol{\theta}}} = \Big(-(1+\lambda_b+\lambda_e)\mathrm{Exp}\left[\lambda_b \mathrm{Log}(^2_1\boldsymbol{R}^\top)\right]\lfloor {}^1\hat{\boldsymbol{p}}_2\times\rfloor J_r(\lambda_b \mathrm{Log}(^2_1\boldsymbol{R}^\top))\lambda_b J_r^{-1}(\mathrm{Log}(^2_1\boldsymbol{R}^\top))\Big) \quad (20)$$

$$\frac{\partial^b \tilde{\boldsymbol{p}}_e}{\partial^1 \tilde{\boldsymbol{p}}_2} = (1+\lambda_b+\lambda_e)\mathrm{Exp}\left[-\lambda_b \mathrm{Log}\left(^2_1\boldsymbol{R}\right)\right] \quad (21)$$

IV. SYSTEM DESIGN

A. Design Motivations

We now present a system that leverages asynchronous sensors in the application of autonomous driving and the proposed asynchronous measurement alignment. To both facilitate the flexibility of the vehicle design and reduce cost, we aim to run the system on a vehicle *without* access to a GPS unit and with low cost asynchronous sensors (i.e., those without the use of electronic triggering). We look to provide a localization solution in the GPS frame of reference without the use of a GPS sensor allowing for existing autonomous systems such as path planning and routing to continue to work as expected. To overcome this challenge, we present a unique prior LIDAR map that allows for the vehicle to both initialize and localize in the GPS frame of reference. Specifically we design a framework with two separate systems as follows:

1) Creation of an accurate prior map using a vehicle that has an additional Real Time Kinematic (RTK) GPS sensor unit.
2) GPS-denied localization leveraging the prior map to localize in the GPS frame of reference.

This framework is flexible and cost effective as only a single "collection" vehicle is needed to build the prior map that multiple lower cost vehicles can leverage. Specifically, this prior map allows for localization in the GPS frame of reference without the use of GPS measurements during runtime and can support localization in GPS-denied environments (e.g., tunnels or parking garages).

B. Prior Map

The first system we propose is one that generates an accurate prior map that can be leveraged by the second system to localize in the GPS frame of reference during autonomous operation. Shown in Figure 4, we fuse odometry measurements from openly available stereo and LIDAR modules, ORB-SLAM2 [15] and LOAM [16], respectively, with a RTK GPS unit. Both of these modules provide six degree of freedom pose estimates.[2] These modules are crucial in providing the orientation information since the RTK GPS only provides a 3D position solution. We estimate LIDAR states connected with consecutive binary factors from LOAM LIDAR odometry. Incoming pairs of pose measurements from ORB-SLAM2 are converted into relative measurements, interpolated to the closest bounding LIDAR state timesteps, and then inserted into the graph as a normal binary factor (Section III-B). To ensure that the estimated states are in the GPS frame of reference, RTK GPS measurements are first interpolated to the correct LIDAR state timestep and then inserted as a unary factor (Section III-A). Both ORB-SLAM2 visual odometry and RTK GPS measurements need to be interpolated because both sensors are asynchronous to the LIDAR sensor.

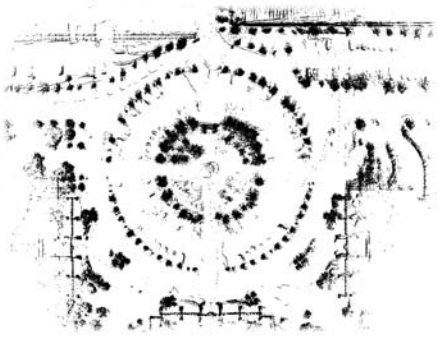

Fig. 5: Prior map generated from the experimental dataset.

The graph can be solved in real-time using an incremental solver such as iSAM2 [13] or offline with a full batch solver. It is then simple to construct a prior map using the estimated states and their corresponding LIDAR point clouds. To evaluate the overall quality of the generated prior map point cloud, the cloud is visually inspected for misalignment on environmental planes such as walls or exterior of buildings. The generated prior map from the experimental dataset can

[2]Note that both modules do *not* normally provide a corresponding covariance needed for batch optimization. We refer the reader to the appendices in the companion tech report [14]. We run LOAM in the default mode, while ORB-SLAM2 is run first in "mapping" mode to generate a map of the environment. We then use ORB-SLAM2 in "localization" mode during autonomous operation to limit jumps due to loop closures. We also note that both odometry modules must provide "to-scale" information.

be see in Figure 5.

