ORBCam: Toward Ultra-Low-Power Quantized VIO via In-Sensor ORB Feature Processing

Yiwen Liang[1,†], Yuxiang Peng[2,†], Guoquan Huang[2], Weidong Cao[1], Chuchu Chen[1]

***Abstract*— Micro-robots and always-on wearable systems require continuous ego-motion estimation to operate without external infrastructure or GPS. Visual-inertial odometry (VIO) has emerged as a practical solution, but enabling high-rate pose estimation under *milliwatt-level* power consumption remains a fundamental challenge. Conventional VIO pipelines fully acquire and transfer images off-chip, even though the estimator ultimately requires only feature tracks. Such extensive image acquisition and associated data movement between sensors and processors result in the primary power bottleneck in VIO systems. To address it, this work presents ORBCam, a cross-layer sensor-estimator co-design framework that directly generates motion-required feature measurements within the sensor subsystem. Specifically, FAST detection and rBRIEF descriptor construction are performed in-sensor, while matching, state-free outlier rejection, and measurement quantization are executed in a lightweight digital logic unit. ORBCam is evaluated using system-level simulations at 752×480 image resolution and 100 FPS, and is compared against a conventional image sensor with full-frame acquisition and transmission. Our evaluations show that ORBCam achieves up to a 13.3× energy-efficiency improvement while maintaining comparable odometry accuracy. These results highlight the great potential of sensor-estimator co-design for ultra-low-power VIO systems.**

I. Introduction and Related Work

Continuous ego-motion tracking is a key capability for robots, AR/VR wearables, and always-on vision systems. Visual-inertial odometry (VIO) that integrates camera observations with high-rate IMU measurements has become one of the most practical and widely adopted solutions [1]. Energy efficiency is key to deploying VIO on these platforms, which have strict size, weight, and power (SWaP) constraints yet require motion estimation at tens to hundreds of frames per second. High power consumption not only shortens battery life but also increases heat dissipation, limiting long-time operation on compact devices. Extensive research efforts, together with advances in hardware acceleration, have significantly reduced the arithmetic cost of visual-inertial estimation. However, in many of these resource-constrained systems, the dominant energy consumption no longer comes from computation, but from frequent image acquisition and the movement of large volumes of image data from camera sensors to processors.

A conventional VIO pipeline is illustrated in Fig. 1(a). Image frames captured by the camera sensor are first transferred to the host processor, where visual features (e.g., ORB keypoints and descriptors) are extracted and matched across frames to establish correspondences. These visual measurements are then fused with inertial measurements

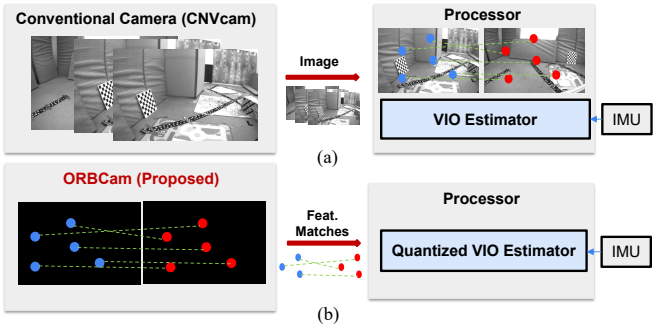


Fig. 1: (a) Conventional VIO pipeline, where the camera acquires full-frame images and transmits the massive raw pixels to the processor, which then performs original VIO estimation. (b) ORBCam pipeline, where the sensor directly produces quantized measurements without forming and transmitting images. Feature extraction, matching, motion validation, and quantization are performed within the sensor, and only compact quantized measurements are transmitted to the host estimator.

from the IMU to estimate the system's ego-motion. Specifically, a conventional CMOS image sensor (CIS) captures and digitizes the entire pixel array through readout circuits and analog-to-digital converters (ADCs), followed by image signal processing (ISP) and high-speed off-chip transmission (e.g., MIPI CSI-2). Consequently, all subsequent processing depends on fully digitized image data, incurring substantial ADC activity, memory traffic, and off-chip bandwidth. This conventional paradigm leads to significant energy overhead for data acquisition and transmission. Prior studies [2]–[4] have shown that *ADCs and output buffers account for up to 69% of total sensor power, while off-chip communication can consume up to 100 pJ/Byte.* Critically, while CIS generates massive high-quality images for motion tracking with high costs, state-of-the-art (SOTA) ego-motion estimation methods rely on establishing correspondences between sparse visual features across frames to form constraints. This reveals a fundamental discrepancy between conventional human-centric vision (for human illustrations) and machine-centric vision (for feature extraction).

Existing research has made substantial progress in optimizing individual operations of the VIO processing pipeline. To reduce the energy and computational load of the host-side estimator, dedicated accelerators have been developed to lower arithmetic complexity and memory traffic in visual-inertial pipelines [5], [6]. Quantized Kalman filtering and MAP-based formulations further improve estimator efficiency by explicitly modeling quantization in the measurement likelihood, enabling statistically consistent updates with low-bit observations [7]. To reduce transmission bandwidth, image compression and descriptor-level quantization have

†Equal contribution.
[1]The George Washington University, Washington, DC 20052, USA. Email: {yiwen.liang, weidong.cao, chuchu.chen}@gwu.edu.
[2]University of Delaware, Newark, DE 19716, USA. Email: {yxpeng, ghuang}@udel.edu.

been proposed to decrease storage and communication overhead [8]–[18]. Near- and on-sensor architectures further push portions of feature extraction closer to the pixel array to alleviate host-side computation and data movement [19]–[23]. However, **these approaches share a fundamental bottleneck**: *full-frame image acquisition remains*. Even when computation is pushed closer to the sensor, pixel-level ADC conversion and communication are still performed in full. As a result, the dominant energy costs associated with data acquisition and transmission remain largely unchanged. What remains unaddressed is the sensing abstraction itself. *Can task-relevant features be generated directly within the sensor, eliminating the image as an intermediate representation?* This motivates us to a paradigm shift in sensor design.

