The FPV Drone Racing VIO Competition: Performance of SVIn2 – a multi-sensor fusion based SLAM system with loop closing and relocalization

Sharmin Rahman[1], Alberto Quattrini Li[2], and Ioannis Rekleitis[1]

***Abstract*— The report gives an overview of our Simultaneous Localization and Mapping (SLAM) system, SVIn2, used for participating in the IROS 2019 FPV Drone Racing VIO Competition, and presents the main experimental setup and results.**

I. Overview of the Proposed Method

SVIn2 [1] augmented the state-of-the-art visual-inertial state estimation package OKVIS [2] to accommodate different sensors in a *non-linear optimization*-based framework and added the *loop losing and relocalization module* to address the drift introduced in sliding window and marginalization based methods. Fig. 1 shows the architecture of SVIn2.

While SVIn2 is originally targeted for underwater domain – where it operates with Sonar, inertial, depth (water pressure) sensors, and stereo camera – it can be easily configured for other scenarios where some of the sensors are not available. The results on the UZH-FPV drone racing datasets are obtained by disabling sonar and depth sensor and *only using the visual-inertial information*.

In the following, we report the problem formulation, that includes loop closing and relocalization; please see [1] for complete details.

A. Notations and States

In our proposed system, the reference frames are denoted as C for Camera, I for IMU, D for Depth, S for Sonar, and W for World. Let us denote ${}_X\mathbf{T}_Y = [{}_X\mathbf{R}_Y | {}_X\mathbf{p}_Y]$ the homogeneous transformation matrix between two arbitrary coordinate frames X and Y, where ${}_X\mathbf{R}_Y$ denotes the rotation matrix with corresponding quaternion ${}_X\mathbf{q}_Y$ and ${}_X\mathbf{p}_Y$ represents the position vector.

Let us now define the robot R state $\mathbf{x}_R$ that the system is estimating as:

$$\mathbf{x}_R = [{}_W\mathbf{p}_I^T, {}_W\mathbf{q}_I^T, {}_W\mathbf{v}_I^T, {\mathbf{b}_g}^T, {\mathbf{b}_a}^T]^T \quad (1)$$

which contains the position ${}_W\mathbf{p}_I$, the attitude represented by the quaternion ${}_W\mathbf{q}_I$, the linear velocity ${}_W\mathbf{v}_I$, all expressed as the IMU reference frame I with respect to the world coordinate W; moreover, the state vector contains the gyroscopes and accelerometers bias $\mathbf{b}_g$ and $\mathbf{b}_a$.

[1]S. Rahman and I. Rekleitis are with the Computer Science and Engineering Department, University of South Carolina, Columbia, SC, USA, srahman@email.sc.edu, yiannisr@cse.sc.edu

[2]A. Quattrini Li is with the Department of Computer Science, Dartmouth College, Hanover, NH, USA, alberto.quattrini.li@dartmouth.edu

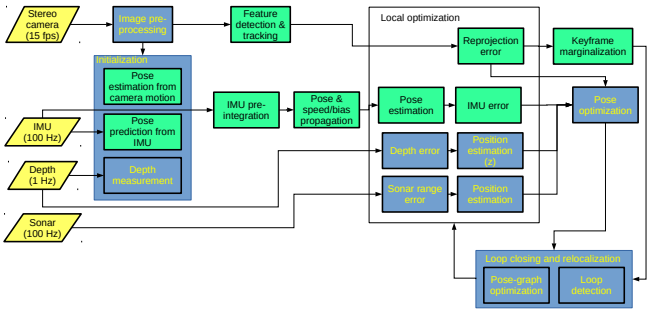


Fig. 1. Overview of the proposed approach, SVIn2; in yellow are the sensor feeds and their frequency; in green the OKVIS [2] components; in blue the components introduced to handle acoustic and depth data, underwater visual effects, and loop closure. [3], [1]

The associated error-state vector is defined in minimal coordinates, while the perturbation takes place in the tangent space of the state manifold. The transformation from minimal coordinates to tangent space can be done using a bijective mapping [2], [4]:

$$\delta\boldsymbol{\chi}_R = [\delta\mathbf{p}^T, \delta\boldsymbol{\alpha}^T, \delta\mathbf{v}^T, \delta{\mathbf{b}_g}^T, \delta{\mathbf{b}_a}^T]^T \quad (2)$$

which represents the error for each component of the state vector with $\delta\boldsymbol{\alpha} \in \mathbb{R}^3$ being the minimal perturbation for rotation.

