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

Matthew Lisondra[†⋆], Junseo Kim[†⋆], Riku Murai[‡], Kourosh Zareinia[†], and Sajad Saeedi[†]

***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 the Single Instruction Multiple Data (SIMD) paradigm. In this paper, we present BIT-VIO, the first Visual Inertial Odometry (VIO) which utilises SCAMP-5. BIT-VIO is a loosely-coupled iterated Extended Kalman Filter (iEKF) which fuses together the visual odometry running fast at 300 FPS with predictions from 400 Hz IMU measurements to provide accurate and smooth trajectories.**

**Project Page: `https://sites.google.com/view/bit-vio/home`**

## I. Introduction

The reduced power consumption and latency associated with Visual Odometry (VO) and Visual Inertial Odometry (VIO) are becoming increasingly important as future mobile devices are anticipated to require rich and accurate spatial understanding capabilities [40]. Currently, conventional camera technology typically operates at 30-60 frames per second (FPS) and transfers a non-trivial amount of data from the sensor to the host device (e.g. a desktop PC). Such data transfer is not free – in terms of both power and latency –, and additionally, all these pixels must be then later processed on the host device.

As an alternative, Focal-Plane Sensor-Processor Arrays (FPSP)s, such as SCAMP-5, is a new technology that enables computation to occur on the imager's focal plane before transferring the data to a host-device [11]. By performing early-stage computer vision algorithms on the focal plane such as feature detections, FPSPs compress the image data down to the size of the features. By transferring only the detected features, redundant pixel information is not transferred or potentially even not digitized as FPSPs such as SCAMP-5 can perform analog computation.

In this work, we extend on BIT-VO [33], [34], a visual odometry algorithm which uses SCAMP-5, and 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. As shown in Fig. 1, BIT-VIO achieves a much smoother trajectory estimate when compared to BIT-VO, while retaining all the advantageous properties of BIT-VO such as low latency and high frame rate pose estimation. In short, the contributions of our work are:

- Efficient Visual Inertial Odometry operating and correcting by loosely-coupled sensor-fusion iterated Extended Kalman Filter (iEKF) at 300 FPS using predictions from IMU measurements obtained at 400 Hz.
- Uncertainty propagation for BIT-VO's pose as it is based on binary-edge-based descriptor extraction, 2D to 3D re-projection.
- Extensive real-world comparison against BIT-VO, with ground-truth obtained using a motion capture system.

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

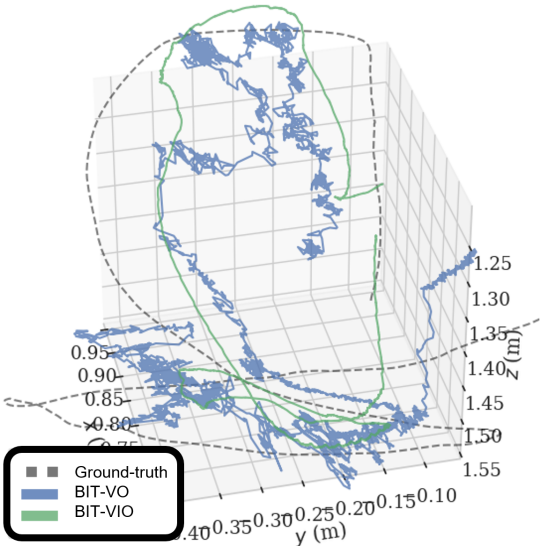


Fig. 1. Comparison of the proposed BIT-VIO algorithm and visual odometry (BIT-VO) overlaid on the reference ground-truth trajectory. BIT-VIO estimates are closer to the ground-truth trajectory compared to predictions from BIT-VO. Notice that BIT-VIO effectively removes the high-frequency noise visible in BIT-VO's trajectory. The plot was generated using evo [19].

## II. Background

In this section, we review SCAMP-5, an FPSP technology developed by the University of Manchester [11], [7], and it's application to robotics and visual odometry.

### A. SCAMP-5 FPSP

SCAMP-5 is a $256 \times 256$ processor array, performing parallel processing in a SIMD fashion on the focal plane

[†]Toronto Metropolitan University (TMU).
[‡]Imperial College London, Department of Computing.
*Both authors contributed equally to this research at TMU.

of the imaging sensor. The parallelism of the SCAMP-5 FPSP camera technology provides high computational capabilities and the on-sensor processing enables low-power operation. Each pixel has a processing element (PE) that contains 7 analog registers, 13 digital registers, and an ALU, enabling on-pixel logical and arithmetic operations. The analog registers can store a real-valued variable, up to around 8-bit resolution. The analog registers can do operations such as add, negate, split, and compare-against-0. 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.