In standard VIO, motion constraints are built by establishing correspondences between visual features across frames. The front-ends typically obtain such correspondences in two ways: (i) direct feature tracking (e.g., KLT/Lucas-Kanade optical flow), which iteratively aligns image patches using pixel intensities and gradients, often over multi-scale pyramids; and (ii) keypoint detection and descriptor matching, which form correspondences by comparing compact descriptors. While direct tracking can be effective, it requires repeated access to full-resolution image data during iterative optimization. In this work, we therefore adopt an ORB-based formulation [24]: ORB is largely comparison-based and non-iterative, producing binary descriptors that are well suited for low-precision and in-sensor implementation.

We present ORBCam, an algorithm-hardware co-design framework that restructures the VIO pipeline from the ground up to address sensing bottlenecks. Instead of digitizing full image frames, ORBCam performs feature detection, description, matching, and motion validation directly within a mixed-signal sensing architecture, exporting only compact and quantized keypoint correspondences. These measurements are then processed by a quantization-aware VIO estimator (Fig. 1(b)). With this co-design, ORBCam reduces sensor energy and off-chip bandwidth while preserving the information required for accurate motion estimation. In summary, this work makes the following key contributions:

- We introduce ORBCam, a mixed-signal machine-centric image sensor that directly generates quantized feature measurements, without full-frame pixel digitization and high-bandwidth image transmission.
- We co-design the visual front-end with ORBCam by partitioning the VIO pipeline across sensing and lightweight digital modules. ORBCam performs in-sensor FAST detection and rBRIEF descriptor construction, followed by matching, state-free outlier rejection, and measurement quantization to produce compact estimator-ready motion measurements.
- Through circuit-level modeling and system-level simulation, ORBCam demonstrates up to $13.3\times$ sensing energy reduction, $2.2\times$ speedup, and $927.6\times$ communication bandwidth reduction compared with the conventional image sensor, while preserving VIO accuracy.

II. Conventional Image-Centric VIO Pipeline

A typical VIO pipeline consists of three stages as shown in Fig. 1(a). The key idea of VIO is to fuse high-rate IMU measurements with visual feature observations to estimate the motion. Following the standard MSCKF formulation [25], at time t_k, the system state $\mathbf{x}_k$ consists of the current navigation state $\mathbf{x}_{I_k}$, pose clones $\mathbf{x}_C$, and features $\mathbf{x}_f$:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_{I_k}^\top & \mathbf{x}_C^\top & \mathbf{x}_f^\top\end{bmatrix}^\top \tag{1}$$

$$\mathbf{x}_{I_k} = \begin{bmatrix}{}^{I_k}_{G}\bar{q}^\top & {}^{G}\mathbf{p}_{I_k}^\top & {}^{G}\mathbf{v}_{I_k}^\top & \mathbf{b}_g^\top & \mathbf{b}_a^\top\end{bmatrix}^\top \tag{2}$$

$$\mathbf{x}_C = \begin{bmatrix}\mathbf{x}_{T_k}^\top \dots \mathbf{x}_{T_{k-c}}^\top\end{bmatrix}^\top, \quad \mathbf{x}_f = \begin{bmatrix}\mathbf{f}_1^\top \dots \mathbf{f}_g^\top\end{bmatrix}^\top \tag{3}$$

where ${}^{I}_{G}\bar{q}$ is the unit quaternion corresponding to the rotation matrix ${}^{I}_{G}\mathbf{R}$ that represents the rotation from the global frame $\{G\}$ to the IMU frame $\{I\}$; ${}^{G}\mathbf{p}_I$, ${}^{G}\mathbf{v}_I$ are the IMU position, velocity; $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases; $\mathbf{x}_{T_k}$ is the cloned pose at time t_k. $\mathbf{f}_i$ denotes the 3D position of the i-th landmark.

A standard 6-axis IMU provides high-rate local linear acceleration and angular velocity measurements $\mathbf{a}_k$ and $\boldsymbol{\omega}_k$. We propagate the state over time based on the IMU kinematics.

$$\mathbf{x}_{I_{k+1}} = \mathbf{f}_I\left(\mathbf{x}_{I_k}, \mathbf{a}_k, \boldsymbol{\omega}_k, \mathbf{n}_I\right) \tag{4}$$

where $\mathbf{n}_I$ consists the noises for the measurement and model.

A camera front-end on the host first extracts and tracks features across frames and performs outlier rejection to establish reliable feature correspondences. The corresponding bearing measurement function is given by:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k =: \boldsymbol{\Lambda}({}^{C_k}\mathbf{f}) + \mathbf{n}_k \tag{5}$$

$${}^{C_k}\mathbf{f} = [x\ y\ z]^\top = {}^{C}_{I}\mathbf{R}\,{}^{I_k}_{G}\mathbf{R}\left({}^{G}\mathbf{f} - {}^{G}\mathbf{p}_{I_k}\right) + {}^{C}\mathbf{p}_I \tag{6}$$

where $\mathbf{z}_k = [u, v]^\top$ is the raw uv pixel coordinate; $\mathbf{n}_k$ is the zero-mean white Gaussian raw pixel noise. $({}^{C}_{I}\mathbf{R}, {}^{C}\mathbf{p}_I)$ denotes the fixed IMU-to-camera extrinsic calibration. The projection $\boldsymbol{\Lambda}(\cdot)$ is the calibrated camera projection from a 3D point in the camera frame to pixel coordinates. The detailed camera model follows [26], [27]. After linearization, the resulting residuals are incorporated into the estimator to update the state estimate.

As mentioned before, although the estimator ultimately operates on sparse feature measurements, conventional systems still generate full image frames and transmit them off-chip, incurring substantial energy cost in both image formation and data movement. Note that each IMU measurement contains only a few values (e.g., 3-axis acceleration and angular velocity), making its data rate negligible compared to the image stream. In parallel, existing work (e.g., QVIO [20], [21]) suggests that accurate state estimation can be performed using reduced-bit visual measurements by explicitly modeling quantization in the measurement likelihood. However, without sensor–estimator co-design, these approaches still rely on conventional full-frame image acquisition. This motivates the design of a new camera sensor architecture for VIO that (1) avoids full-frame image generation and (2) reduces off-chip data transmission. By reducing data acquisition at the source, such cameras would benefit not only the sensors themselves but also the overall processing pipeline.