B. Tightly-coupled Non-Linear Optimization with Sonar-Visual-Inertial-Depth measurements

For the tightly-coupled non-linear optimization, we use the following cost function $J(\mathbf{x})$, which includes the reprojection error $\mathbf{e}_r$ and the IMU error $\mathbf{e}_s$ with the addition of the sonar error $\mathbf{e}_t$ (see [3]), and the depth error e_u:

$$\begin{aligned} J(\mathbf{x}) = & \sum_{i=1}^{2}\sum_{k=1}^{K}\sum_{j\in\mathcal{J}(i,k)} {\mathbf{e}_r^{i,j,k}}^T \mathbf{P}_r^k \mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1} {\mathbf{e}_s^k}^T \mathbf{P}_s^k \mathbf{e}_s^k \\ & + \sum_{k=1}^{K-1} {\mathbf{e}_t^k}^T \mathbf{P}_t^k \mathbf{e}_t^k + \sum_{k=1}^{K-1} {e_u^k}^T P_u^k e_u^k \end{aligned} \quad (3)$$

where i denotes the camera index – i.e., left ($i = 1$) or right ($i = 2$) camera in a stereo camera system with landmark index j observed in the k^{th} camera frame. $\mathbf{P}_r^k$, $\mathbf{P}_s^k$, $\mathbf{P}_t^k$, and P_u^k represent the information matrix of visual landmarks, IMU, sonar range, and depth measurement for the k^{th} frame respectively.

For completeness, we briefly discuss each error terms, please refer to [2] and [3], [1] for more details.

The reprojection error is calculated based on the difference between a keypoint measurement in the camera coordinate frame C and the corresponding landmark back-projection based on the stereo projection model. The IMU error term combines all the accelerometer and gyroscope measurements utilizing the *IMU pre-integration* approach described by Forster *et al*. [4] between successive camera frames and represents the *pose*, *speed*, and *bias* error between the prediction based on the previous and the current states. The IMU kinematics are used to predict the current state based on the previous state. Both the reprojection error and the IMU error term follow the formulation described by Leutenegger *et al*. [2].

The sonar range error, introduced in our previous work [3], represents the difference between the 3D point that can be derived from the range measurement and a corresponding visual feature in 3D. In poor visibility and low contrast environment where vision fails to detect features, Sonar provides additional features and helps in mapping the surroundings. The depth error term can be calculated as the difference between the rig position along the z direction and the water depth measurement provided by a pressure sensor. Depth values are extracted along the *gravity* direction which is aligned with the z of the world W – observable due to the tightly coupled IMU integration. This can correct the position of the robot along the z axis. For the detailed formulation of the above error terms, please refer to our previous work [3], [1].

Ceres Solver nonlinear optimization framework [5] optimizes $J(\mathbf{x})$ then in a sliding window to estimate the state of the system.

Loop-closing and *relocalization* is achieved using the binary bag-of-words place recognition module DBoW2 [6]. In a sliding window and marginalization based method a global optimization and relocalization scheme is necessary to eliminate the drift that accumulates over time. In our system, a pose-graph maintains the connections between keyframes where a node represents a keyframe and an edge between two keyframes exists if there is sufficient overlap between them. With every new frame in the local window, the loop-closing module searches for loop candidates in the BoW database. When a candidate is found with enough match, feature correspondences are obtained to establish connection between the current frame and the loop candidate frame. Then, a PnP RANSAC is performed to obtain the geometric validation. The relocalization module is responsible for aligning the current keyframe pose in the local window with the loop candidate keyframe by sending the drift in pose to the windowed sonar-visual-inertial-depth optimization thread.

II. RESULTS

Experiments were conducted for each dataset on a computer with an Intel i3-6100U CPU @ 2.30GHz (2 cores, 4 threads), 32 GB RAM, running Ubuntu 16.04 and ROS Kinetic. Stereo camera and IMU have been used to estimate the trajectory for all datasets. After some initial experiments to properly tune the parameters, the same set of parameters (e.g, maximum number of keypoints per image, minimum number of matches to detect a loop) is used throughout all the sequences. Every sequence has been run in real-time, showing that the method is capable to run on a small computing unit at the same rate as the sensor data.

Figures 2, 3, and 4 show the trajectories over the 6 datasets selected from the UZH FPV Drone Racing Dataset [7]

III. CONCLUSIONS

REFERENCES

[1] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor," in *Intelligent Robots and Systems (IROS)*, 2019, (accepted).