Beyond SIMD parallelism, SCAMP-5's digital registers can be read out as events. Event readout only transfers coordinates of the digital registers set to 1 and is more efficient than transferring the whole image plane if only a sparse set and pixels have a register set to 1 [7]. Flooding is a feature of the SCAMP-5 that enables DREG propagation of 1s to help further accelerate image processing on its hardware. Through the SCAMP-5 camera technology's asynchronous propagation network, the flooding speed is nearly $62\times$ faster than accessing the neighbouring pixels on the SCAMP-5 via conventional message passing [4].

### B. Application of SCAMP-5 to Robotics

The SCAMP-5 has been utilized across many robotic systems. The odometry system by Greatwood et al. performs HDR edge-based odometry on the SCAMP-5 [17] and achieves lower power consumption than a conventional camera system. Additionally, the high frame rate nature of the system meant that it suffered less from motion blurs under agile motions. In [3], SCAMP-5 is used to track the 4 DoF camera motions using direct image alignment, and all computation is performed on the sensor itself. In [9], another algorithm is proposed, performing optical flow to estimate 4 DoF camera motion. In [29], SCAMP-5 is utilized for visual odometry on unmanned UAVs. Compared to using a conventional camera, they demonstrated that SCAMP-5-based systems have a clear practical advantage, for example, by computing HDR on the SCAMP-5, UAVs can transition from outdoor to indoor environments whilst successfully tracking, despite the changes in the lighting conditions.

The high frame-rate nature of SCAMP-5 also opens up the possibility for many interesting applications. For example, in-sensor CNN inference can perform hand gesture recognition for a rock, paper, scissors game at 8000 FPS, and can always make a robot play a winning hand [25]. While it is a simple setup, the game requires the system to have low end-to-end latency and is challenging to replicate using a conventional camera.

Fully utilising the different computational capabilities available on the SCAMP-5 device, [5] performs all-on-sensor mapping and localization. It performs visual route mapping and localization on the SCAMP-5 and runs at more than 300 FPS on various large-scale datasets.

SCAMP-5 has also been applied for controls, for example, using focal-plane processing, a ground target was detected and tracked to guide a small, agile quadrotor UAV [16]. In [18], they perform drone racing, using the SCAMP-5 to detect the gates. The gate size and location are the only data that is transferred, with minimal data transfer resulting in 500 FPS. In [8], different visual features such as corner points, blobs, and edges are extracted on the SCAMP-5 and fed into a recurrent neural network (RNN) for obstacle avoidance. In-sensor analog convolutions have been proposed in [48] and [10]. AnalogNavNet [42] utilized Cain [43] to implement a CNN that operates on the analog registers on SCAMP-5 for robotic navigation inside a corridor and racetrack environment.

### C. BIT-VO

Our method builds on the previous work BIT-VO [34], which performs 6 degrees of freedom (DoF) visual odometry at 300 FPS using a SCAMP-5 camera. In BIT-VO, the VO system is clearly separated into a frontend, which performs feature extraction, and a backend, which performs the matching of the features and the camera pose optimization. Following this separation, BIT-VO performs the frontend feature detection on the SCAMP-5 camera itself, where corners and binary edges are detected and transferred at 300 FPS utilising the SIMD processing capability of SCAMP-5 and the event readouts.

The detected features are then transferred to a host device, which performs the backend processing. Here, for each corner feature, a descriptor is formed using the binary edges. Using brute form matching, corners across frames are matched using the descriptors, similarly to ORB-SLAM [32]. Once the correspondences are established, the system is initialised using a 5-point algorithm [35] and after the initialization, the camera pose is optimized by minimising the map-to-frame reprojection error.

By operating at 300 FPS, BIT-VO is robust against rapid, agile camera motions. However, the estimated trajectory contains a high-frequency noise, which is due to the noisy feature detection on the focal plane. In this work, we aim to address this problem by incorporating IMU measurements.

### D. Visual Inertial Odometry

Visual inertial odometry (VIO) is the process of estimating camera pose by combining visual information from a camera and inertial measurements from IMUs. VIO provides accurate and robust pose estimates. The sensors complement each other and are used in many applications and products such as AR headsets. VIO can be categorised into loosely-coupled and tightly-coupled methods. In a loosely-coupled method, the visual and inertial measurements are independently processed to estimate the motion and then are fused together for correction. On the other hand, the tightly-coupled method directly estimates the motion from the visual and inertial measurements [41].

