Visual Inertial Odometry using Focal Plane Binary Features (BIT-VIO)

Matthew Lisondra[1,*], Junseo Kim[1,*], Riku Murai[2], Kourosh Zareinia[1], and Sajad Saeedi[1]

***Abstract*—Focal-Plane Sensor-Processor Arrays (FPSP)s are an emerging technology that can execute vision algorithms directly on the image sensor. Unlike conventional cameras, FPSPs perform computation on the image plane – at individual pixels – enabling high frame rate image processing while consuming low power, making them ideal for mobile robotics. FPSPs, such as the SCAMP-5, use parallel processing and are based on Single Instruction Multiple Data (SIMD) paradigm. Here, we present BIT-VIO, the first-ever 6-Degrees of Freedom (6-DOF) Visual Inertial Odometry (VIO) algorithm using binary features computed on the focal plane using SCAMP-5. The BIT-VIO algorithm is tested on a moving robotic system, operating and correcting by loosely-coupled iterated Extended Kalman Filter (iEKF) at 300 FPS with an IMU at 400 Hz. We evaluate BIT-VIO experimentally, demonstrating its performance. The code/data will be made available online.[3]**

I. Introduction

The need for a lower latent, more efficient, and lower power usage camera technology grows increasingly important when dealing with vision-IMU-fused state estimation algorithms. These are known under Visual Inertial Odometry (VIO) systems. Conventional camera technology is used on these systems, most typically operating at 40-80 frames per second (FPS), where they generally take in information from their environment, transferring all data over to digital processing hardware such as a PC host or external computer where it is then further processed and used in vision-based applications such as Visual Simultaneous Localization and Mapping (VSLAM). High computational power is often necessary, and processing resources on the PC are used extensively.

The Focal-Plane Sensor-Processor Arrays (FPSP)s are a new technology where much of the computational load can be offput by allowing image and signal processing to be done on the imager's focal plane before transferring to a PC host or other external device to be further processed if still necessary [6]. Such compression on the focal plane is desirable for Visual Odometry (VO) and VIO as it provides low latency and low power consumption.

Extending on the previous work BIT-VO [25], we present BIT-VIO, the first 6-Degrees of Freedom (6-DOF) Visual Inertial Odometry (VIO) algorithm to utilize the advantages of the FPSP for vision-IMU-fused state estimation. Fig. 1 shows an example where the proposed BIT-VIO algorithm is compared with independent inertial and visual estimates, all overlaid on the ground-truth trajectory. The contributions of the work are:

(I) Efficient Visual Inertial Odometry operating and correcting by loosely-coupled sensor-fusion iterated Extended Kalman Filter (iEKF) at 300 FPS with an IMU at 400 Hz. (II) Uncertainty propagation for BIT-VO's pose as it is based on binary-edge-based descriptor extraction, 2D to 3D reprojection, and transform from pose to pose. (III) Comparison with BIT-VO and ground-truth, tested on a moving robotic system against a motion capture system.

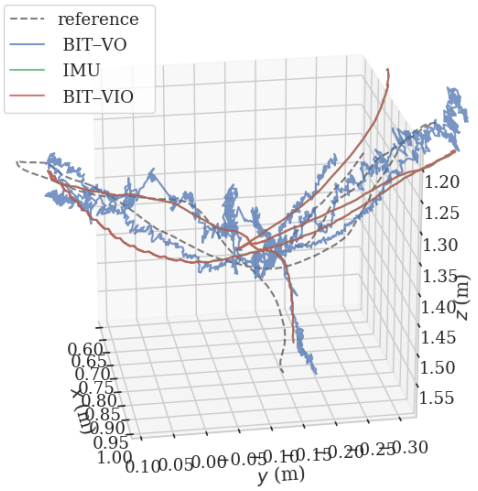

Fig. 1. Comparison of the proposed BIT-VIO with inertial (IMU) and visual odometry (BIT-VO) overlaid on the reference ground-truth trajectory. Our BIT-VIO estimates are closer to the ground-truth compared to predictions from IMU or BIT-VO. Notice that our BIT-VIO effectively removes the high-frequency noise visible in BIT-VO's trajectory. The plot was generated using evo [11]. This is Traj. K from Table I.

The remainder of this work is organized as follows. Section II presents the literature review. Section III describes the background about SCAMP-5 FPSP. Section IV explains the proposed BIT-VIO algorithm. Section V details our experimental results. Finally, Section VI concludes the work.

II. Literature Review

Most filtering-based VIO algorithms are based on efficient state estimation by updating only the last state. By filters, we mean various implementations of Bayesian filtering for the state estimation, such as the Extended Kalman Filter [28]. Classical approaches estimate both the pose and landmarks, and the complexity grows quadratically based on the number of these estimated landmarks. The work from Davison et al. [4] was the first where a few landmarks from a conventional camera system, operating at low 20-30 FPS, were tracked to allow real-time operation. Later, a structureless approach, Multi-State Constraint Kalman Filter (MSCKF), where landmark positions are marginalized out of the state-vector, was done [23]. The processing needed to be delayed until all

[1]Toronto Metropolitan University (TMU).
[2]Imperial College London, Department of Computing.
*Both authors contributed equally to this research at TMU.
[3]https://sites.google.com/view/bit-vio/home

measurements of a single landmark are obtained, which affects accuracy greatly as the filter cannot use all of the present visual information due to the real-time constraints and limited resources available on a robotic system.