C. GPS-Denied Map-Based Localization

Using the generated prior map, localization in the GPS frame can be performed *without* the use of a GPS sensor. As seen in Figure 4, we estimate LIDAR states that are connected with binary LIDAR factors. As in the prior map system, ORB-SLAM2 measurements are first converted to relative measurements, interpolated to bounding LIDAR state timestamps, and then inserted into the graph as a binary factor (Section III-B). In addition to these two binary factors, we perform Iterative Closest Point (ICP) matching between the newest LIDAR point cloud and the generated prior map. This ICP transform can then be added into the factor graph as a unary factor and provides constraints to the prior GPS frame of reference.

To provide real-time localization capabilities, we leverage the iSAM2 solver during GPS-denied state estimation [13]. The estimation operates at the frequency of the LIDAR sensor limited only by the speed the LOAM module can process measurements. It was found that creating unary factors using ICP matching to the prior map took upwards of 1-2 seconds. To overcome this long computation time, incoming LIDAR clouds are processed at a lower frequency in a secondary thread, and then added to the graph after successful ICP matching.

V. EXPERIMENTAL RESULTS

A. System Validation

To access the overall performance of the GPS denied system, we constructed a data collection vehicle with both a combination of low cost sensors and a RTK GPS sensor. The vehicle is equipped with a 8 channel Quanergy M8 LIDAR, ZED stereo camera, and RTK enabled NovAtel Propak6 GPS sensor. The Quanergy M8 LIDAR was run at 10Hz, while the ZED stereo camera was run at 30Hz with a resolution of 672 by 376. The RTK enabled NovAtel Propak6 GPS sensor operated at 20Hz with an average accuracy of $\pm$15 centimeters. The GPS solution accuracy allows for the creation of a high quality prior map (see Figure 5). To facilitate the GPS denied system, a dataset was first collected on the vehicle and then processed using a full batch solver. Following the proposed procedure in Section IV-B, binary LIDAR factors from LOAM, binary stereo ORB-SLAM2 factors, and unary RTK GPS factors are used. Both incoming ORB-SLAM2 and RTK GPS measurements are transformed into the correct sensor frame and interpolated before their addition into the factor graph. The resulting LIDAR point cloud, created in the GPS frame of reference, can then be used during GPS denied navigation.

To represent the real world, the GPS-denied localization system was tested on the day following the data collection for the prior map. This was to introduce changes in the environment, such as changes in car placement and shrubbery, while also showing that the prior map can still be leveraged. Using the same vehicle, RTK GPS was instead used to provide an accurate ground truth for comparison and was not used in the estimation solution. Following the proposed procedure in Section IV-C, incoming LIDAR point clouds are matched to the map generated the previous day and added to the factor graph after successful ICP alignment. Binary stereo ORB-SLAM2 factors are used after they have been transformed into the correct sensor frame and interpolated to the given LIDAR state timestamps.

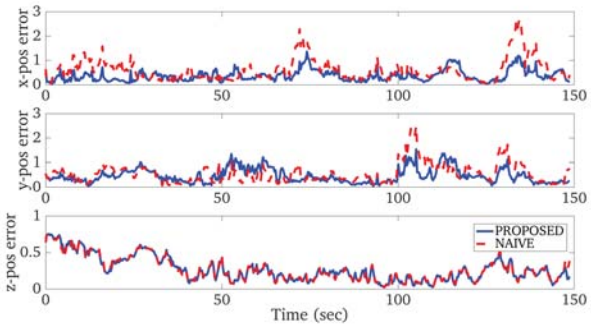

Fig. 6: Average position error magnitude (meters) over 10 runs. Error calculated relative to the RTK GPS position. Average vehicle speed of 6mph over the 500 meter long run.

For evaluation, the estimated vehicle position during online operation is compared to the corresponding output of the RTK GPS. As seen in Figure 6, when performing GPS denied localization, the system was able to remain within a stable 2 meter accuracy. We compared the proposed asynchronous measurement alignment against a naive approach of factor addition into the graph which ignores the issue of time delay and directly attaches incoming factors to the closest nodes without performing interpolation. The average RMSE was 0.93 meters for the naive approach with the proposed method having 0.71 meters (overall 23.6% decrease).