There are many different ways one can implement a VIO system. For example, one can use filtering [2], [30]

approaches, or using non-linear optimization, perform fixed-lag smoothing [23], [36] or even full smoothing [12]. For the full smoothing, solving the entire system at every observation quickly becomes infeasible, hence they rely on iSAM2 [20] for incremental factor graph optimization.

### E. Visual Inertial Odometry on Unconventional Cameras

While a conventional camera is typically used in VO and visual SLAM, other camera technologies such as event-based cameras are used in many state-of-the-art VIO algorithms [50], [37], [46], [31]. Event cameras provide low power usage and low latency benefits over conventional cameras and are also robust against illumination changes [24]. However, while event cameras compress visual information into a continuous stream of events, they are not user-programmable and cannot extract a specific feature such as FAST-corner on the sensor itself [6], [39]. Furthermore, the data volume transferred by an event camera is proportional to camera motion, and such a characteristic is not optimal; for instance, a robot has more data to process during rapid motion.

## III. Proposed Method

In this section, we first introduce the notations and then present an overview of the whole system.

### A. Notations

The following notation conventions are used in this work, adopted from [47], [45]:

- Units of a variable $A$ as $[A]$ (e.g. $[a_x] = m/s^2$).
- Skew-symmetric matrix of $A$ is $\lfloor A \rfloor$.
- $p_A^B$ represents the translation from frames $A \to B$.
- $q_A^B$ represents the Hamiltonian quaternion rotation $(q_{A\,w}^B, q_{A\,x}^B, q_{A\,y}^B, q_{A\,z}^B)$ from frames $A \to B$.
- $\hat{p}, \hat{q}$ 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 $\omega$.
- $\delta q = q \otimes \hat{q} \approx \left[\frac{1}{2}\delta\theta^T, 1\right]^T$ approx. for quaternion $\delta q$.
- $\vec{q} \otimes \vec{p} = (q_4 + q_1 i + q_2 j + q_3 k)(p_4 + p_1 i + p_2 j + p_3 k)$ where the quaternion multiplication is defined by operation $\otimes$.

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

### B. System Overview

Fig. 3 demonstrates an overview of the system. The visual odometry pipeline is shown on the right, and the inertial pipeline 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.

### C. IMU Model and State Prediction

We use iterated Extended Kalman Filter Multi-Sensor Fusion framework (iEKF-MSF) [27] and assume absolute IMU measurements have bias $b_\omega, b_a$ with Gaussian noise

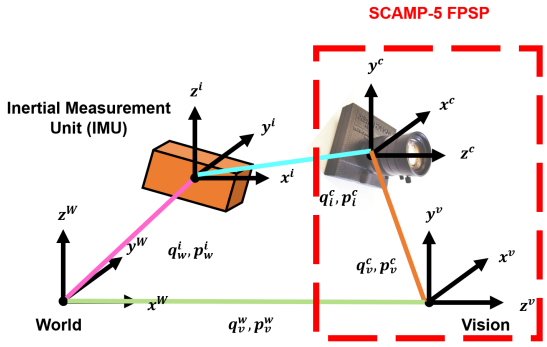


Fig. 2. Coordinate frame definition of the IMU and the SCAMP-5. In total, we define four coordinate frames. Notation $p_A^B$ and $q_A^B$ are used to represent transformation from $A$ to $B$.

$n_\omega, n_a$. IMU measures and outputs angular velocities $\omega$ and linear accelerations $a$ in the IMU-frame [47]:

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

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

In Eq. (1) and (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 $x_{BIT-VO}^T$. The $x_{IMU}^T$, which is a 16-element state, is formed by the IMU measurements and dynamic models, as follows [47]:

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

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

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

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

In Eq. (3)-(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 the rate of the IMU.

### D. Camera Pose Measurement by FPSP BIT-VO

In the BIT-VO [34], the front-end visual processing occurs on the SCAMP-5 FPSP. FAST corner and binary edge features are detected on the chip before it is transferred to a PC host or other external device. On the host device, the visual features are further processed to obtain camera pose estimation (unscaled as the system is monocular), $x_{BIT-VO}^T$, which is composed of ${p_w^v}^T, {q_w^v}^T$, i.e. position and orientation.

