Supplemental material to: ”SIGNAV: Semantically-Informed GPS-Denied Navigation and Mapping in Visually-Degraded Environments”

Alex Krasner*, Mikhail Sizintsev*, Abhinav Rajvanshi, Han-Pang Chiu,
Niluthpol Mithun, Kevin Kaighn, Philip Miller, Ryan Villamil, Supun Samarasekera
SRI Internaional, USA

1. Factor Formulation for SIGNAV

In this section, we describe the details of the factor graph formulation for SIGNAV.

Factor graphs [4] are graphical models that are well suited to modeling complex estimation problems, such as SLAM (Simultaneous Localization and Mapping). A factor graph represents the joint probability distribution as a bipartite graph $G = (F, X, E)$ with two node types: factor nodes $f_i \in F$ and state variable nodes $x_j \in X$. An edge $e_{ij} \in E$ exists if and only if factor f_i involves state variables x_j.

There are two kinds of state variable nodes in our formulation for SLAM systems: The navigation state nodes includes the platform information (such as pose and velocity) at all given time steps, while the landmark states encodes the estimated 3D position of external visual landmarks. We define the navigation state of the platform at time i as $x_i = (p_i, v_i, b_i)$. Each state x covers three kinds of nodes: the pose node p includes 3D translation $\mathbf{t}$ and 3D rotation R, the velocity node v represents 3D velocity in the world coordinate system, and b denotes sensor-specific bias block which are varied for different sensors. Note that the 3D rotation R represents the rotation from the world coordinate system to the local body's coordinate system, while the 3D translation $\mathbf{t}$ represents the position of the local coordinate system's origin in the world coordinate system. Pose node and velocity node are included in every navigation state, while sensor-specific bias block are propagated only through the navigation states which have measurements from the correspondent sensor. To simplify the notation, we assume all sensors have the same center (i.e. calibrated in such a way), which is the origin of the body coordinate system.

Sensor measurements are then formulated into factor representations. The connectivity of each factor - how the factor node connects to correspondent state nodes - is decided based on how a measurement affects the appropriate state variables. For example, a GPS position measurement only involves a navigation state x at a single time. A camera feature observation can involve both a navigation state x and a state of unknown 3D landmark position $\mathbf{l}$.

Our framework represents a sensor measurement as a factor affecting state variables. A generative model

$$z_i = h_i(X_i) + w_i \tag{1}$$

predicts a sensor measurement z_i using a function $h_i(X_i)$ with measurement noise w_i.

There are many efficient solutions to solve the inference process using this factor graph representation for real-time SLAM systems. One popular solution is iSAM2 [3], which uses a Bayes tree data structure to keep all past information and only updates variables influenced by each new measurement. For the details on the factor graph representation and its inference process for SLAM systems, we refer to [1].

SIGNAV fuses sensor measurements from four sensor modalities (IMU, cameras, LiDAR, and wheel odometry) for navigation using this factor graph framework as follows.

1.1. IMU

A single factor typically encodes only one sensor measurement. However, IMU sensors produce measurements at a much higher rate than other sensor types. Therefore, we formulate a single factor to summarize multiple consecutive IMU measurements between two navigation states. A navigation state is only created at the time when a non-IMU measurement comes, and a binary factor is built to connect two sequential navigation states by integrating IMU measurements between them:

$$z_i = h_i(x_{i-1}, x_i) + w_i. \tag{2}$$

We formulate this factor using a pre-integrated IMU mechanism [2]. The IMU factor generates 6 degrees of freedom relative pose and corresponding velocity change between time $i - 1$ and i. It also tracks the IMU-specific bias as part of the state variables, assuming a random-walk

*The first two authors contributed equally to this work. All authors are with Center for Vision Technologies, SRI International, USA. The contact author is Han-Pang Chiu {han-pang.chiu@sri.com).

model for the IMU bias evolution. This way avoids the dynamic modeling of the complex kinematics associated with chaotic or rapid movements, and replaces the system dynamics with a motion model derived from IMU propagation. The error dynamics evolve with lower frequency and thus make estimation equations better suited for linearization. It allows for better handling of the uncertainty propagation through the whole system.Therefore, the IMU factor also provides the linearization point for the navigation state at the current time i which is necessary for all other factors to be linearized. In contrast to tradition filtering techniques, the IMU motion factor is part of the nonlinear optimization leading to improved accuracy, while we avoid the burden of repeated integrations by using IMU pre-integration.

