

# Quantized Visual-Inertial Odometry

Yuxiang Peng, Chuchu Chen, and Guoquan Huang

**Abstract**—As edge devices equipped with cameras and inertial measurement units (IMUs) are emerging, it holds huge implications to endow these mobile devices with spatial computing capability. However, ultra-efficient visual-inertial estimation at the size, weight and power (SWAP)-constrained edge devices to provide accurate 3D motion tracking remains challenging. This is exacerbated by data transfer (between different processors and memory) that consumes significantly more energy than computing itself. To push the state of the art, this paper proposes the *first-of-its-kind* quantized visual-inertial odometry (QVIO) to offer energy-efficient 3D motion tracking. In particular, we first quantize raw visual measurements in an intuitive way with a given small number of bits and then perform an EKF update with these quantized measurements (termed zQVIO). To improve this ad-hoc quantizer (although it works well in practice), we systematically quantize each measurement residual into a single bit and perform maximum-a-posterior (MAP) estimation. Thanks to these quantizers, the proposed QVIO estimators significantly reduce the data transfer and thus improve energy efficiency. As shown in our extensive experiments, the proposed residual-quantized VIO (rQVIO) achieves remarkably competing performance even when using an average of only 3.7 bits per measurement, equivalent to a data reduction of 8.6 times compared to transmitting single-precision measurements.

## I. INTRODUCTION

As edge devices equipped with cameras and inertial measurement units (IMUs) are emerging, such as augmented reality (AR) glasses, virtual reality (VR) headsets, and micro aerial vehicles (MAVs), it holds great potential to endow these mobile devices with efficient spatial computing capability of high-accuracy motion tracking and human-like scene understanding [1]–[3]. However, robust and efficient visual-inertial 3D motion tracking at the edge – which imposes significant size, weight, and power (SWAP) constraints – to enable consistent and immersive situational awareness still eludes computer vision and robotics communities. Even though edge computing is increasing, energy-efficient computing is still in great demand, as low-end edge computers and single-precision microcontrollers still dominate the market. Moreover, one of the major factors causing the performance and power bottlenecks on edge devices is data transfer, including not only wireless data transfer but even transferring data from one chip inside the device to another. For example, on Meta XR wearable devices, the essential techniques (SLAM and hand-tracking) use most of their power simply moving data to and from RAM [4]. This calls for rethinking how computing should be handled on these edge devices and how to reduce the amount of data transfer

to reduce power consumption so as to meet battery life and heat requirements.

Realizing that data transfer consumes most of the energy on visual-inertial edge computing systems where the raw imagery data is typically processed in a central off-sensor edge-processor (i.e., aggregator), Gome et al. [5] proposed a distributed on-sensor compute architecture. This architecture includes on-sensor processors for the first level of processing and a nearby aggregator for further processing and thus allows for minimum data movement with rapid and localized inference on the sensors. As such, it is possible to transmit the complete raw image from the cameras to the on-sensor processors through the low-energy and high-bandwidth uTSV interconnects, while only the pre-processed visual data is transferred from the sensors to the aggregator through the energy-hungry MIPI interfaces. Although this on-sensor compute architecture, from a hardware perspective, decreases the usage of the energy-intensive serial interface due to the on-sensor processing and subsequent feature compression, the data (e.g., detected and tracked visual feature observations) required to transmit over the MIPI interface for visual-inertial estimation may still be over-killing. Similarly, many MAVs employ heterogeneous embedded computing systems including system-on-chip (SoC) for complex computations such as visual perception and microcontroller unit (MCU) for state estimation and flight control. UART is often used to transmit and receive serial data between SoC and MCU, but with limited bandwidth. Also, because MCU is a small computer on a single integrated circuit and provides only minimal memory and processing power, it is always compelling to reduce the data transfer as much as possible from SoC to MCU to perform estimation and control.