III. ORBCam: A Machine-Centric Camera

To address the above limitations, we propose ORBCam, an ORB-based machine-centric camera designed for ultra-low-power VIO, as illustrated in Fig. 2(b)-(c). Specifically, ORBCam implements feature extraction directly in the sensor array to avoid full-frame image generation and communication. Since transmitting all extracted features would still incur communication overhead, as hundreds to thousands of

features are typically required for reliable matching. ORBCam further integrates feature matching and outlier rejection into a lightweight digital logic unit, so that only validated correspondences are produced. Finally, to further compress the communication bandwidth, the resulting measurements are quantized and processed by a quantized VIO estimator designed to operate directly on these compressed measurements. Through this tightly coupled sensing and estimation pipeline, ORBCam substantially reduces I/O bandwidth and system energy. The following sections describe the ORBCam architecture and its implementation in detail.

A. ORBCam Architecture and Operation

Architecture overview. Fig. 2(c) illustrates the proposed in-sensor ORB processing architecture, which integrates three tightly coupled components: (i) a pixel array augmented with per-pixel processing elements (PEs), where each PE includes a sample-and-hold (S/H) buffer to latch the pixel intensity for reuse and a modified single-slope ADC (comparator) to perform in-pixel intensity evaluation. (ii) shared row/column analog interconnect buses that enable data exchange and access among pixels along the same row or column. (iii) a lightweight on-chip digital logic unit, local memory, and a microcontroller to coordinate PE/interconnect configuration, schedule comparisons, and manage intermediate feature data and final matched outputs. Compared to a conventional image sensor, newly added components are highlighted in blue.

Processing element (PE) design. ORB FAST detection and rBRIEF descriptor generation are fundamentally comparison-based operations, relying on intensity differencing and pairwise relationships among neighboring pixels. As shown in Fig. 2(c), to support this computation pattern, each pixel is augmented with a lightweight PE tightly coupled with the underlying 4-T active pixel sensor (APS). The PE is designed to provide two essential capabilities: local intensity storage and configurable comparison. A sample-and-hold (S/H) buffer temporarily stores the analog pixel value as a voltage (V_{pixel}), enabling reuse across multiple comparison operations without repeated readout. And, a configurable comparator, derived from a modified single-slope ADC structure, performs intensity differencing between selected pixel pairs or between a pixel and programmable thresholds. Through shared row/column interconnect buses, pixel values can be selectively routed for differential comparison, supporting both FAST threshold tests ($\pm T$) and binary rBRIEF pairwise evaluations. The resulting decisions are forwarded to the on-chip digital logic unit for subsequent operations.

Timing diagram. Fig. 2(a) shows the steady-state execution pipeline. Within each frame, ORB processing consists of FAST detection, descriptor generation, matching, outlier rejection, and measurement quantization. These stages are triggered after exposure and precede data transmission. The quantized measurements are transmitted to the backend for motion estimation using a quantization-aware VIO estimator. Once the pipeline is filled, ORB processing of frame t is executed in parallel with the exposure of frame $t+1$. Since processing latency is shorter than exposure time, feature extraction and matching are effectively hidden behind exposure latency. As a result, the steady-state frame rate is primarily bounded by exposure rather than computation. Quantitative latency and energy benefits are reported in Section V-B.

B. ORB Feature Extraction

1) FAST Detection and Orientation: For each candidate pixel, ORBCam performs FAST detection followed by orientation estimation using lightweight mixed-signal operations. The FAST engine (red box in Fig. 2(c)) scans as a sliding window across the pixel array and evaluates the standard 16-point FAST circle around each center pixel. To improve parallelism, the image is partitioned into multiple spatial grids, allowing multiple engines to operate concurrently across the frame. For FAST evaluation, the intensities of the 16 surrounding pixels are first sampled onto their per-pixel comparator local capacitor (C_{az1}). The center pixel intensity (I_C), stored in the S/H buffer, is then broadcast through the row/column analog interconnect buses to enable capacitive differencing with each ring sample. This forms the analog intensity differences $(I_i - I_C)$, which are then evaluated by a programmable comparator against thresholds $\pm T$. The comparator generates two 16-bit binary masks corresponding to the brighter and darker tests. These masks are forwarded to the digital logic unit, which performs a circular contiguous run-length test with length 9. For pixels satisfying the FAST criterion, ORBCam estimates the keypoint orientation using a hardware-efficient approximation. Instead of full moment computation, we derive a pseudo-gradient from four axis-aligned samples: $g_x = I_5 - I_{13}$ and $g_y = I_1 - I_9$. The signs and relative magnitudes of (g_x, g_y) are quantized into one of eight discrete orientation bins, which are used to steer the subsequent rBRIEF descriptor generation. Finally, the digital logic performs score-based selection and enforces a minimum spatial separation to retain a stable set of keypoints for subsequent descriptor construction and matching.

2) Binary Descriptor Generation: For each detected keypoint, a 256-bit rBRIEF descriptor is constructed from pairwise intensity comparisons within its local neighborhood. rBRIEF represents a feature using binary comparisons between selected pixel pairs, which makes it particularly suitable for hardware implementation based on comparator operations. To achieve rotation invariance, the sampling pattern is rotated according to the quantized orientation estimated in the previous stage. Instead of performing rotation every time, ORBCam precomputes rotated sampling patterns for eight orientation bins and stores them in a look-up table (LUT). Given the keypoint's orientation bin, the corresponding set of pixel index pairs is fetched from the LUT. The descriptor bits are then generated through sequential pairwise comparisons. For each pair, the two pixel voltages are routed through the row/column interconnect buses to the shared PE comparator (green points in Fig. 2(c)), which performs a single binary comparison result. After generating all 256 bits of the descriptor, the resulting descriptors are written to local memory together with the keypoint coordinate and then forwarded to the matching engine.

C. Feature Matching and Motion Validation

After descriptor construction, ORBCam performs feature matching and motion validation to produce compact motion-consistent measurements for the backend estimator. These operations are implemented in the on-chip digital logic unit using fixed-point arithmetic and minimal control logic, introducing negligible area and latency overhead.