There is also fixed-lag smoothing VIO, based on optimization via a window of states, and often requires a nonlinear optimization approach to output the state [28]. They use conventional camera technology operating at a low frame rate. Their benefit and advantage over filtering-based VIO is they allow re-linearization and are still fast [28]. They also are generally more accurate than the filtering VIO method. Using robust cost functions or more explicit outlier rejection after optimization makes these approaches more robust to outliers. With fixed-lag-smoothing-based VIO still resorting to marginalization of the state, like with filter-based VIO, they still suffer from some inconsistency and linearization errors [12], [5], [13]. Marginalization of states outside the estimation window can also lead to dense Gaussian priors, which is bad on the computational front when matrix operations are done. To alleviate this, some and certain measurements are dropped, as was proposed by Leutenegger et al. [17]. However, all this is still done extensively on a PC host or external computer. The computational burden on the system is again a disadvantage, too, and even more so with fixed-lag smoothing VIO, as optimization is too heavy of a task.

Another VIO approach is a full-smoothing algorithm that triumphs over the last two [28] in terms of accuracy. These VIO algorithms optimize all states of the full trajectory of the robotic system and do so, similar to the last approach, on nonlinear optimization [28]. They have the added benefit of re-linearization and deal with sparse matrices. They have the highest accuracy of the three VIO approaches but worse power usage. The major disadvantage is that they are generally slower and inefficient and take more computational power and resources than the other two. The complexity of the optimization problem is approximately cubic with respect to the dimension of the states. This limits its use in real-time robotic systems as the map grows over time. What can be done is only to keep selected keyframes, as was proposed by Leutenegger et al. [17], Qin et al. (VINS-Mono [26]), and/or run the optimization in a PTAM (parallel tracking and mapping) architecture [24]. A major leap has been the development of incremental smoothing methods such as iSAM [15] and iSAM2 [14]. They are based on factor graphs to identify and update only a typically small subset of variables affected by a new measurement.

There exist other camera technologies used in VO and VSLAM, such as event-based cameras, used in many state-of-the-art VIO algorithms especially, based on the illumination and changes in light [18]. Event cameras, in particular, also provide low power usage and low latency benefits over conventional cameras. However, there are two downsides: First, while event cameras compress visual information into a continuous stream of events, they don't offer the flexibility of the SCAMP-5, which supports user-programmed algorithms like the FAST-corner or edge-coordinates detector [27]. Second, the data volume transferred by an event camera is proportional to camera motion. This characteristic is not optimal; for instance, a robot has a shorter window during rapid movements to determine its subsequent action.

III. Background: SCAMP-5 FPSP

A particular FPSP technology of great interest is the SCAMP-5 camera technology [6], [2], which contains a 256×256 processor array, doing parallel processing in a SIMD fashion in the imaging sensor. The parallelism architecture of the SCAMP-5 FPSP camera technology helps to provide large computing power and high efficiency. Each Pixel has a processing element (PE) that contains 7 analog registers, 13 digital registers, and an ALU, which means they can perform logical and arithmetic operations. The analog registers can store a real-valued variable, storing around 250 different quantity values, like of the 8-bit nature. The analog registers can do operations such as add, negate, split, and compare-against-0. The digital registers can store 13 binary values, with one being special in having influence over the analog registers. The digital registers can do MOV, OR, NOR, and NOT operations. Each PE can communicate EAST, WEST, NORTH, and SOUTH with its neighboring PEs analog registers and digital registers.

The SCAMP-5 FPSP camera technology is designed to have additional benefits on top of its parallel architecture of SIMD [6]. As each PE contains 13 digital registers, events can be read out, and the coordinates of the events can be scanned, so the time cost is only proportional to the number of events [2]. In a conventional camera, the time cost would have been much more proportional to the scanning area taken. Next, flooding is digital register propagation of 1s used for hardware acceleration. Through the SCAMP-5 FPSP camera technology's asynchronous propagation network, the flooding speed is much faster than in a conventional camera, almost $62\times$ more. In essence, the SCAMP-5 FPSP camera technology has the advantage of providing lower latency with lower power usage on its PC host or external computer.

A. Robotic Implementation

The SCAMP-5 has already been utilized on many robotic systems. Early works such as [8] did tracking of a ground target using the FPSP, which was then used for a small, agile quadrotor UAV. Greatwood et al. worked to perform HDR edge detection odometry on the FPSP [9]. It was ground-truth compared with a motion capture system for different trajectories via different lighting and movement conditions. The FPSP outperformed when the motion blur of conventional cameras became a problem and did so with lower power consumption [9]. [22] did work on visual odometry using FPSPs for unmanned UAVs in GPS-denied environments. The advantage and now first-hand aid and application are much more apparent. HDR on the FPSP was fully utilized and allowed the UAV to transition from outdoor to indoor environments whilst still being able to track, despite the light transitioning conditions [22]. [19] dealt with direct servo control from in-sensor CNN inference from the FPSP. Doing a classic rock, paper, scissors classification problem at 8000 FPS, the FPSP at low latency acted in play. Work of [1]

dealt with visual route mapping and localization on the FPSP. This work runs at more than 300 FPS on various large-scale datasets. This was the first on-sensor mapping and localization robotic system all on a sensor. Other works, such as [10], perform drone racing, using the FPSP to detect the gates. The gate size and location are the only data that is transferred, with minimal data transfer resulting in 500 FPS. [3] implemented a recurrent neural network (RNN) for obstacle avoidance that uses features extracted from conventional vision algorithms like motion parallax static and dynamic corners. They were able to achieve 250 FPS. AnalogNavNet [29] utilized CAIN to implement a CNN for robotic navigation inside a corridor and racetrack environment.