Following the power, in this work, we, therefore, aim to quantize visual measurements to minimize data movements between different (co-)processors and memory (co-processor and host computer), and *for the first time ever*, design quantized visual-inertial odometry (QVIO) algorithms. In particular, inspired by near-sensor or on-sensor processing [6], [7], we preprocess images and transmit only the quantized measurements or residuals to the host computer where visual-inertial estimation is performed. This significantly reduces the data transfer, and releases some computation from the host processor, while facilitating the fusion of vision, inertial, and potentially other sensor data. Leveraging this system architecture, we develop two QVIO estimators. First, we directly quantize a visual measurement into a given number of bits (e.g., chosen by trial and error), and then perform EKF update to fuse the quantized measurements, which is thus termed measurement-quantized VIO (zQVIO). By doing so, it bypasses the need for data transfer from the host processor to the co-processor. While this zQVIO appears to be intuitive and simple, it generally works well in practice, but with

This work was partially supported by the University of Delaware (UD) College of Engineering, Delaware NASA/EPSCoR Seed Grant, NSF (IIS-1924897, SCH-2014264), Google ARCore, and Meta Reality Labs.

The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. Email: {yxpeng, ccchu, ghuang}@udel.edu

no performance guarantee and requires an oracle choice of quantization bits. As such, we further develop the residual-quantized VIO (rQVIO) algorithm, inspired by the sign-of-innovation (SOI)-KF from wireless sensor networks [8]. The proposed rQVIO performs feature tracking and residual quantization on the co-processor and transmits only *a single* bit of each quantized measurement residual to the host processor for estimation. We then formulate the MAP problem of the quantized visual-inertial estimation, which is solved recursively by EKF-like update assuming the previous and current states are Gaussian. Note that we also exchange state linearization points between the co-processor and the host computer, and rather than transferring the entire state each time, only transmit the quantized increment  $\delta\mathbf{x}$  to reduce extra communication overhead.

## II. RELATED WORK

While there is no prior work in visual-inertial systems (VINS [1]) leveraging quantization for state estimation, quantization is not a new concept and has been used in different tasks, as it can significantly reduce the memory footprint, and operations on quantized data require fewer bit-wise actions. For example, quantization is often used in deep learning to reduce inference time, model size, and power consumption [9]. In robotics, such technique has been leveraged to perform descriptor compression [10]–[12] or image compression [13]–[15] to reduce memory or communication requirement.

In state estimation, one can also quantize measurements and perform updates with them. As a predominant example, the SOI-KF, modeling quantized residual as Gaussian tail distribution, uses only 1-bit to perform estimation and achieve only minimal accuracy drop [8], which is later extended to include multi-bit and is able to have very close performance compared with using the real-value measurement [16]. However, it is limited to sequential updates, which is inefficient as more BLAS level-3 [17] operation is desirable in modern computers. The quantization-based estimator was also preliminarily investigated in multi-robot cooperative localization [18] which proposed to solve a batch MAP (instead of minimum mean square error, or MMSE) iteratively to reduce linearization error and was shown to reduce inter-robot data communication significantly. This idea has been further extended to include multi-bit and allow the use of local raw data along with quantized data from other robots [19]–[21]. Nevertheless, no work has yet investigated quantization in visual-inertial estimation on an edge device.

## III. QUANTIZATION-BASED ESTIMATION

In this section, we present a new formulation of quantization-based state estimation, which serves as the foundation for the proposed QVIO algorithms. In particular, we propose to perform estimation based on two different quantization strategies: (i) directly quantizing raw measurements into a given number of bits, and (ii) quantizing measurement residuals into a single bit.

Assume state  $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$  with its estimate,  $\hat{\mathbf{x}}$ , and

covariance,  $\mathbf{P}$ .<sup>1</sup> Given the measurement  $\mathbf{z} = [\dots z_i \dots]$ , the measurement function and its linearization can be derived as:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{n} \Rightarrow \mathbf{r} = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) \simeq \mathbf{H}\hat{\mathbf{x}} + \mathbf{n} \quad (1)$$

$$\mathbf{r} = [\dots r_i \dots]^\top, \mathbf{H} = [\dots \mathbf{H}_i \dots]^\top \quad (2)$$