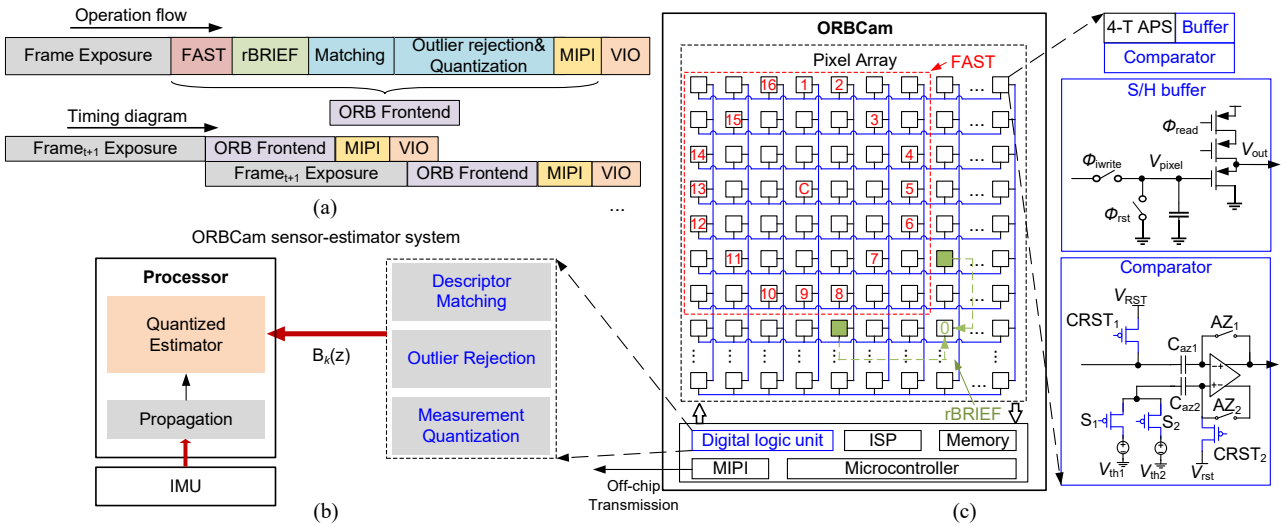


Fig. 2: ORBCam architecture and execution flow. (a) Timing diagram of the ORBCam visual processing pipeline. (b) System overview of the ORBCam-based VIO architecture. ORBCam performs in-sensor feature extraction without full-frame image readout. Digital logic unit conducts descriptor matching, state-free outlier rejection, and measurement quantization before off-chip transmission. The processor receives only quantized visual measurements $B_k(z)$, which are fused with IMU propagation in a quantization-aware estimator. (c) In-sensor ORB processing architecture with augmented pixel array and digital logic unit. Newly added components are highlighted in blue. Circuit design of per-pixel processing element enabling local storage and intensity comparison.

1) Descriptor Matching: Descriptor matching determines the similarity between two keypoints using the Hamming distance between their binary descriptors. As the descriptors are binary, matching reduces to counting the number of bits that differ. In ORBCam, this operation is performed in the on-chip digital logic unit using a fully parallel 256-bit XOR followed by a balanced adder tree to evaluate the popcount. For each descriptor, the nearest neighbor in Hamming space is selected as the candidate correspondence. The resulting matches provide frame-to-frame keypoint associations that serve as the basis for motion estimation.

2) Outlier Rejection: In conventional VIO systems, once images are received by the processor, feature detection and tracking are first performed to obtain visual correspondences. These correspondences are then subjected to outlier rejection to remove unreliable visual measurements, a crucial step for maintaining estimator consistency and preventing drift caused by incorrect matches [26]. Common techniques in the community include the ratio test [28], geometric RANSAC [29], [30], IMU-integrated 2-point RANSAC [31], and reprojection checks when 3D landmarks are available [32], [33], etc. In the ORBCam, we aim to move this stage closer to the sensor. However, since we only transfer quantized pixels or optical flow, two-view geometry-based RANSAC may lack the necessary accuracy. Furthermore, because the processor-side has no access to the original descriptors, the standard ratio test becomes infeasible. On the other hand, performing outlier rejection at the sensor-end introduces specific constraints. First, the algorithm must remain computationally efficient; the complexity of standard 7/8-point RANSAC or k-NN-based ratio tests makes them unsuitable for a digital logic unit. Second, because the sensor lacks access to the global system state, such as current pose, 3D map points, or covariance, it cannot support state-dependent algorithms like 2-point RANSAC or reprojection-based methods. Consequently, we implement a *state-free* motion validation scheme, relying purely on frame-to-frame optical flow consistency.

First, bidirectional matching is enforced through a symmetric cross-check: a correspondence $(i \leftrightarrow j)$ is retained only if $j^\star(i) = j$ and $i^\star(j) = i$. For each retained match, an optical flow is computed as $\mathbf{f}_i = [u_i^k - u_i^{k-1},\, v_i^k - v_i^{k-1}]^\top$. We estimate the mean flow $\bar{\mathbf{f}}$ and covariance Σ_f from all matches in the frame and apply a chi-square gate $d_i = (\mathbf{f}_i - \bar{\mathbf{f}})^\top \Sigma_f^{-1} (\mathbf{f}_i - \bar{\mathbf{f}}) \le \tau$. A maximum flow bound $\|\mathbf{f}_i\| \le f_{\max}$ is additionally enforced to reject large-displacement outliers. These lightweight geometric checks can be implemented efficiently using shared accumulators and a small matrix evaluation unit.

3) Measurement Quantization: After motion validation, ORBCam outputs compact motion measurements instead of raw feature data. To further reduce bandwidth, we adopt differential measurement quantization. Rather than directly quantizing each measurement z_k, the sensor quantizes the temporal difference:

$$B_k = Q(z_k - z_{k-1}^q), \qquad z_k^q = z_{k-1}^q + B_k \tag{7}$$

Because inter-frame motion is typically small, the differential signal has a reduced dynamic range and can be represented using fewer bits with minimal quantizaiton error. This operation is implemented in the digital logic unit using a lightweight datapath consisting of registers, subtractors, and a fixed-point quantizer. For the first measurement z_0^q, we store and send its quantized raw value. The previous quantized measurement z_{k-1}^q is stored in a small register file. At each frame, the difference $(z_k - z_{k-1}^q)$ is computed using a subtractor, followed by a uniform quantization unit that produces the encoded measurement B_k. An accumulator then reconstructs z_k^q for use in the next frame. The resulting quantized measurements are transmitted to the backend processor, where a quantization-aware VIO estimator processes them within a MAP-based estimation framework.