BIT-VO uses a BIT-descriptor (44-bit long feature), which is created from local binary edge information around the corner features and is used to establish feature correspondences between frames. It differs from other binary descriptors because BIT-VO does not have access to the image intensity information. To create a BIT-descriptor, around a corner feature, BIT-VO creates a $7 \times 7$ patch and rotates the patch to be rotationally invariant. In the $7 \times 7$ patch, BIT-VO creates 3 rings $r \in \{r_1, r_2, r_3\}$. To establish a correspondence,

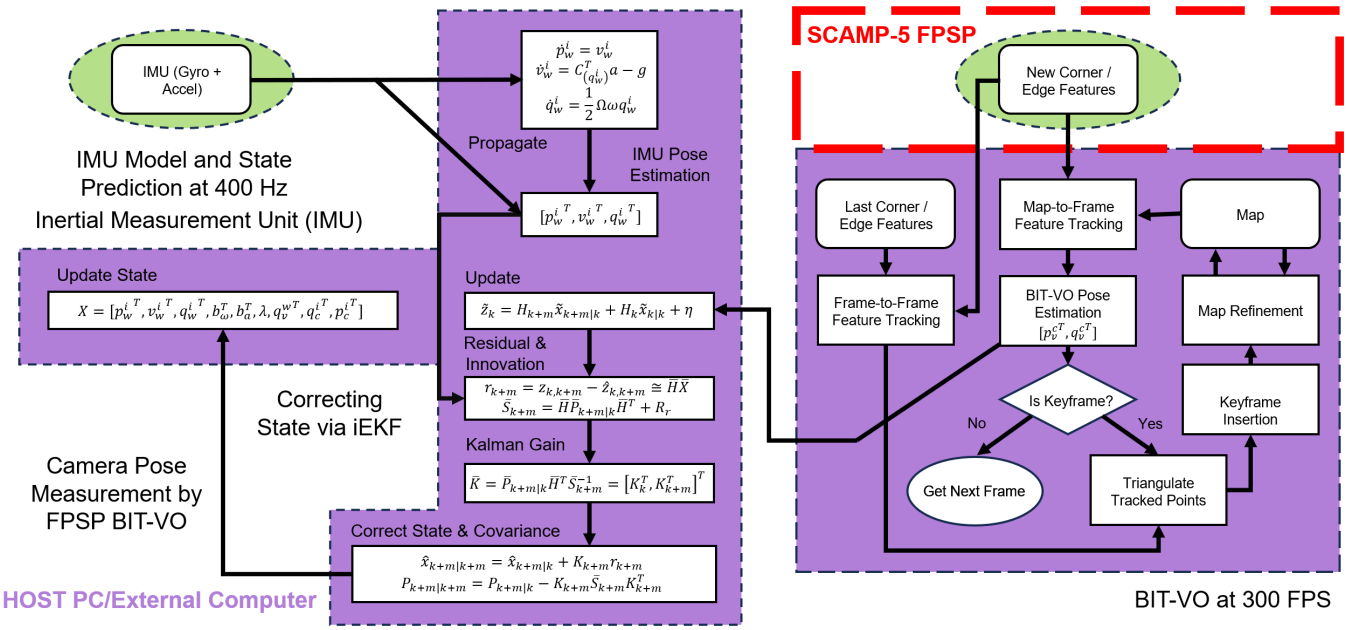


Fig. 3. Pipeline of BIT-VIO. The multi-sensor fusion is to the left. BIT-VO is to the right. From the BIT-VO algorithm [34], the vision sensor utilizes the SCAMP-5 FPSP, highlighted in red. New corner/edge features are detected via the FPSP, off-putting computational load by allowing some image and signal processing to be done on the chip before transferring to a PC host or other external device to be further processed.

hamming distance between two features (as the feature is binary) is taken to measure how similar two descriptors are. Though the BIT descriptor is rotation invariant, it is not scale invariant.

The high frame rate feature detection using SCAMP-5 FPSP simplifies the frame-to-frame and map-to-frame matching processes, as the inter-frame motion between frames is small. This allows the feature matching to be based upon simple brute-force search-and-match around a small radius (3-5 pixels) of the said features.

The map refinement and keyframe selection of the BIT-VO algorithm are similar to PTAM [22] and SVO [13]. Initialization is done by the 5-point algorithm with RANSAC.

Once the 3D map points and their corresponding $k$-projected points on the image plane are found, the pose is estimated by minimizing the reprojection error:

$$[p_v^{cT}, q_v^{cT}] = \underset{[p_v^{cT}, q_v^{cT}]}{\operatorname{argmin}} \frac{1}{2} \sum_{i=0}^{k} \rho \left( \|u_i - \pi(T_v^c \cdot {}_v p_i)\|^2 \right), \quad (7)$$

where $\pi(T_v^c \cdot {}_v p_i)$ is the function projecting 3D points on the vision image plane and $\rho(\cdot)$ is the Huber loss function, reducing the effect of outlying data.