B. Evaluating the Asynchronous Measurement Alignment

Having shown that the system is able to accurately localize in real-time without the use of GPS, we next evaluated how the interpolation impacts the estimation. To do so, we disable the LIDAR prior cloud factors and instead only used odometry factors from LOAM and ORB-SLAM2.

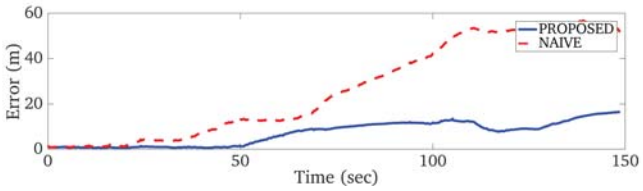

Fig. 7: Comparison of the proposed method and naive approach, over 10 runs, using pure odometry measurements.

Seen in Figure 7, the proposed asynchronous measurement alignment out performed the estimation accuracy of the naive approach. The RMSE of the naive approach was 26.74 meters and the proposed method's average error was 7.026 meters (overall 73.7% decrease). This shows that the alignement of measurements to the state timestamps through linear interpolation can greatly increase the estimation accuracy, without simultaneously increasing the graph size.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed a general approach of asynchronous measurement alignment within the graph-based optimization framework of mapping and localization in order for optimal fusion of multimodal sensors. The designed framework provides a modular system with the ability to replace individual modules and allow for any sensor that provides to-scale 3D pose estimates to be incorporated. The system has been tested on an experimental dataset and compared to a naive approach to show the improvement due to the proposed asynchronous measurement alignment. Looking forward, we will incorporate other sensors, such as Inertial Measurement Units (IMUs), through the use of analytically IMU preintegration developed in our prior work [17] to robustify the system to sensor failures and dynamic trajectories. We will also investigate how to improve the current mapping and localization, in particular, when autonomously driving in dynamic urban environments.

APPENDIX

A. Useful Identities

Given some small value $\delta\theta$ we can define the following approximations [18]:

$$\lfloor\phi\times\rfloor = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix} \tag{22}$$

$$\mathrm{Exp}(\delta\theta) \approx \mathbf{I} + \lfloor\delta\theta\times\rfloor \tag{23}$$

$$\mathrm{Exp}(\phi+\delta\theta) \approx \mathrm{Exp}(\phi)\mathrm{Exp}(J_r(\phi)\delta\theta) \tag{24}$$

$$\mathrm{Log}(\mathrm{Exp}(\phi)\mathrm{Exp}(\delta\theta)) \approx \phi + J_r^{-1}(\phi)\delta\theta \tag{25}$$

The Right Jacobian of $SO(3)$ $J_r(\phi)$ is defined by:

$$J_r(\phi) = \mathbf{I} - \frac{1-cos(\parallel\phi\parallel)}{\parallel\phi\parallel^2}\lfloor\phi\times\rfloor + \frac{\parallel\phi\parallel - sin(\parallel\phi\parallel)}{\parallel\phi\parallel^3}\lfloor\phi\times\rfloor^2 \tag{26}$$

and the Right Jacobian inverse of $SO(3)$, $J_r^{-1}(\phi)$, is:

$$J_r^{-1}(\phi) = \mathbf{I} + \frac{1}{2}\lfloor\phi\times\rfloor + \left(\frac{1}{\parallel\phi\parallel^2} - \frac{1+cos(\parallel\phi\parallel)}{2\parallel\phi\parallel sin(\parallel\phi\parallel)}\right)\lfloor\phi\times\rfloor^2 \tag{27}$$

B. Relative Measurement Static Transformation

Given a relative 3D pose measurement in a given sensor frame we would like to convert it to the same sensor frame of reference as our state (e.g., ORB-SLAM2 relative measurements are in the left camera sensor frame but the state is in the LIDAR sensor frame). Given this relative transform $\{{}^{C2}_{C1}\boldsymbol{R}, {}^{C1}\boldsymbol{p}_{C2}\}$ in the camera frame and a corresponding covariance $\boldsymbol{P}_{C12}$, we would like to transform it as follows:

$${}^{L2}_{L1}\boldsymbol{R} = {}^{L}_{C}\boldsymbol{R}\ {}^{C2}_{C1}\boldsymbol{R}\ {}^{L}_{C}\boldsymbol{R}^{\top} \tag{28}$$