IV. Proposed Method

In this section, first, the notations are introduced, then an overview of the system is presented, followed by the IMU model, the measurement model, and the fusion.

A. Notations

The following notation conventions are used in this work:

- Units of a variable A as $[A]$ (ex. $[a_x] = m/s^2$).
- Skew-symmetric matrix of A is $\lfloor A \rfloor$.
- p_A^B represents the translation from frames $A \rightarrow B$.
- q_A^B represents the Hamiltonian quaternion rotation $(q_{A\,w}^B, q_{A\,x}^B, q_{A\,y}^B, q_{A\,z}^B)$ from frames $A \rightarrow B$.
- $\hat{p}, \hat{q}$ are the expected translation and rotation.
- $\tilde{p}, \tilde{q}$ are the error in translation and rotation.
- $C_{(q)}$ is the rotational matrix to the quaternion q.
- $\Omega(\omega)$ is quaternion-multiplication matrix of ω.
- $\delta q = q \otimes \hat{q} \approx \left[\frac{1}{2}\delta\Theta^T, 1\right]^T$ approx. for quaternion δq.

Fig. 2 shows the coordinate frames used in this work.

B. System Overview

Fig. 3 demonstrates an overview of the system. The visual odometry part is shown on the right, and the inertial part is shown on the left. The algorithmic components of the visual odometry are mostly done on a remote host, except the corner/edge detection, which is done on the SCAMP-5 FPSP. Details of the inertial and visual processing and modeling of the system are described next.

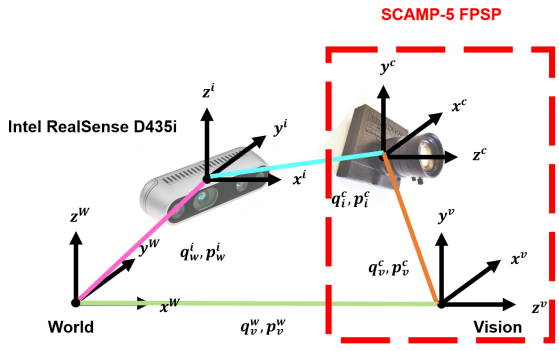

Fig. 2. IMU and FPSP frames. Intel RealSense D435i IMU is used in this work, and SCAMP-5 is the FPSP used as a camera sensor. Four coordinate frames, with 2 being a part of SCAMP-5 FPSP (camera and vision frames). p_A^B and q_A^B are between frames A, B.

C. IMU Model and State Prediction

From the iterated Extended Kalman Filter MultiSensor Fusion framework (iEKF-MSF) [21], assume absolute IMU measurements have bias b_ω, b_a with Gaussian noise n_ω, n_a. IMU outputs angular velocities ω and linear accelerations a and we have in the IMU-frame,

$$\omega = \omega_{\text{meas}} - b_\omega - n_\omega\,, \quad \dot{b}_\omega = n_{b_\omega} \tag{1}$$

$$a = a_{\text{meas}} - b_a - n_a\,, \quad \dot{b}_a = n_{b_a} \tag{2}$$

The subscript "meas" means the measured value. Terms $\dot{b}_\omega, \dot{b}_a$ are the dynamic models of the IMU biases.

The iEKF-MSF states x are represented in two parts: the x_{IMU}^T and x_{BIT-VO}^T parts of the states. The x_{IMU}^T, which is a 16-element state, is driven by the IMU measurements and dynamic models:

$$x_{IMU}^T = [{p_w^i}^T, {v_w^i}^T, {q_w^i}^T, b_\omega^T, b_a^T] \tag{3}$$

$$\dot{p}_w^i = v_w^i \tag{4}$$

$$\dot{v}_w^i = C_{(q_w^i)}^T a - g \tag{5}$$

$$\dot{q}_w^i = \frac{1}{2}\Omega\omega q_w^i \tag{6}$$

${p_w^i}^T, {v_w^i}^T, {q_w^i}^T$ represents the translation, velocity, and quaternion rotation of the IMU w.r.t. world (or inertial frame). The dynamic models $\dot{p}_w^i, \dot{v}_w^i, \dot{q}_w^i$ propagate the state and do so at IMU rate.

D. Camera Pose Measurement by FPSP BIT-VO

From the BIT-VO algorithm [25], the vision sensor utilizes the FPSP. New Corner/Edge Features are detected via the FPSP, off-putting computational load by allowing some image and signal processing to be done on the chip before transferring to a PC host or other external device to be further processed for x_{BIT-VO}^T state estimation (unscaled).