The $x_{BIT-VO}^T$ (scaled with scale $\lambda$) part of the 10-element state is defined as:

$$x_{BIT-VO}^T = [\Delta\lambda, \delta\theta_i^{cT}, \Delta p_w^{v\,T}, \delta\theta_w^{v\,T}] \quad (8)$$

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 [45][47].

### E. Uncertainty Propagation of FPSP BIT-VO Pose

BIT-VO itself does not propagate an uncertainty or consist of covariance for its vision 6-DOF pose. 3D map points and correspondences are computed on the PC, where the pose is optimized by minimizing the reprojection error. Once the optimal pose $[p_v^{cT}, q_v^{cT}]$ is found from the set, we take the pose and, using Ceres [1], generate a $6 \times 6$ covariance block for the optimized parameters based on the optimal pose. It starts with forming the Jacobian of the residual blocks with respect to $[p_v^{cT}, q_v^{cT}]$, then the Hessian $H$ is approximated as $J^T J$, lastly with the covariance being computed as the inverse of the approximated Hessian $\Sigma = H^{-1} = (J^T J)^{-1}$. Note, here the covariance matrix is a $6 \times 6$ positive definite matrix, correctly matching the system's DOF rather than the state's dimensionality (which is 7 as we have 3 parameters for the translation and 4 parameters for the quaternion).

### 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 (e.g. IMU or GPS measurements). If it is a relative measurement, see Alg. 1. Otherwise, if absolute, the algorithm remains the same, but the updates must occur on $k$ not $k+m$.

**Algorithm 1** Correcting State via iEKF-MSF Update Process

1: Build full covariance matrix $\bar{P}_{k+m|k}$.
2: Update is $\tilde{z}_k = H_{k+m}\tilde{X}_{k+m|k} + H_k\tilde{X}_{k|k} + \eta$, where $H$ 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 $\bar{S}_{k+m} = \bar{H}\bar{P}_{k+m|k}\bar{H}^T + R_r$, where $R_r$ is covariance of measurement and $\bar{H} = [H_{k|k}, H_{k+m|k}]$.
5: Compute $\bar{K} = \bar{P}_{k+m|k}\bar{H}^T\bar{S}^{-1}_{k+m} = [K_k^T, K_{k+m}^T]^T$.
6: Correct state $\hat{x}_{k+m|k+m} = \hat{x}_{k+m|k} + K_{k+m}r_{k+m}$ and covariance $P_{k+m|k+m} = P_{k+m|k} - K_{k+m}\bar{S}_{k+m}K^T_{k+m}$.

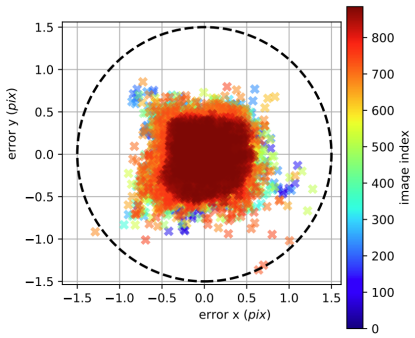


Fig. 4. FPSP SCAMP-5 reprojection error on a $256 \times 256$ focal-plane imaging output with a less than one-pixel error (within the dashed black).

## IV. Experimental Results

This section presents the experimental results. First, we explain the experimental setup and the calibration of the system, then we discuss the experimental results. We evaluate the system on eight different trajectories (labelled A-H).

### A. Experimental Setup

The proposed BIT-VIO algorithm is tested at 300 FPS, running BIT-VO on a SCAMP-5 FPSP device. Additionally, we attach an Intel D435i RealSense Camera to provide IMU measurements at 400 Hz.

Evaluations are done against ground-truth data from a Vicon motion capture system, which consisted of 14 cameras calibrated and time-synced. As both BIT-VO and BIT-VIO assume a fast frame rate, hence small inter-frame motion, we cannot evaluate our method using a standard benchmark dataset for direct comparison with other methods. Hence, BIT-VIO is evaluated on eight real-world trajectories against BIT-VO. These trajectories are designed to mimic practical applications and are a compilation of circular, straight, curved, and zigzag trajectories. The recorded trajectories are aligned and scaled to the ground-truth trajectory as our setup is monocular. We measure the Absolute Trajectory Error (ATE) [26] and report the Root Mean Squared Error (RMSE) [44] together with the median to evaluate the accuracy against the ground-truth. BIT-VO and BIT-VIO use a host device to perform the visual odometry backend, and for the host device, we use an external laptop with 13th Gen Intel Core i7-12700 CPU.