D. Communication Protocol Design

Efficient feature track offloading requires a lean communication protocol to prevent protocol overhead from dominating the data packet, especially when data association is also part of the communication. For the 752×480 resolution used in our experiments, a pixel coordinate requires at least 19 bits, and tracking IDs for 1024 features require 10 bits each, we minimize bandwidth through a specialized packet structure. The packet contains global metadata (header, timestamp, optical flow statistics) followed by a tracked feature indicator, quantized tracked flows, and new feature raw pixels, secured by a checksum. Rather than transmitting explicit 10-bit IDs, the tracked feature indicator establishes data association relative to previous packets. This reduces the per-feature ID overhead to approximately 1 bit when tracking is consistent, significantly lowering the total bitrate.

IV. ORBCam-VIO

ORBCam fundamentally changes the measurement interface of a conventional VIO system. Instead of receiving full image frames and extracting features on the host processor, the VIO backend directly operates on compact visual measurements produced within the sensor subsystem (Fig. 2(b)). Since these measurements are represented in a reduced-bit quantized form, we adopt the quantized MAP-based VIO estimator proposed in [20], which is specifically designed to process quantized observations in a statistically consistent manner. Due to its simplified formulation, efficient communication protocol, and competitive performance, we employ the zQVIO2 (Measurement Quantization VIO 2.0) framework in our system. Note that the state and IMU propagation remains identical to the conventional formulation (see Section II).

Specifically, given a K-bit quantized value $\mathbf{b}(\mathbf{z})$, where $\mathbf{z}$ denotes the raw measurement defined in (5), the posterior can be formulated as:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}) \prod_{i=1}^{m} \left(\mathbf{Q}(\chi_i^1) - \mathbf{Q}(\chi_i^2) \right) \tag{8}$$

where $Q(\cdot)$ is the Gaussian Q-function and χ_i^1, χ_i^2 represent the normalized quantization lower and upper bounds.

Assuming a Gaussian prior, this MAP problem can be solved recursively using an EKF-like update without requiring ad-hoc noise inflation. The quantized measurement function is expressed as:

$$\mathbf{r}^m := \mathbf{b}(\mathbf{z}) - \mathbf{h}(\hat{\mathbf{x}}) \simeq \mathbf{H}\tilde{\mathbf{x}} + \mathbf{n}_b \tag{9}$$

where $\mathbf{b}(\mathbf{z})$ represents the multi-bit quantized measurements and $\mathbf{n}_b$ denotes noise induced by quantization. For detailed noise modeling, we refer the interested reader to [20]. In our implementation, the quantized-likelihood update is implemented into the standard $\sqrt{\text{VINS}}$ update pipeline [27], after triangulation and nullspace projection. This integration preserves the computational complexity of the original estimator while enabling consistent inference with ORBCam's multi-bit quantized outputs.

V. Experimental Results

We evaluate ORBCam-VIO from two perspectives. First, we quantify the front-end sensing benefits, i.e., energy, latency, and off-chip bandwidth, using circuit-level models under matched CMOS technology and I/O assumptions. Second, we assess the algorithmic impact of ORBCam outputs on state estimation by feeding the simulated feature correspondences into representative VIO back-ends (MSCKF and SLAM) and reporting VIO performance. This separation allows us to isolate (i) the sensor-side gains from eliminating full-frame readout and reducing data movement, and (ii) whether the proposed feature-level interface preserves the information needed for accurate motion estimation.

A. Experimental Methodology

1) ORBCam Evaluation Setup: **Hardware configuration and evaluation methodology.** ORBCam is designed in a standard 65-nm Complementary Metal-Oxide-Semiconductor (CMOS) process, consistent with current commercial image sensor technology [34]. The digital logic unit is described in RTL and synthesized using a Synopsys/Cadence Electronic Design Automation (EDA) flow, while on-chip memory is emulated using synthesized SRAM blocks. The pixel array adopts a conventional 4-T active pixel sensor (APS) design with additional circuitry to support in-sensor feature extraction, and the Energy and latency are estimated using circuit-level analytical models derived from prior studies [35], [36]. These models account for pixel array readout, mixed-signal processing, ADC operations, SRAM accesses, and off-chip I/O transmission.

Feature generation is simulated using the proposed ORBCam processing pipeline, which models the circuit-level operations of the sensor architecture. To account for environmental variations and hardware non-idealities, noise models derived from prior hardware characterization [37] are injected during the operations. We evaluate three feature budgets (256, 512, and 1024 per frame) to study the trade-off between sensing cost and motion estimation accuracy.

Baseline. We compare ORBCam against a conventional CMOS image sensor (CNV-CIS) baseline that captures and transmits full-resolution image frames. To ensure a fair comparison, both systems operate at a resolution of 752×480 and a frame rate of 100 FPS under the same technology assumptions. For off-chip communication, we assume a four-lane MIPI CSI-2 interface operating at 1.2 Gbps per lane.

Evaluation metrics. Due to the difficulty of accurately modeling host processor energy across diverse software stacks and workloads, our energy and latency evaluation mainly focuses on the sensor front-end. This is justified because, in resource-constrained VIO systems, the dominant energy cost typically arises from data acquisition and transmission rather than host processor computation. Therefore, we evaluate the sensing and communication of ORBCam and compare it with the CNV-CIS baseline. We report sensing energy in milliwatts (mW) and latency in milliseconds per frame (ms/frame), focusing on sensor-side processing and data transmission. We also report the off-chip communication bandwidth required by each system. The resulting quantized measurements are then fed into a VIO estimator to evaluate the impact of ORBCam on motion estimation accuracy.