BIT-VO has an FPSP-based feature-descriptor scheme titled BIT-descriptor (44-bit in length), which is used to make use of the local binary edge information and establish feature correspondences between frames. It is different from other binary descriptors because we cannot access the image intensity information. On a feature corner by FAST-16, create a 7×7 patch around it and account for rotation invariance. In the 7×7 patch, create 3 rings $r \in \{r_1, r_2, r_3\}$. The descriptor is then computed by taking the disjunction of $\{r_1, r_2, r_3\}$. These BIT-descriptors are then compared with each other. Note: though the BIT-descriptor is rotation invariant, it is not scale invariant.

The FPSP is utilized again in the frame-to-frame and map-to-frame matching processes. The low latency/high frame-rate produces small inter-frame motion between frames $F_1, F_2, \ldots, F_k$ that allows feature matching to be based upon simple pixel search-and-match around a small radius (3-5 pixels) of the said features.

The map refinement and keyframe selection of BIT-VO algorithm are similar to PTAM [16] and SVO [7]. Initialization is done by the 5-point algorithm with RANSAC.

${p_w^v}^T, {q_w^v}^T$ is the x_{BIT-VO}^T state estimation, unscaled. 3D map points and their corresponding projected points on the

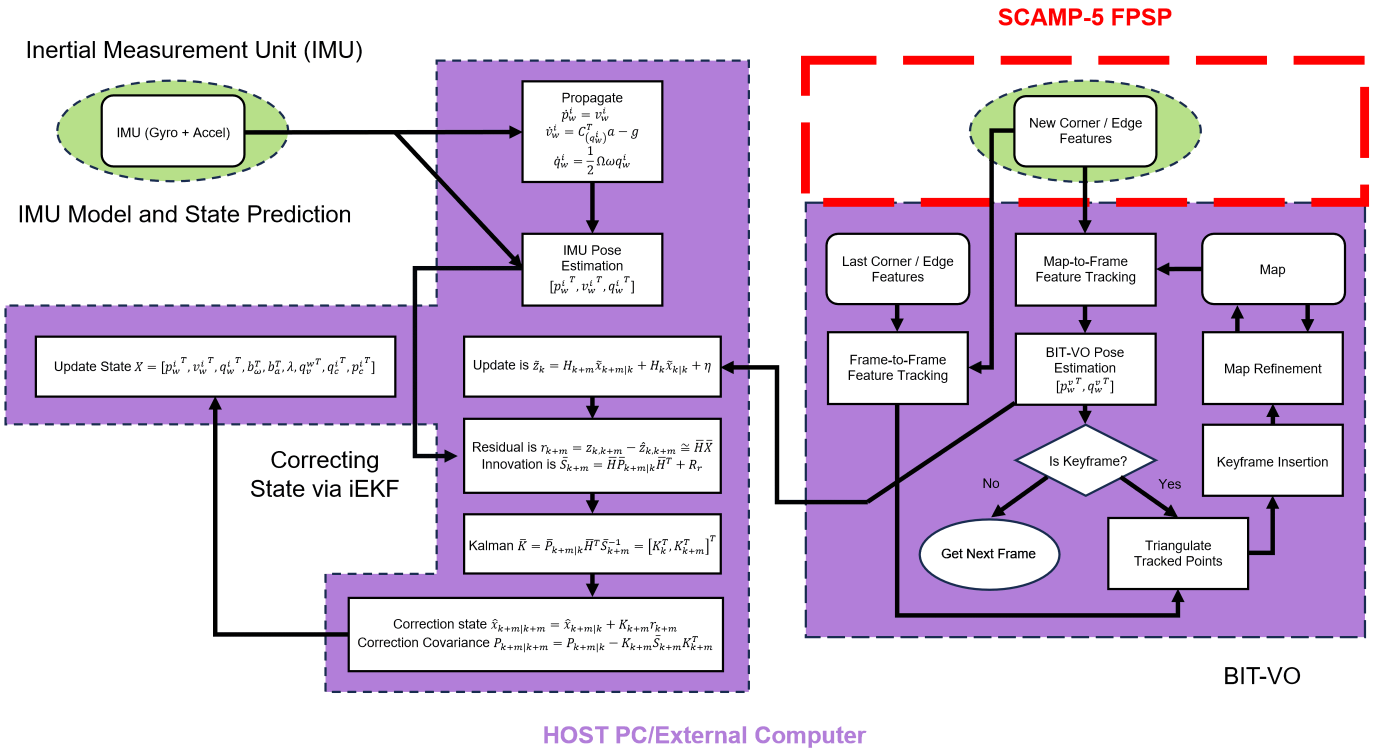

Fig. 3. Pipeline of BIT-VIO. Multi-Sensor Fusion (iEKF-MSF) is to the left. BIT-VO is to the right. From the BIT-VO algorithm[25], the vision sensor utilizes the FPSP, highlighted in red. New Corner/Edge Features are detected via the FPSP, off-putting computational load by allowing some image and signal processing to be done on the chip before transferring to a PC host or other external device to be further processed. i.e., some of the framework is alleviated on FPSP.

vision image plane are found. The pose is estimated by minimizing the reprojection error on the k possible poses:

$$[p_w^{v\,T}, q_w^{v\,T}] = \underset{[p_w^{v\,T}, q_w^{v\,T}]}{\operatorname{argmin}} \frac{1}{2} \sum_{i=0}^{k} \rho \left(\|u_i - \pi(T_{c,w} w p_i)\|^2 \right) \quad (7)$$

$\pi(T_{c,w} w p_i)$ is the projected 3D points on the vision image plane, $\rho(.)$ is the Huber loss function, reducing the effect of outlying data.