TABLE I
ATE comparison of BIT-VIO and BIT-VO. The lower ATE is emphasized in bold.

| Traj. | Type | BIT-VO ATE (m) | BIT-VIO ATE (m) | Length (m) |
|---|---|---|---|---|
| A | RMSE: | 0.215732 | **0.167631** | 2.1 |
| | median: | 0.170214 | **0.152106** | |
| B | RMSE: | 0.134617 | **0.12071** | 2.6 |
| | median: | 0.119079 | **0.111856** | |
| C | RMSE: | 0.094479 | **0.086911** | 1.7 |
| | median: | 0.07561 | **0.068756** | |
| D | RMSE: | 0.175323 | **0.153335** | 2.12 |
| | median: | 0.174444 | **0.140952** | |
| E | RMSE: | 0.206866 | **0.195263** | 2.4 |
| | median: | 0.15714 | **0.149103** | |
| F | RMSE: | **0.134361** | 0.134328 | 2.01 |
| | median: | **0.116587** | 0.124618 | |
| G | RMSE: | 0.132624 | **0.10535** | 1.8 |
| | median: | 0.125924 | **0.095664** | |
| H | RMSE: | 0.10864 | **0.104366** | 2.55 |
| | median: | 0.089689 | **0.08788** | |

### B. Sensor Calibration

We use Kalibr [15] to perform the calibration between SCAMP-5 and the IMU. For the extrinsic calibration, we obtain $p_i^c = (0.006m, 0.04m, 0.07m)$ with respect to the IMU-frame. To calibrate the IMU intrinsics: acceleration, gyroscopic, and bias noises, $n_a, n_\omega, b_a, b_\omega$, we conducted a calibration using the Allan variance method [14], [15], [38], [49]. We achieve an accelerometer noise density and random walk: $[0.001865m/s^2/\sqrt{Hz}, 0.002m/s^3/\sqrt{Hz}]$ and gyroscope noise density and random walk $[0.001865m/s^2/\sqrt{Hz}, 4 \times 10^{-6}m/s^3/\sqrt{Hz}]$. The IMU gyroscope bias intrinsics error estimates are within the 3-$\sigma$ error bound. To calibrate the camera intrinsics we first estimated the focal length, camera center $[fx, fy, cx, cy]$, and distortion coefficients, and then optimized the intrinsics by optimization on a radtan lens [21], [28]. We achieve a focal length: $[257.27, 258.00]$ pixels and principal point: $[127.44, 128.17]$ pixels, with a less than one pixel error as shown in Fig. 4.

### C. Accuracy and Robustness

As shown in Table I, when incorporating an IMU, the state generally enhances its estimation with a more accurate trajectory, showcasing lower RMSE and median closer to the ground-truth values. Traj. A and B are circular and curved, Traj. C is straight, and the rest are combinations of all with zigzag. The case of Traj. G in Fig. 5, shows that IMU-alone accumulates error and drifts away from ground-truth data, as shown in the large translational, rotational RMSE. In fact, it has the largest RMSE compared to BIT-VO and BIT-VIO. The BIT-VIO algorithm fixes this IMU error drift, using the BIT-VO update to align it closer to ground-truth data, hence why its RMSE is the lowest of the three. This is true in both the translational and rotational context, where we can