where  $\mathbf{r}$  and  $\mathbf{H}$  are the residual and measurement Jacobian, respectively.  $\mathbf{n}$  is the white Gaussian measurement noise  $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ . To reduce the data transfer, we quantize the raw measurements (or their residuals) to binary data as follows:

$$\mathbf{b}(\mathbf{z}) = [\dots b_i \dots] \quad (3)$$

where  $b_i$  is a 1-bit scalar measurement. Note that this quantization introduces severe nonlinearity and estimation with the quantized data clearly is not trivial. For this reason, we now formulate the quantized estimation as the following MAP optimization problem similar as [18]:

$$\arg \max p(\mathbf{x}|\mathbf{b}(\mathbf{z})) = \arg \max p(\mathbf{x}) \prod_{i=1}^m p(b_i|\mathbf{x}) \quad (4)$$

To solve this problem recursively, EKF-like update equations can be employed to approximate the MAP estimate.

### A. Estimation with Measurement Quantization

We first directly quantize the data without modifying the estimator. That is, use the quantized measurement to compute the measurement residual when formulating the measurement function Eq. (1):

$$\mathbf{r}^m := \mathbf{B}(\mathbf{z}) - \mathbf{h}(\hat{\mathbf{x}}) \simeq \mathbf{H}\hat{\mathbf{x}} + \mathbf{n}_b \quad (5)$$

where  $\mathbf{b}(\mathbf{z})$  represents the multi-bit quantized measurements,  $\mathbf{n}_b$  denotes noise after quantization. Any estimator can use this quantized measurement, for example, using the following EKF update equations:

$$\hat{\mathbf{x}}^\oplus = \hat{\mathbf{x}}^\ominus \boxplus \mathbf{P}^\ominus \mathbf{H}^\top (\mathbf{H} \mathbf{P}^\ominus \mathbf{H}^\top + \mathbf{R})^{-1} \mathbf{r}^m \quad (6)$$

$$\mathbf{P}^\oplus = \mathbf{P}^\ominus - \mathbf{P}^\ominus \mathbf{H}^\top (\mathbf{H} \mathbf{P}^\ominus \mathbf{H}^\top + \mathbf{R})^{-1} \mathbf{H} \mathbf{P}^\ominus \quad (7)$$

where  $\hat{\mathbf{x}}^\ominus/\mathbf{P}^\ominus$  and  $\hat{\mathbf{x}}^\oplus/\mathbf{P}^\oplus$  are the state mean / covariance before and after the update, respectively. As compared with the raw measurements Eq. (1), the only difference is the residual  $\mathbf{r}^m$  is calculated with the quantized measurement, while the Jacobian is computed as usual.

### B. Estimation with Residual Quantization

Drawing inspiration from the SOI-KF [8], we quantize the measurement residual as follows. For the sake of clarity, we assume a scalar measurement  $h(\cdot)$  in our explanation:

$$b := \text{sign}(z - h(\hat{\mathbf{x}})) = \begin{cases} +1 & \text{if } z - h(\hat{\mathbf{x}}) > 0 \\ -1 & \text{if } z - h(\hat{\mathbf{x}}) < 0 \end{cases} \quad (8)$$

The term  $p(b_i|\mathbf{x})$  in Eq. (4) can be derived to be expressed in terms of the Q-function, which gives the tail probability of the standard normal distribution:

$$p(b_i = b|\mathbf{x}) = Pr\{b(z - h(\hat{\mathbf{x}})) > 0|\mathbf{x}\} \quad (9)$$

$$= Pr\{b(h(\mathbf{x}) + n - h(\hat{\mathbf{x}})) > 0|\mathbf{x}\} \quad (10)$$

$$= Pr\left\{\frac{n}{\sigma} > b \frac{h(\hat{\mathbf{x}}) - h(\mathbf{x})}{\sigma} \middle| \mathbf{x}\right\} := \mathbf{Q}(\chi) \quad (11)$$

where  $\chi = \frac{b}{\sigma}(h(\hat{\mathbf{x}}) - h(\mathbf{x}))$  and  $\sigma$  is a scalar of noise standard deviation that normalizes the measurement noise.