The x_{BIT-VO}^T (scaled with scale λ) part of the 10-element state is driven by the BIT-VO vision sensor measurements:

$$x_{BIT-VO}^T = [\Delta\lambda, \delta\Theta_i^{c\,T}, \Delta p_w^{v\,T}, \delta\Theta_w^{v\,T}] \quad (8)$$

We assume BIT-VO vision sensor measurement z_{BIT-VO} has Gaussian noise in position and rotational n_p, n_q. The measurement model is given by,

$$z_{BIT-VO} = \begin{bmatrix} p_v^c \\ q_v^c \end{bmatrix} \quad (9)$$

$$= \begin{bmatrix} C_{(q_w^v)}(p_w^i + C_{(q_w^i)}^T p_i^c)\lambda + p_w^v + n_{p_v} \\ q_i^c \otimes q_w^i \otimes q_w^{v-1} + n_{q_v} \end{bmatrix} \quad (10)$$

p_v^c, q_v^c propagate the state and do so at the BIT-VO vision sensor rate, which is the rate of 300 FPS.

E. Uncertainty Propagation of FPSP BIT-VO Pose

BIT-VO itself does not propagate an uncertainty or consist of a covariance for its vision 6-DOF pose. As mentioned, 3D map points and correspondences are found on an image plane, where the pose is optimized by minimizing the reprojection error for that instant.

Once the optimal pose is found from the set, we take the pose and, using Ceres, generate a 6×6 covariance block for the optimized parameters based on the optimal pose. It is often a mistake to count the 4-quaternion orientation (q_w, q_x, q_y, q_z) of a 6-DOF pose to have a covariance of 7×7, but it is 6×6 to account only for the 3-DOF in translation and 3-DOF orientation, a combination of the 4-quaternion. Our covariance is positive-definite.

F. Correcting State via iEKF

We may either assume BIT-VO vision sensor measurements as relative (as in depending between time-instants k and $k+m$) or absolute (ex. like IMU or GPS measurements).

If relative measurement:

1) Build full covariance matrix $\bar{P}_{k+m|k}$.
2) Update is $\tilde{z}_k = H_{k+m}\tilde{X}_{k+m|k} + H_k\tilde{X}_{k|k} + \eta$, H is the measurement Jacobian found via z_{BIT-VO}.
3) Compute residual $r_{k+m} = z_{k,k+m} - \hat{z}_{k,k+m} \cong \bar{H}\bar{X}$.
4) Compute innovation $\bar{S}_{k+m} = \bar{H}\bar{P}_{k+m|k}\bar{H}^T + R_r$, R_r is relative covariance $\bar{H} = [H_{k|k}, H_{k+m|k}]$.
5) Compute $\bar{K} = \bar{P}_{k+m|i}\bar{H}^T\bar{S}_{k+m}^{-1} = [K_k^T, K_{k+m}^T]^T$.
6) Correct state $\hat{x}_{k+m|k+m} = \hat{x}_{k+m|k} + K_{k+m}r_{k+m}$ and covariance $P_{k+m|k+m} = P_{k+m|k} - K_{k+m}\bar{S}_{k+m}K_{k+m}^T$.

If absolute, same format but use update just on k not $k+m$.

V. Experimental Results

This section presents the experimental results. First, the hardware setup is explained, then two different sets of experiments are demonstrated using 10 different trajectories.

A. Hardware and Calibration

The proposed BIT-VIO algorithm is tested on a moving robotic system at 300 FPS with an IMU at 400 Hz. For the calibration, $p_i^c = (0.006m, 0.04m, 0.07m)$ w.r.t. IMU-frame. We first calibrated the SCAMP-5 FPSP, then the IMU, then together.

To calibrate the IMU intrinsics, being the acceleration, gyroscopic, and bias noises, $n_a, n_\omega, b_a, b_\omega$, we conducted a calibration using the Allan variance method, which involved fine-tuning the estimation parameters for the IMU's acceleration and gyroscopic biases and noises. This approach allowed us to analyze the sensor readings across various observation intervals, enhancing the accuracy and reliability of our measurements. In the end, we minimize our IMU error. We achieve a accelerometer noise density and random walk: $[0.001865, 0.002]$ and gyroscope noise density and random walk $[0.001865, 4e06]$. The IMU gyroscope bias intrinsics error estimates are within the 3-σ error bound. To calibrate the camera intrinsics we first estimated the the focal length, camera center, and distortion coefficients $[fx, fy, cx, cy]$, and then optimized the intrinsics by optimization on a radtan lens. We achieve a Focal length: $[257.27, 258.00]$ and Principal point: $[127.44, 128.17]$ with a less than $< 1px$ error.

We have evaluated our robotic system against ground-truth data from a Vicon motion capture system, which consisted of 14 cameras calibrated and time-synced.