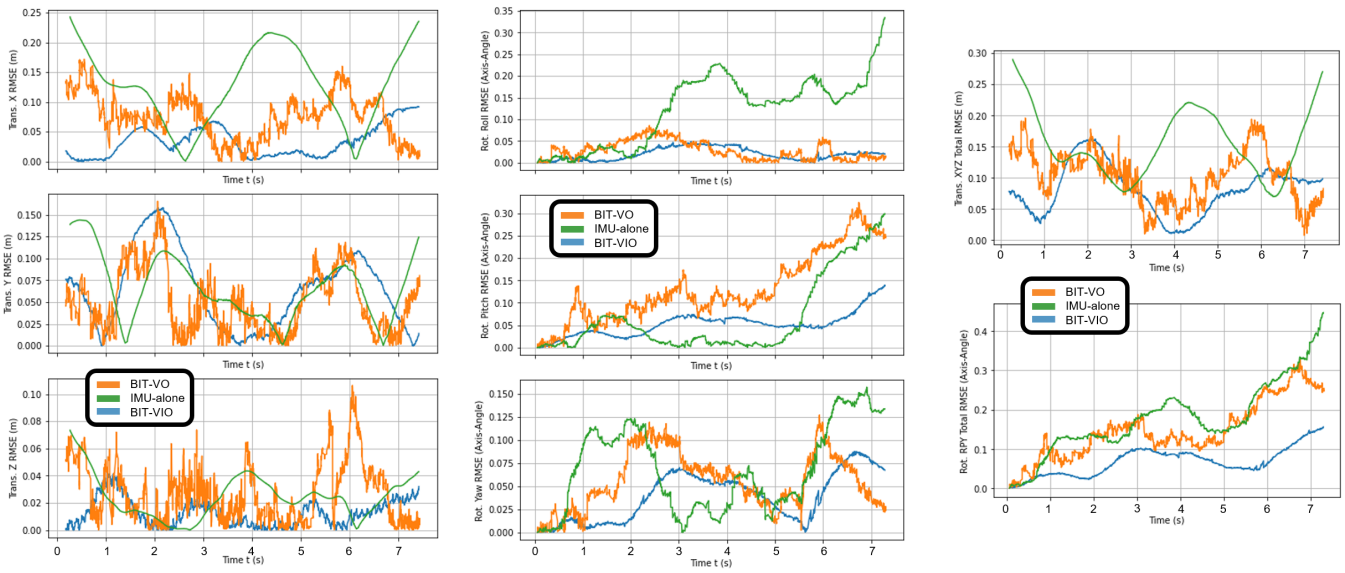


Fig. 5. Plots of the estimated translational RMSE (left) and rotational RMSE (middle) for Traj. G from Table I. To the very right is the total translational RSME (top) and total rotational RMSE (bottom). For both translation and rotation, BIT-VIO is much closer and smoother to ground-truth data than IMU-alone and BIT-VO. The drift of the IMU-alone is very evident, as well as the high-frequency noise of BIT-VO.

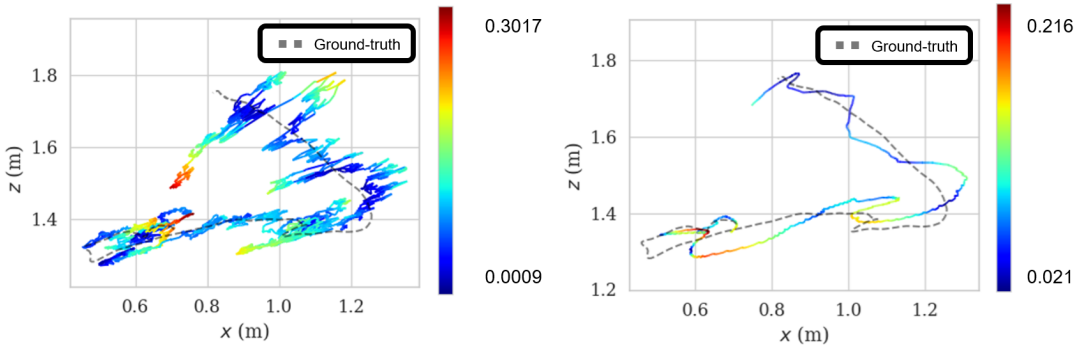


Fig. 6. Projection of Traj. H from Table I onto xz-plane for BIT-VO (left) and BIT-VIO (right). We observe that BIT-VO suffers from high-frequency noise when compared to BIT-VIO's estimates, although the overall RMSE ATE are similar.

see that in the left and middle plots of Fig. 5, BIT-VIO RMSE generally resides much more below both IMU and BIT-VO. To add, the BIT-VIO algorithm deals well with fast, hostile motions, covering the main limitation of the prior work BIT-VO with the high-frequency noise on its predicted trajectories. Through all plots in Fig. 5, BIT-VO maintains its noise. BIT-VIO not only maintains itself closer to ground-truth trajectory but also does well to track smoothly with less noise, especially in more violent, quick, hostile motions.

In Fig. 6, we can see projecting the trajectory error for both BIT-VO and BIT-VIO on ground-truth Traj. H, onto an xz-plane, qualitatively gives us further insight into the magnitude of the high-frequency error present in BIT-VO and how much is removed by BIT-VIO. Not only the trajectory estimated by BIT-VIO is smoother, but it is also closer to the ground-truth trajectory.

## V. 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 SCAMP-5 FPSP for vision-IMU-fused state estimation. BIT-VIO operates and corrects by loosely-coupled sensor-fusion iterated Extended Kalman Filter (iEKF) at 300 FPS with an IMU at 400 Hz. We evaluate BIT-VIO against BIT-VO and demonstrate improvements in ATE across many trajectories. Moreover, the high-frequency noise evident in BIT-VO is effectively filtered out, resulting in a smoother estimated trajectory.