2) VIO Estimator Evaluation Setup: We consider two commonly used VIO estimation setups. In `MSCKF`, visual features are used to constrain motion and are marginalized after providing measurements, without being included as persistent state variables. In `SLAM`, selected visual features are explicitly maintained in the state vector and jointly optimized with the pose [38]. We compare three system variants. (1) **VIO (Baseline)**: uses the original CNV-CIS front-end with the standard VIO pipeline. (2) **VIO**: the same

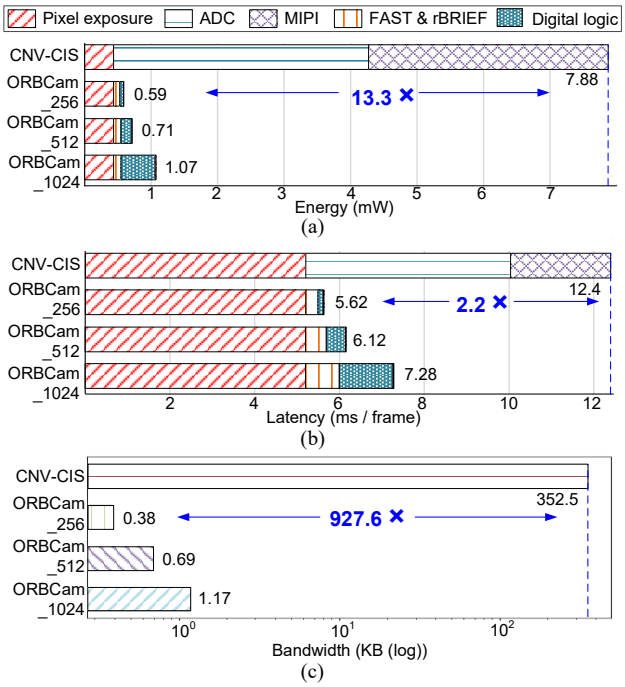


Fig. 3: ORBCam evaluation results under 5-bit measurement quantization. (a) ORBCam energy breakdown. At a feature budget of 256, ORBCam achieves 13.3× energy efficiency compared with a conventional image sensor (CNV-CIS). (b) ORBCam latency breakdown. At the same feature budget, ORBCam achieves a 2.2× speedup compared with CNV-CIS. (c) Per-frame communication bandwidth comparison. ORBCam reduces per-frame communication by up to 927.6× compared with CNV-CIS.

estimator using our simulated ORB front-end outputs *without* measurement quantization and a standard $\sqrt{\text{VINS}}$ backend; (3) **ZQVIO (Proposed)**: our quantization-aware VIO system, which integrates ORBCam outputs *with* measurement quantization and a quantization-aware state estimator.

B. ORBCam Evaluation

1) Energy: We evaluate the sensing energy of ORBCam using the circuit-level analytical models and tools described in Section V-A.1. Specifically, the total energy is decomposed into pixel array readout, mixed-signal processing, ADC operations, on-chip memory access, digital logic operations, and off-chip data transmission. Each component is estimated using technology-scaled circuit models and the operation counts derived from the ORBCam processing pipeline. We evaluate ORBCam across three feature budgets, while both ORBCam and the CNV-CIS baseline operate at 100 FPS. Fig. 3(a) presents the detailed energy breakdown. The ORBCam energy savings primarily stem from two architectural factors: (i): The conventional pipeline digitizes (ADC) and transmits (MIPI) all pixels per frame. ORBCam removes full-frame ADC conversion and high-bandwidth pixel transmission, transmitting only compact quantized feature coordinates. (ii): Most feature detection and comparison operations are performed in the analog domain, avoiding energy-intensive digital computation and memory accesses. These results demonstrate that moving feature extraction into the sensor fundamentally shifts the energy bottleneck. At a budget of 256 features per frame, ORBCam consumes only 0.59 mW, achieving a 13.3× improvement in sensing energy efficiency compared with the CNV-CIS baseline.

TABLE I: Feature statistics under different feature budgets on the EuRoC dataset. The table reports the average number of tracked and newly detected features per frame.

Feature Budget	Tracked Features	New Features	Total Features
256	91.2	102.1	193.3
512	170.1	183.7	353.8
1024	305.5	307.1	612.6

2) Latency: Latency is evaluated by analyzing the execution schedule of the ORBCam processing pipeline and summing the latency of each sensing and processing stage. These latencies are estimated using the same circuit-level models, tools, and operation counts described in Section V-A.1. As a result, the total end-to-end latency of ORBCam is 5.62 ms at 256 feature budget and 7.28 ms at 1024 feature budget, as a larger number of features requires additional processing cycles. In contrast, the conventional architecture requires 12.4 ms per frame for front-end sensing and data transmission. Therefore, ORBCam achieves a 2.2× reduction in front-end latency. Fig. 3(b) shows the latency breakdown across different stages. Since ORBCam significantly reduces both processing latency and transmission time, the system is no longer limited by computation or I/O but instead by the exposure time of the image sensor. Consequently, if the exposure duration is shortened, the system can theoretically support higher frame rates.

3) Communication Bandwidth Analysis: We analyze the actual communication cost across different feature budgets on the EuRoC dataset. The tracking statistics are reported in Table I. Fig.3(c) compares the per-frame data size under different visual front-end representations. For the conventional CIS pipeline, which transmits full-resolution images with 8-bit encoding, the communication cost reaches hundreds of kilobytes (KB) per frame. In contrast, ORBCam transmits only compact multi-bit motion measurements, resulting in a substantially smaller data size per feature. At a feature budget of 256, ORBCam achieves up to a 927.6× reduction in communication bandwidth compared with the CNV-CIS baseline. Furthermore, as the feature budget increases from 256 to 1024, the transmission bandwidth of the ORBCam scales linearly. However, this scaling exhibits a decreasing slope, as the fixed communication protocol overheads—such as timestamps and other metadata—are smoothed over an increasing number of features. It is important to note that tracking-level communication includes not only measurements but also data association. Without careful protocol design, such data can dominate bandwidth consumption. The proposed encoding scheme ensures that both motion measurements and association information remain compact.

C. VIO Performance

To evaluate whether the proposed sensor-side ORB frontend preserves estimation performance, we report VIO accuracy on the EuRoC MAV [39] benchmark under two representative back-ends: an MSCKF-style filter and a sliding-window SLAM estimator. For each sequence, we report the average absolute trajectory error (ATE), expressed in degrees for orientation error and meters for position estimation error. Table II shows that replacing the conventional image-centric ORB frontend with ORBCam feature tracks yields compa-

TABLE II: Absolute Trajectory Error (ATE) (deg/m) on the EuRoC MAV dataset under different system configurations. **ORB** denotes feature tracking from raw images using a conventional ORB frontend. **ORBCam** denotes feature tracks generated by the proposed ORBCam frontend. **VIO** denotes the conventional VIO pipeline using $\sqrt{\text{VINS}}$, while **zQVIO** denotes the measurement-quantized VIO using quantized visual measurements. Results are reported under both MSCKF and SLAM modes up to 1024 tracked features; zQVIO uses 5 bits per incremental flow measurement.