Given the inherent limitations of the SCAMP-5 FPSP, which precludes its compatibility with video datasets for direct comparison, we extended our evaluation to encompass fourteen real-world trajectory scenarios. These scenarios, designed to mimic practical applications, and are compiled of Circular, Straight, Curve, and Zigzag part-trajectories. We aligned and scale-adjusted our recorded trajectories along ground-truth as this is monocular SLAM. We measure the Absolute Pose Error (APE) [20] and report the Root Mean Squared Error (RMSE) [30] and the median to evaluate the accuracy of BIT-VO compared with our BIT-VIO algorithm after this with respect to ground-truth. We also do error mapping on BIT-VO and BIT-VIO and compare. The host computations were made on a PC host or external computer, with 13th Gen Intel Core i7-12700 CPU.

TABLE I
IMU at 400 Hz and BIT-VO at 300 FPS. BIT-VIO has lower APE.

APE with IMU at 400 Hz, BIT-VO at 300 FPS					
Traj.	Type	BIT-VO APE (m)	IMU APE (m)	BIT-VIO APE (m)	Length (m)
A	RMSE:	0.156247	0.156138	**0.156138**	3.6
	median:	0.138958	0.138315	**0.133202**	
B	RMSE:	0.144674	0.151913	**0.150545**	3.42
	median:	0.096717	0.106351	**0.09959**	
C	RMSE:	0.134617	0.121777	**0.12071**	2.6
	median:	0.119079	0.112582	**0.111856**	
D	RMSE:	0.05454	0.06597	**0.063498**	1.1
	median:	0.050929	0.053554	**0.049605**	
E	RMSE:	0.094479	0.08533	**0.086911**	1.7
	median:	0.07561	0.067952	**0.068756**	
F	RMSE:	0.175323	0.154923	**0.153335**	2.12
	median:	0.174444	0.142438	**0.140952**	
G	RMSE:	0.206866	0.196163	**0.195263**	2.4
	median:	0.15714	0.149788	**0.149103**	
H	RMSE:	**0.088185**	0.090189	0.090376	3.3
	median:	**0.070431**	0.07014	0.070212	
I	RMSE:	0.134361	0.13838	**0.134328**	2.01
	median:	0.116587	0.12805	**0.124618**	
J	RMSE:	0.057402	0.058275	**0.057461**	1.6
	median:	0.044689	0.044147	**0.042597**	
K	RMSE:	0.075214	0.092716	**0.092355**	2.48
	median:	0.067571	0.062949	**0.062812**	
L	RMSE:	0.10864	0.104469	**0.104366**	2.55
	median:	0.089689	0.088151	**0.08788**	
M	RMSE:	**0.094062**	0.108688	0.109156	2.9
	median:	**0.075766**	0.093151	0.09367	
N	RMSE:	0.101385	0.107546	**0.105227**	2.71
	median:	0.09016	0.089743	**0.087536**	

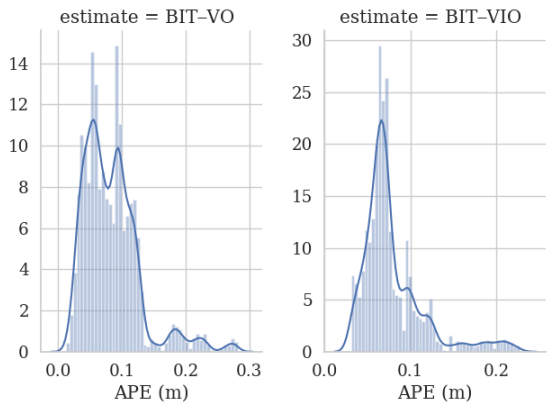

Fig. 4. As shown for Traj. E from Table I the concentration is shifted to the left and the median APE is more distinct after IMU propagation and BIT-VO filtering. BIT-VIO reduces the APE error of prior BIT-VO.

B. Accuracy and Robustness

As shown in Table I., when incorporating with an IMU, the pose generally enhances its estimation, yielding a more accurate trajectory, as showcased with the lower RMSE and Median and closer to ground-truth values of our BIT-VIO algorithm over prior BIT-VO's values. Traj. A-C are Circular and Curved, Traj. D is Straight and the rest are combinations of all with Zigzag. Certain like Traj. L are fast and hostile.

The main limitation of BIT-VO was the high-frequency noise in the predicted trajectory. If we focus on Traj. L from Table I, where length $L = 2.55m$ going from above, quickly moving around and Zigzagging and doing a Circular curve in a fast, hostile motion of short time length 14.5 s, we can see that our BIT-VIO algorithm effectively removes the high frequency noise visible in BIT-VO's trajectory as shown in Fig. 5 (top and middle). The Roll, Pitch and Yaw in these fast motions are tracked and our BIT-VIO algorithm outperforms BIT-VO by being closer to ground-truth in its angular motions, as well as its XYZ motion.