1.2. Camera

In our system, tracked visual features extracted from camera images are modeled by an extrinsic factor type. This extrinsic factor estimates both navigation states and the 3D location of the associated visual landmark. The measurement model for the extrinsic factor between the landmark state $\mathbf{l}$ and the pose state x_i is:

$$z_i = h_i(x_i) + w_i = Proj(\mathsf{R}_i(\mathbf{l} - \mathbf{t}_i)) + w_i, \qquad (3)$$

where z is the normalized pixel describing the projection of the 3D landmark $\mathbf{l}$ onto the camera at time i, the platform pose $p_i = (\mathsf{R}_i, \mathbf{t}_i)$ from x_i is used to transform the landmark to the local (camera) frame, and w_i is the pixel noise. The function $Proj(\mathbf{m})$ is the function that projects a 3D point $\mathbf{m}$ onto the image plane.

For the details of the linearized model and the corresponding Jacobian matrices, we refer the reader to [3].

1.3. LiDAR

SIGNAV registers 3D LiDAR points obtained from sequential scans (scan-to-scan registration). A 3D related pose measurement across sequential scans is then generated and formulated as a binary factor which involves only pose nodes p_{i-1} and p_i.

$$z_i = h_i(x_{i-1}, x_i) + w_i, \qquad (4)$$

where the function h extracts the 3D pose p from the state x. The detailed formulation of the Jacobian is in [5].

1.4. Wheel Odometry

We utilize each wheel odometry measurement as a 3D velocity vector. Specifically, this measurement reports the 3D velocity of the moving platform in the world coordinate system at the current time. It is formulated as a unary factor to the velocity node v_i in current state x_i. This factor is modeled as follows:

$$z_i = h_i(x_i) + w_i, \qquad (5)$$

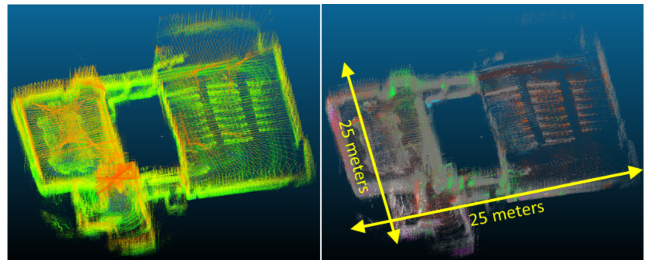

Figure 1. The 3D maps generated from SIGNAV in Scenario 3: (left) the 3D map, and (right) the 3D semantic map. Different colors in the 3D semantic map represent different semantic classes.

where the function h extracts the 3D velocity from the state x_i. The linearized model, including the measurement Jacobian H can be simply computed as:

$$\delta z = \mathsf{H}\delta v_i + w, \qquad \mathsf{H} = \mathsf{I}_{3\times 3}. \qquad (6)$$

2. Additional 3D Mapping Results

In the paper, due to page limitation, we are only able to include the 3D maps from three (Scenario 1, Scenario 2, Scenario 4) of four scenarios conducted by SIGNAV. Here we show the 3D maps (Figure 1) constructed from Scenario 3. This scenario is to navigate across two conference rooms (conduct two loops - with partially different routes - in each room): one room is totally dark (right room), while the other room has little light from outside (left room). From the visualization of the 3D map, we can see the chairs (red color in the semantic map in Figure 1) are reconstructed and separated nicely in the dark room.

References

[1] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. In *Georgia Institute of Technology*, 2012.

[2] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems (RSS)*, 2015.

[3] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research*, 31:217–236, 2012.

[4] F. Kschischang, B. Fey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory*, 47(2), 2001.

[5] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, pages 3565–3572, Rome, Italy, April 2007.