<sup>1</sup>Note that throughout the paper  $\hat{\mathbf{x}}$  is used to denote the *current* best estimate of a random variable  $\mathbf{x}$  with  $\tilde{\mathbf{x}} = \mathbf{x} \boxplus \hat{\mathbf{x}}$  denotes the error state. The “ $\boxplus$ ” and “ $\boxminus$ ” operations map elements to and from a given manifold and equate to simple “+” and “-” for vector variables [22].

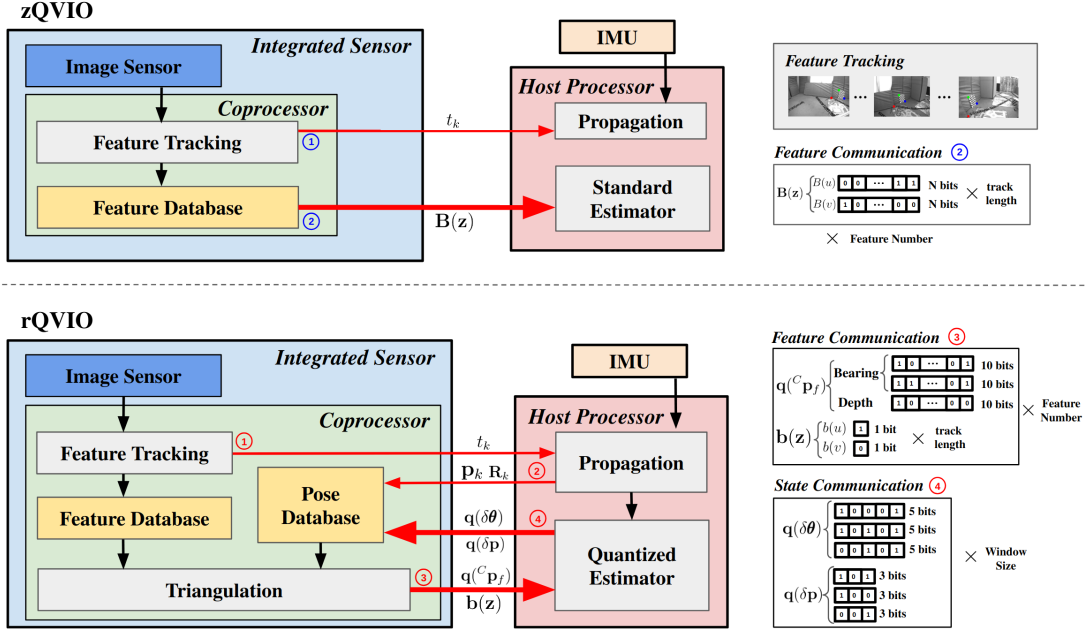


Fig. 1: The system architectures of the proposed measurement-quantized VIO (zQVIO) and residual-quantized VIO (rQVIO).

Now consider multiple measurements. By substituting Eq. (11) back to the posterior Eq. (4), we derive the MAP estimates as:

$$\arg \max p(\mathbf{x}|\mathbf{b}(\mathbf{z})) = \arg \max p(\mathbf{x}) \prod_{i=1}^m \mathbf{Q}(\chi_i) \quad (12)$$

Take logarithm on Eq. (12) and perform second-order Taylor series expansion on the  $\log(\mathbf{Q}(\chi_i))$  (the high-order terms ( $\geq 2$ ) of  $h(\mathbf{x})$  are ignored), the MAP problem is equivalent to minimizing:

$$\mathcal{C}(\mathbf{x}) = \|\mathbf{x} \boxminus \hat{\mathbf{x}}\|_{\mathbf{P}}^2 + \sum_{i=1}^m \|\alpha_i \mathbf{H}_i^\top \mathbf{x} - r'_i\|^2 \quad (13)$$

where  $\alpha_i = (\beta_i^2 - \beta_i \chi_i)^{\frac{1}{2}}$ ,  $r'_i = b_i \left(1 - \frac{\chi_i}{\beta_i}\right)^{-\frac{1}{2}}$ ,  $\beta_i = \frac{e^{-\chi_i^2}}{\sqrt{2\pi} \mathbf{Q}(\chi_i)}$