$${}^{L1}\boldsymbol{p}_{L2} = {}^{L}_{C}\boldsymbol{R}\left({}^{C2}_{C1}\boldsymbol{R}^{\top}\ {}^{C}\boldsymbol{p}_L + {}^{C1}\boldsymbol{p}_{C2} - {}^{C}\boldsymbol{p}_L\right) \tag{29}$$

where we define the LIDAR frame of reference as $\{Li\}$, $i \in \{1,2\}$ and the camera frame of reference as $\{Ci\}$, $i \in \{1,2\}$. It is assumed that the static transform, $\{{}^{L}_{C}\boldsymbol{R}, {}^{C}\boldsymbol{p}_L\}$, from the LIDAR to camera frame of reference are known from offline calibration. The covariance in the LIDAR frame is as follows:

$$\boldsymbol{P}_{L12} = \boldsymbol{H}_s\boldsymbol{P}_{C12}\boldsymbol{H}_s{}^{\top} \tag{30}$$

where $\boldsymbol{P}_{C12}$ is the relative camera covariance. For Jacobian derivations, please see the companion tech report [14]. The resulting Jacobian matrix $\boldsymbol{H}_s$ is defined as the following:

$$\boldsymbol{H}_s = \begin{bmatrix} {}^{L}_{C}\boldsymbol{R} & \mathbf{0}_{3\times3} \\ -{}^{L}_{C}\boldsymbol{R}\ {}^{C2}_{C1}\boldsymbol{R}^{\top}\lfloor{}^{C}\boldsymbol{p}_L\times\rfloor & {}^{L}_{C}\boldsymbol{R} \end{bmatrix} \tag{31}$$

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332.

[3] Z. Taylor and J. Nieto. "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation". In: *IEEE Transactions on Robotics* 32.5 (2016), pp. 1215–1229.

[4] F. Dellaert and M. Kaess. "Square Root SAM: Simultaneous localization and mapping via square root information smoothing". In: *The International Journal of Robotics Research* 25.12 (2006), pp. 1181–1203.

[5] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. "Information fusion in navigation systems via factor graph based incremental smoothing". In: *Robotics and Autonomous Systems* 61.8 (2013), pp. 721–738.

[6] N. Sünderhauf, S. Lange, and P. Protzel. "Incremental Sensor Fusion in Factor Graphs with Unknown Delays". In: *Advanced Space Technologies in Robotics and Automation (ASTRA)*. 2013.

[7] S. L. Alonso Patron-Perez and G. Sibley. "A Spline-Based Trajectory Representation for Sensor Fusion and Rolling Shutter Cameras". In: *International Journal of Computer Vision* 113.3 (2015), pp. 208–219.

[8] C. X. Guo, D. G. Kottas, R. DuToit, A. Ahmed, R. Li, and S. I. Roumeliotis. "Efficient Visual-Inertial Navigation using a Rolling-Shutter Camera with Inaccurate Timestamps." In: *Robotics: Science and Systems*. Citeseer. 2014.

[9] S. Ceriani, C. Sánchez, P. Taddei, E. Wolfart, and V. Sequeira. "Pose interpolation SLAM for large maps using moving 3D sensors". In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE. 2015, pp. 750–757.

[10] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. "g^{2}o: A general framework for graph optimization". In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011.

[11] K. Eckenhoff, L. Paull, and G. Huang. "Decoupled, Consistent Node Removal and Edge Sparsification for Graph-based SLAM". In: *in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. Oct. 2016.

[12] F. Dellaert. *Factor graphs and GTSAM: A hands-on introduction*. Tech. rep. 2012.

[13] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering". In: *2011 IEEE International Conference on Robotics and Automation*. May 2011.

[14] P. Geneva, K. Eckenhoff, and G. Huang. *Asynchronous Multi-Sensor Fusion for 3D Mapping and Localization*. Tech. rep. RPNG-2017-ASYNC. Available: `http://udel.edu/~ghuang/papers/tr_async.pdf`. University of Delaware, 2017.

[15] R. Mur-Artal and J. D. Tardos. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *arXiv preprint arXiv:1610.06475* (2016).

[16] J. Zhang and S. Singh. "LOAM: Lidar Odometry and Mapping in Real-time." In: *Robotics: Science and Systems*. Vol. 2. 2014.

[17] K. Eckenhoff, P. Geneva, and G. Huang. "High-Accuracy Preintegration for Visual-Inertial Navigation". In: *Proc. of International Workshop on the Algorithmic Foundations of Robotics*. San Francisco, CA, Dec. 2016.

[18] T. D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.