VIO Setup	Config	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
MSCKF	ORB + VIO	2.21 / 0.17	0.57 / 0.14	1.59 / 0.25	0.69 / 0.28	0.46 / 0.28	0.67 / 0.06	1.90 / 0.07	1.79 / 0.10	0.90 / 0.05	1.29 / 0.10	1.49 / 0.14
	ORBCam + VIO	1.67 / 0.16	0.53 / 0.11	1.74 / 0.27	0.69 / 0.27	0.60 / 0.30	0.60 / 0.08	1.87 / 0.09	1.85 / 0.13	1.01 / 0.06	1.35 / 0.10	1.20 / 0.16
	ORBCam + zQVIO	1.82 / 0.17	0.57 / 0.12	1.90 / 0.28	0.62 / 0.23	0.60 / 0.33	0.58 / 0.08	1.83 / 0.08	1.78 / 0.13	1.02 / 0.06	1.36 / 0.11	1.10 / 0.12
SLAM	ORB + VIO	1.76 / 0.12	0.62 / 0.14	1.43 / 0.17	0.75 / 0.16	0.56 / 0.25	0.56 / 0.03	1.83 / 0.08	1.65 / 0.07	0.92 / 0.05	1.39 / 0.09	1.41 / 0.15
	ORBCam + VIO	1.64 / 0.12	0.64 / 0.08	1.60 / 0.20	0.74 / 0.19	0.69 / 0.48	0.54 / 0.04	2.02 / 0.07	2.14 / 0.11	0.96 / 0.04	1.26 / 0.09	1.12 / 0.17
	ORBCam + QVIO	1.67 / 0.11	0.64 / 0.08	1.57 / 0.18	0.80 / 0.19	0.70 / 0.39	0.67 / 0.04	1.91 / 0.07	1.93 / 0.13	0.84 / 0.04	1.52 / 0.10	1.27 / 0.17

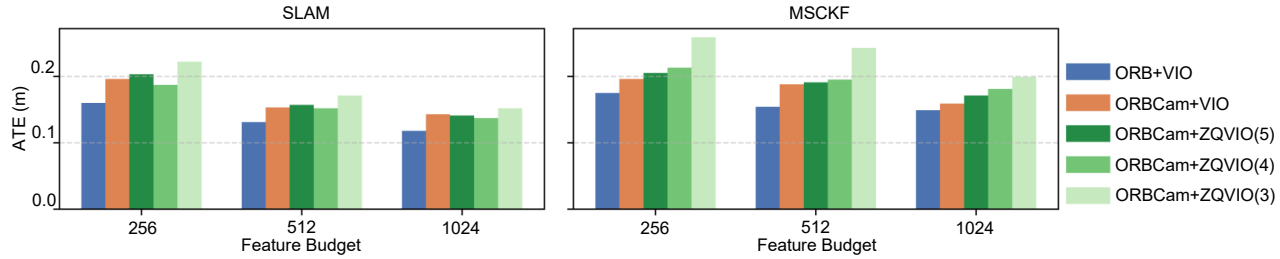


Fig. 4: VIO average position error (m) across different sequences in EurocMAV in different setups.

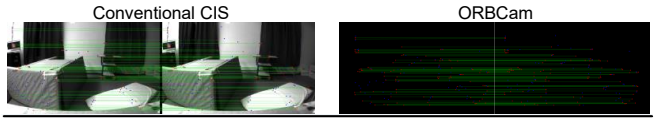
rable ATE across all sequences. Moreover, introducing 5-bit quantization of the incremental flow measurements (zQVIO) incurs only marginal degradation, indicating that substantial bandwidth reduction can be achieved without sacrificing trajectory accuracy.

Fig. 4 shows the average VIO position error on the EuRoC dataset under different feature budgets and quantization levels. Compared with the conventional ORB frontend, ORBCam results in slightly larger estimation errors, which is expected due to the simplified sensing and processing pipeline. However, the overall performance remains comparable across both SLAM and MSCKF back-ends. We also evaluate the effect of measurement quantization. As the quantization bit-width decreases, the estimation error gradually increases due to the larger quantization noise. Nevertheless, using 4–5 bit quantization produces accuracy close to the non-quantized ORBCam pipeline. Finally, increasing the number of tracked features consistently improves estimation accuracy. This trend reflects the typical trade-off in VIO systems: a larger number of visual features provides stronger geometric constraints and reduces estimation error, but also increases sensing and processing cost.

D. Summarization and Discussion

The above evaluations characterize the end-to-end behavior of ORBCam-VIO, spanning from front-end sensing efficiency to backend motion estimation accuracy. Across different feature budgets, ORBCam achieves comparable VIO accuracy to conventional pipelines while substantially reducing sensing energy and communication bandwidth without original pixel acquisition (Fig. 5).

Several points should be noted. First, the energy comparison in this work focuses on the sensing front-end and does not include the backend processor without accurate end-to-end system level evaluation, as discussed in Sec. V-A.1. However, ORBCam directly outputs validated feature correspondences, the backend computation is expected to be reduced compared with conventional pipelines. Second, the current evaluation is based on architecture modeling and circuit-level simulation (but with real imagery data).



	Energy (mW)	Latency (ms)	Bandwidth (KB)	ATE (deg / m)
Conventional	7.88	12.4	352.5	1.590 / 0.108
ORBCam	0.59	5.62	0.38	1.724 / 0.129
Improvement	**13.3 ×**	**2.2 ×**	**927.6 ×**	——

ORBCam adopts a 256-feature budget with 5-bit quantization.

Fig. 5: Comparison between conventional image sensor and ORBCam.

Although the design has not yet been fabricated, the evaluation is grounded in established circuit models and EDA-based synthesis of the digital logic units. Future work will include full-chip implementation and silicon validation of the proposed sensor architecture in a real VIO system.