[2] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

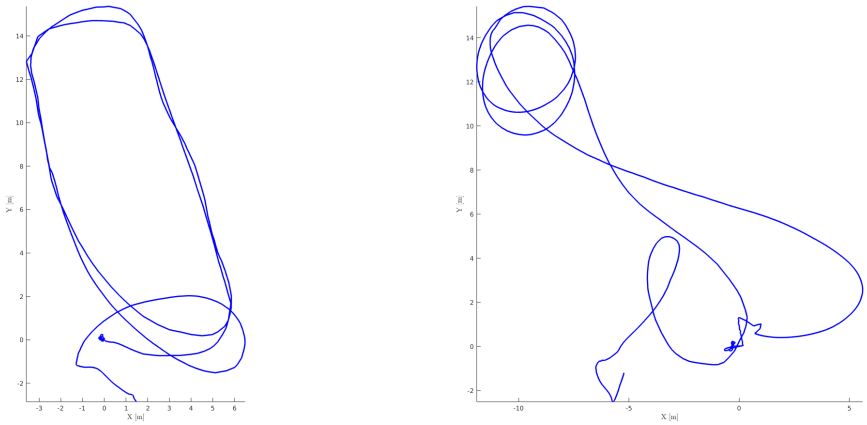
[3] S. Rahman, A. Quattrini Li, and I. Rekleitis, "Sonar Visual Inertial SLAM of Underwater Structures," in *Proc. ICRA*, 2018.

[4] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, 2017.

[5] S. Agarwal, K. Mierle, and Others, "Ceres Solver," http://ceres-solver.org, 2015.

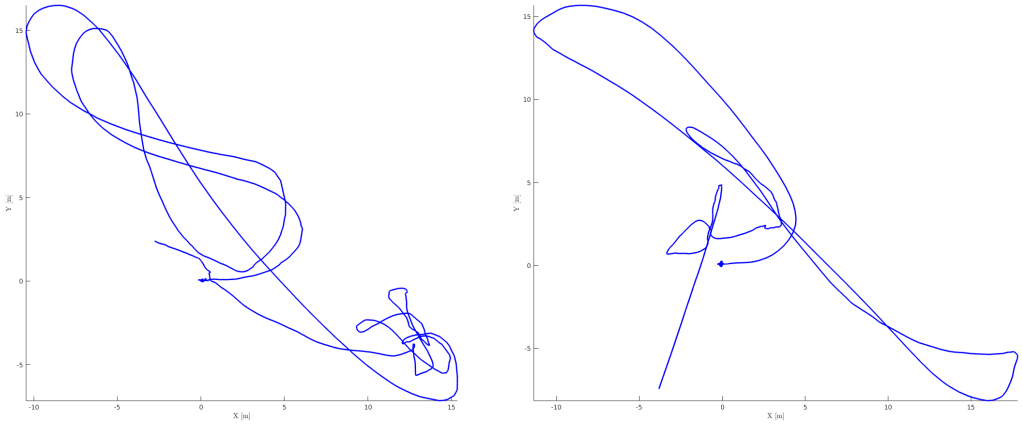
[6] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, 2012.

[7] "The FPV Drone Racing VIO Competition," https://github.com/uzh-rpg/IROS2019-FPV-VIO-Competition, 2019.



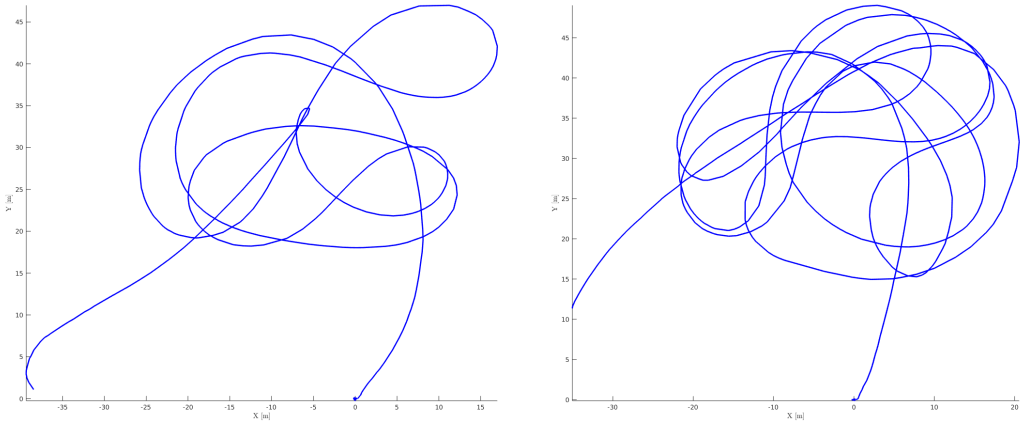
(a) (b)

Fig. 2. Indoor 45 degree downward facing: (a) sequence 3 (b) sequence 16



(a) (b)

Fig. 3. Indoor forward facing: (a) sequence 11 (b) sequence 12



(a) (b)

Fig. 4. Outdoor forward facing: (a) sequence 9 (b) sequence 10