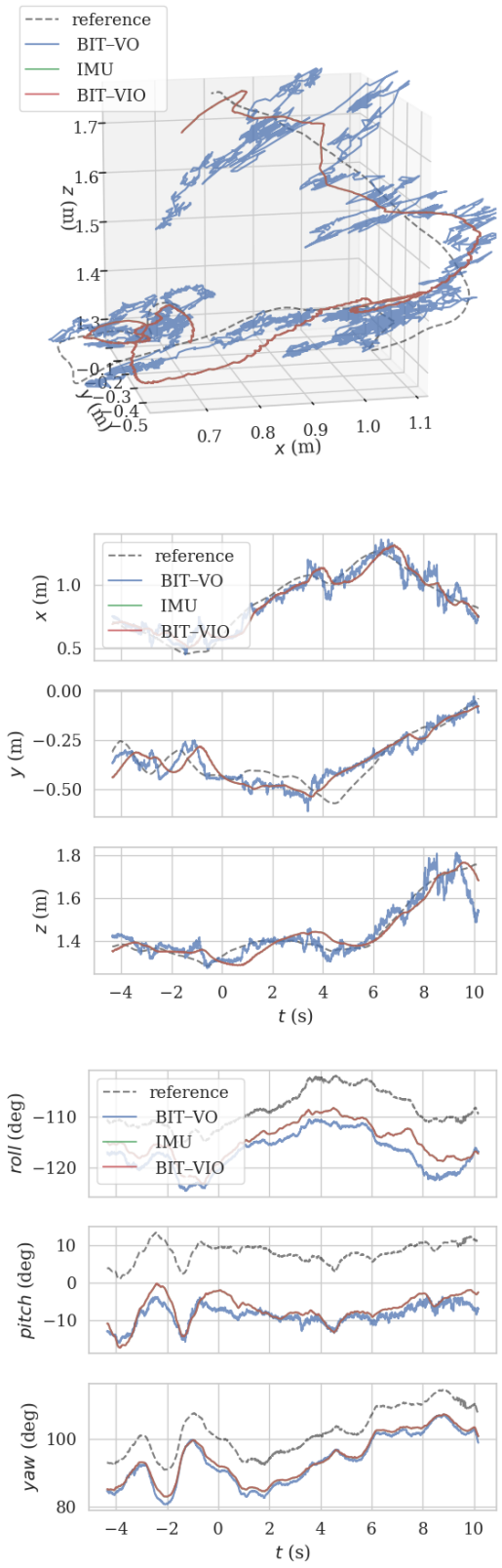

Fig. 5. Traj. L from Table I of $L = 2.55m$ ZigZag and Circular Curve. IMU (green) is propagated, and then the BIT-VO vision sensor (orange) acts as an update. Our BIT-VIO is closer to ground-truth and smoother than IMU and BIT-VO alone. BIT-VIO Roll, Pitch and Yaw in these fast motions are tracked better than BIT-VO.

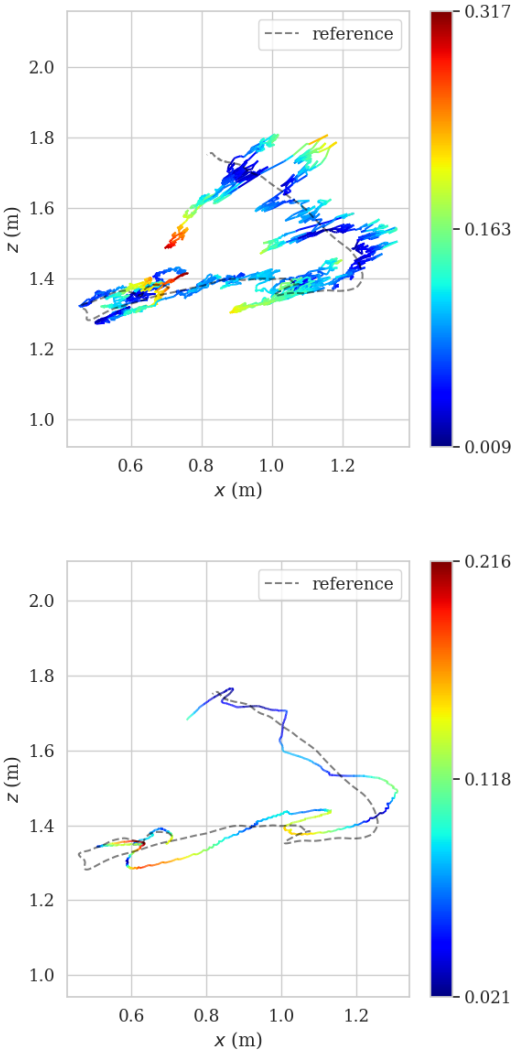

Fig. 6. When projecting the error on BIT-VO (top) and BIT-VIO (bottom) for Traj. L of Table I, we see that BIT-VO has high frequency noise with red tail-ends on its trajectory when compared to BIT-VIO's stabler closer to ground-truth trajectory with little regions of large APE error (in red).

C. Error Mapping Against BIT-VO vs. BIT-VIO

Doing error mapping on BIT-VO and BIT-VIO gives us greater details in the closeness of our BIT-VIO algorithm over BIT-VO. Further projecting the error map on Traj. L from Table I BIT-VO (as shown in Fig. 6 top) and on BIT-VIO (as shown in Fig. 6 top) projected on the xz-plane, we see that our BIT-VIO algorithm is stabler with IMU propagation. As seen in Fig. 6 top for BIT-VO there are regions of red at the tails of the trajectory with maximal error of $0.317m$ on the APE. In the same tail-end region for our BIT-VIO algorithm, the APE error is within the Median of a small bound around $0.118m$ with those regions being in lighter blue. We showcase the color as a measure of error of the APE. So, in the fast hostile motion of on Traj. L and other trajectories, we tend to see fewer regions of large APE error (in red) as is shown on prior BIT-VO. This is true for all our Traj. A-N from Table I.