VI. Conclusion and Future Work

This paper presents ORBCam, a machine-centric sensing architecture that directly generates quantized sparse ORB feature measurements instead of full-resolution images. By tightly co-designing sensing and feature extraction, ORBCam eliminates redundant pixel-level acquisition and transmission, substantially reducing front-end energy consumption. Furthermore, the sensing architecture is jointly optimized with backend VIO algorithms to enable efficient end-to-end feature-based processing. Experimental results on the EuRoC dataset demonstrate that ORBCam achieves comparable motion estimation accuracy to conventional image-sensor-based pipelines. At the same time, ORBCam improves sensing energy efficiency by up to $13.3\times$, reduces sensing latency by $2.2\times$, and decreases transmission bandwidth by up to $927.6\times$. Beyond VIO, ORBCam illustrates a broader machine-centric sensing paradigm in which sensors generate task-relevant representations rather than dense images. Such a sensing interface can significantly reduce data movement and computation for feature-driven perception workloads. Future work will include silicon implementation and validation of the proposed architecture, as well as extending the sensing framework to support additional perception tasks that rely on sparse visual measurements, such as eye tracking for AR/VR devices, motion-based gesture recognition.

REFERENCES

[1] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[2] T. Ma, A. J. Boloor, X. Yang, W. Cao, P. Williams, N. Sun, A. Chakrabarti, and X. Zhang, "Leca: In-sensor learned compressive acquisition for efficient machine vision on the edge," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–14.

[3] Y. Liang, Z. Yi, T. Ma, and W. Cao, "Lorasense: Learnable low-rank acquisition in sensors for efficient edge machine vision," in *2025 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2025, pp. 1–7.

[4] Y. Liang and W. Cao, "Late breaking results: Less sense makes more sense: In-sensor compressive learning for efficient machine vision," in *2025 62nd ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2025, pp. 1–2.

[5] "Reality labs chief scientist outlines a new compute architecture for true ar glasses," https://www.roadtovr.com/michael-abrash-iedm-2021-compute-architecture-for-ar-glasses/.

[6] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze, "Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1106–1119, 2019.

[7] E. J. Msechu, S. I. Roumeliotis, A. Ribeiro, and G. B. Giannakis, "Decentralized quantized kalman filtering with scalable communication cost," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3727–3741, 2008.

[8] Q. Picard, S. Chevobbe, M. Darouich, and J.-Y. Didier, "Image quantization towards data reduction: robustness analysis for slam methods on embedded platforms," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 4158–4162.

[9] O. Christie, J. Rego, and S. Jayasuriya, "Analyzing sensor quantization of raw images for visual slam," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 246–250.

[10] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[11] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys, "Hyperpoints and fine vocabularies for large-scale location recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2102–2110.

[12] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization." in *Robotics: Science and Systems*, vol. 1, 2015, p. 1.

[13] L. Baroffio, A. E. Redondi, M. Tagliasacchi, and S. Tubaro, "A survey on compact features for visual content analysis," *APSIPA Transactions on Signal and Information Processing*, vol. 5, p. e13, 2016.

[14] D. Van Opdenbosch and E. Steinbach, "Collaborative visual slam using compressed feature exchange," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 57–64, 2018.

[15] M. Mera-Trujillo, B. Smith, and V. Fragoso, "Efficient scene compression for visual-based localization," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 1–10.

[16] L. Zheng, K. Xu, J. Jiang, M. Wei, B. Zhou, and H. Cheng, "Real-time efficient environment compression and sharing for multi-robot cooperative systems," *IEEE Transactions on Intelligent Vehicles*, 2024.

[17] M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale, "The gist of maps-summarizing experience for lifelong localization," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2767–2773.

[18] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving rgb-d slam in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110–122, 2017.

[19] J. Gomez, S. Patel, S. S. Sarwar, Z. Li, R. Capoccia, Z. Wang, R. Pinkham, A. Berkovich, T.-H. Tsai, B. De Salvo, and C. Liu, "Distributed on-sensor compute system for ar/vr devices: A semi-analytical simulation framework for power estimation," 2022. [Online]. Available: https://arxiv.org/abs/2203.07474

[20] C. Chen, Y. Peng, and G. Huang, "Qvio2: Quantized map-based visual-inertial odometry," in *Proc. International Conference on Robotics and Automation*, Atlanta, GA, May 2025.

[21] Y. Peng, C. Chen, and G. Huang, "Quantized visual-inertial odometry," in *Proc. International Conference on Robotics and Automation*, Yokohama, Japan, May 2024.

[22] J. Kühne, M. Magno, and L. Benini, "Low latency visual inertial odometry with on-sensor accelerated optical flow for resource-constrained uavs," *IEEE Sensors Journal*, vol. 25, no. 5, pp. 7838–7847, 2025.

[23] J. Chen, S. J. Carey, and P. Dudek, "Feature extraction using a portable vision system," in *IEEE/RSJ Int. Conf. Intell. Robots Syst., Workshop Vis.-based Agile Auton. Navigation UAVs*, vol. 2, 2017, p. 3.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[25] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3565–3572.

[26] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[27] Y. Peng, C. Chen, K. Wu, and G. Huang, "sqrt-vins: Robust and ultra-fast square-root filter-based 3d motion tracking," *IEEE Transactions on Robotics*, Sept. 2025.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[30] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[31] C. Troiani, A. Martinelli, C. Laugier, and D. Scaramuzza, "2-point-based outlier rejection for camera-imu systems with applications to micro aerial vehicles," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 5530–5536.

[32] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[33] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[34] J. Ohta, *Smart CMOS image sensors and applications*. CRC press, 2020.

[35] T. Ma, Y. Feng, X. Zhang, and Y. Zhu, "Camj: Enabling system-level energy modeling and architectural exploration for in-sensor visual computing," in *Proceedings of the 50th annual international symposium on computer architecture*, 2023, pp. 1–14.

[36] T. Ma, Z. Gao, Z. Chen, R. Kakarala, C. Shan, W. Cao, and X. Zhang, "Systematic methodology of modeling and design space exploration for cmos image sensors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2025.

[37] J. E. Farrell, P. B. Catrysse, and B. A. Wandell, "Digital camera simulation," *Applied optics*, vol. 51, no. 4, pp. A80–A90, 2012.

[38] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *IROS 2019 Workshop on Visual-Inertial Navigation: Challenges and Applications*, Macau, China, Nov. 2019. [Online]. Available: https://github.com/rpng/open_vins

[39] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.