In the future, we plan to take the next steps toward a tightly-coupled VIO approach using the SCAMP-5 FPSP.

## VI. Acknowledgements

We would like to thank Ali Babaei and Abrar Ahsan for their early help with the SCAMP-5. 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] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 2023. https://github.com/ceres-solver/ceres-solver.

[2] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 298–304. IEEE, 2015.

[3] Laurie Bose, Jianing Chen, Stephen J. Carey, Piotr Dudek, and Walterio Mayol-Cuevas. Visual Odometry for Pixel Processor Arrays. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4614–4622. IEEE, October 2017.

[4] Stephen J Carey, Alexey Lopich, David RW Barr, Bin Wang, and Piotr Dudek. A 100,000 FPS vision sensor with embedded 535gops/w 256× 256 SIMD processor array. In *2013 symposium on VLSI circuits*, pages C182–C183. IEEE, 2013.

[5] 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.

[6] Jianing Chen, Stephen Carey, and Piotr Dudek. Feature extraction using a portable vision system. 2017.

[7] 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.

[8] 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.

[9] Thomas Debrunner, Sajad Saeedi, Laurie Bose, Andrew J. Davison, and Paul H. J. Kelly. Camera tracking on focal-plane sensor-processor arrays. In *High Performance and Embedded Architecture and Compilation (HiPEAC), Workshop on Programmability and Architectures for Heterogeneous Multicores (MULTIPROG)*, 2019.

[10] Thomas Debrunner, Sajad Saeedi, and Paul H J Kelly. AUKE: Automatic Kernel Code Generation for an Analogue SIMD Focal-Plane Sensor-Processor Array. *ACM Transactions on Architecture and Code Optimization*, 15:1–26, 01 2019.

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

[12] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2016.

[13] 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.

[14] Paul Furgale, T D Barfoot, and G Sibley. Continuous-time batch estimation using temporal basis functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2088–2095, St. Paul, MN, 2012.

[15] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] J. Kannala and S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006.

[22] 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.

[23] 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.

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

[25] 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.

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

[27] 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.

[28] J. Maye, P. Furgale, and R. Siegwart. Self-supervised calibration for robotic systems. In *Proc. of the IEEE Intelligent Vehicles Symposium (IVS)*, 2013.

[29] 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.

[30] 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.

[31] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6):1425–1440, 2018.

[32] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[33] Riku Murai, Sajad Saeedi, and Paul H. J. Kelly. High-frame rate homography and visual odometry by tracking binary features from the focal plane. *Autonomous Robots*, Jul 2023.

[34] 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.

[35] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[36] 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.

[37] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. 2017.

[38] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016.

[39] 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.

[40] Sajad Saeedi, Bruno Bodin, Harry Wagstaff, Andy Nisbet, Luigi Nardi, John Mawer, Nicolas Melot, Oscar Palomar, Emanuele Vespa, Tom Spink, Cosmin Gorgovan, Andrew Webb, James Clarkson, Erik To-

musk, Thomas Debrunner, Kuba Kaszyk, Pablo Gonzalez-De-Aledo, Andrey Rodchenko, Graham Riley, Christos Kotselidis, Björn Franke, Michael F.P. O'Boyle, Andrew J. Davison, Paul H. J. Kelly, Mikel Luján, and Steve Furber. Navigating the landscape for real-time localization and mapping for robotics and virtual and augmented reality. *Proceedings of the IEEE*, 106(11):2020–2039, 2018.

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

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

[43] Edward Stow, Riku Murai, Sajad Saeedi, and Paul H. J. Kelly. Cain: Automatic code generation for simultaneous convolutional kernels on focal-plane sensor-processors. In *International Workshop on Languages and Compilers for Parallel Computing*, pages 181–197. Springer, 2020.

[44] 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.

[45] Nikolas Trawny and Stergios I Roumeliotis. Indirect Kalman filter for 3D attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2:2005, 2005.

[46] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[47] Stephan Weiss and Roland Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *2011 IEEE international conference on robotics and automation*, pages 4531–4537. IEEE, 2011.

[48] Matthew Z. Wong, Benoît Guillard, Riku Murai, Sajad Saeedi, and Paul H. J. Kelly. AnalogNet: Convolutional neural network inference on analog focal plane sensor processors. *ArXiv*, abs/2006.01765, 2020.

[49] Li Zhang, Zhen Liu, and C. Honghui Xia. Clock synchronization algorithms for network measurements. In *Proceedings of the IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002.

[50] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2017.