VI. Conclusion

We have presented BIT-VIO, the first-ever 6-Degrees of Freedom (6-DOF) Visual Inertial Odometry (VIO) algorithm, which utilizes the advantages of the FPSP for vision-IMU-fused state estimation. Our BIT-VIO algorithm operates and corrects by loosely-coupled sensor-fusion iterated Extended Kalman Filter (iEKF) at 300 FPS with an IMU at 400 Hz . We showed that a high frame rate on the vision and a larger difference in rates between prediction and update help to minimize the APE. We evaluate BIT-VIO qualitatively against BIT-VO and demonstrate improvements in APE across many trajectories. Moreover, the high-frequency noise evident in BIT-VO is effectively filtered out, resulting in a smoother estimated trajectory. We plan to take the next steps toward a tightly-coupled VIO approach with the FPSP. Feature extraction would be integrated directly with IMU instead of two separate pose estimates, making the trajectories more robust and loop-closure better achievable. Further, this loosely-coupled work can also be improved by traversing larger scenes and workspaces. We also hope to better online and offline camera-IMU calibration.

VII. Acknowledgements

This research is supported by Natural Sciences and Engineering Research Council of Canada (NSERC). We would like to thank Piotr Dudek, Stephen J. Carey, and Jianing Chen at the University of Manchester for kindly providing access to SCAMP-5.

References

[1] Hector Castillo-Elizalde, Yanan Liu, Laurie Bose, and Walterio Mayol-Cuevas. Weighted node mapping and localisation on a pixel processor array. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6702–6708. IEEE, 2021.

[2] Jianing Chen, Stephen J Carey, and Piotr Dudek. Scamp5d vision system and development framework. In *Proceedings of the 12th International Conference on Distributed Smart Cameras*, pages 1–2, 2018.

[3] Jianing Chen, Yanan Liu, Stephen J Carey, and Piotr Dudek. Proximity estimation using vision features computed on sensor. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 2689–2695. IEEE, 2020.

[4] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[5] Tue-Cuong Dong-Si and Anastasios I Mourikis. Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In *2011 IEEE International Conference on Robotics and Automation*, pages 5655–5662. IEEE, 2011.

[6] Piotr Dudek. Scamp-3: A vision chip with simd current-mode analogue processor array. *Focal-plane sensor-processor chips*, pages 17–43, 2011.

[7] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016.

[8] Colin Greatwood, Laurie Bose, Thomas Richardson, Walterio Mayol-Cuevas, Jianing Chen, Stephen J Carey, and Piotr Dudek. Tracking control of a UAV with a parallel visual processor. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4248–4254. IEEE, 2017.

[9] Colin Greatwood, Laurie Bose, Thomas Richardson, Walterio Mayol-Cuevas, Jianing Chen, Stephen J Carey, and Piotr Dudek. Perspective correcting visual odometry for agile MAVs using a pixel processor array. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 987–994. IEEE, 2018.

[10] Colin Greatwood, Laurie Bose, Thomas Richardson, Walterio Mayol-Cuevas, Robert Clarke, Jianing Chen, Stephen J Carey, and Piotr Dudek. Towards drone racing with a pixel processor array. In *Proceeding of 11th International Micro Air Vehicle Competition and Conference, IMAV 2019*, pages 76–82, 2019.

[11] M. Grupp. evo: Python package for the evaluation of odometry and slam, 2017. https://github.com/MichaelGrupp/evo.

[12] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-IMU-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.

[13] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. An observability-constrained sliding window filter for SLAM. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 65–72. IEEE, 2011.

[14] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[15] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[16] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[17] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[18] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[19] Yanan Liu, Jianing Chen, Laurie Bose, Piotr Dudek, and Walterio Mayol-Cuevas. Direct servo control from in-sensor CNN inference with a pixel processor array. *arXiv preprint arXiv:2106.07561*, 2021.

[20] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4:333–349, 1997.

[21] Simon Lynen, Markus W Achtelik, Stephan Weiss, Margarita Chli, and Roland Siegwart. A robust and modular multi-sensor fusion approach applied to MAV navigation. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 3923–3929. IEEE, 2013.

[22] Alexander McConville, Laurie Bose, Robert Clarke, Walterio Mayol-Cuevas, Jianing Chen, Colin Greatwood, Stephen Carey, Piotr Dudek, and Tom Richardson. Visual odometry using pixel processor arrays for unmanned aerial systems in GPS denied environments. *Frontiers in Robotics and AI*, 7:126, 2020.

[23] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 3565–3572. IEEE, 2007.

[24] Anastasios I Mourikis and Stergios I Roumeliotis. A dual-layer estimator architecture for long-term localization. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[25] Riku Murai, Sajad Saeedi, and Paul HJ Kelly. BIT-VO: Visual odometry at 300 FPS using binary features from the focal plane. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8579–8586. IEEE, 2020.

[26] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[27] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006.

[28] Davide Scaramuzza, Zichao Zhang, Marcelo H Ang, Oussama Khatib, and Bruno Siciliano. Aerial robots, visual-inertial odometry of. 2020.

[29] Edward Stow, Abrar Ahsan, Yingying Li, Ali Babaei, Riku Murai, Sajad Saeedi, and Paul HJ Kelly. Compiling CNNs with cain: focal-plane processing for robot navigation. *Autonomous Robots*, 46(8):893–910, 2022.

[30] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.