Stacking all measurements, the linear least-squares problem Eq. (13) essentially assumes the following linear residual:

$$\mathbf{r}' = \mathbf{H}' \tilde{\mathbf{x}} + \mathbf{n}_r \quad (14)$$

where  $\mathbf{r}' = [\dots r'_i \dots]^\top$  and  $\mathbf{H}' = [\dots \alpha_i \mathbf{H}_i \dots]^\top$ ,  $\mathbf{n}_r$  is the new measurement noise. Since the quantized measurement-based posterior pdfs will still be close to bell-shape Gaussian pdfs (see [18]), we assume the estimation result of previous and current timestamps still follows Gaussian distribution. Thus, we can again adopt the standard EKF equations to recursively and efficiently update the estimate and covariance:

$$\hat{\mathbf{x}}^\oplus = \hat{\mathbf{x}}^\ominus \boxplus \mathbf{P}^\ominus \mathbf{H}'^\top \mathbf{S}^{-1} \mathbf{r}' \quad (15)$$

$$\mathbf{P}^\oplus = \mathbf{P}^\ominus - \mathbf{P}^\ominus \mathbf{H}'^\top \mathbf{S}^{-1} \mathbf{H}' \mathbf{P}^\ominus \quad (16)$$

where  $\mathbf{S} = \mathbf{H}' \mathbf{P}^\ominus \mathbf{H}'^\top + \mathbf{I}$  is the residual covariance.

#### IV. QUANTIZED-VIO ALGORITHMS

Within the preceding quantization-based estimation framework, in this section, we develop the first-of-its-kind quantized VIO (QVIO) estimators with the two different quantizers, whose architectures are illustrated in Figure 1. In the proposed visual-inertial system, we assume there is an

on-sensor co-processor (e.g., FPGA, or vision silicon) that can perform image processing and transmit/receive quantized data to/from the host computer where the main computations of visual-inertial estimation are carried out. We now explain how the proposed EKF-like visual-inertial estimators integrate the measurement/residual quantizers.

In particular, the state vector  $\mathbf{x}$  at time  $t_k$  consists of the current navigation states  $\mathbf{x}_{I_k}$ , and a set of historical IMU pose clones  $\mathbf{x}_C$  (see [23]):

$$\mathbf{x} = [\mathbf{x}_{I_k}^\top \mathbf{x}_C^\top]^\top, \quad \mathbf{x}_C = [\mathbf{x}_{T_k}^\top \dots \mathbf{x}_{T_{k-c}}^\top]^\top \quad (17)$$

$$\mathbf{x}_{I_k} = [{}^I_G \bar{\mathbf{q}}^\top \quad {}^G \mathbf{p}_{I_k}^\top \quad {}^G \mathbf{v}_{I_k}^\top \quad \mathbf{b}_g^\top \quad \mathbf{b}_a^\top]^\top \quad (18)$$

where  ${}^I_G \bar{\mathbf{q}}$  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  $\{I\}$ ;  ${}^G \mathbf{p}_{I_k}$ ,  ${}^G \mathbf{v}_{I_k}$  are the IMU position, velocity;  $\mathbf{b}_g$  and  $\mathbf{b}_a$  are the gyroscope and accelerometer biases;  $\mathbf{x}_{T_i} = [{}^I_G \bar{\mathbf{q}}^\top \quad {}^G \mathbf{p}_{I_i}^\top]^\top$ . As in the standard MSCKF-based VIO [23], [24], we propagate the state over time based on the following nonlinear IMU kinematics:

$$\mathbf{x}_{I_{k+1}} = \mathbf{f}_I(\mathbf{x}_{I_k}, {}^I \mathbf{a}_k, {}^I \boldsymbol{\omega}_k, \mathbf{n}_I) \quad (19)$$

where  $\mathbf{n}_I = [\mathbf{n}_g^\top \mathbf{n}_a^\top \mathbf{n}_{wg}^\top \mathbf{n}_{wa}^\top]^\top$ ,  $\mathbf{n}_g^\top$  and  $\mathbf{n}_a^\top$  are the noises,  $\mathbf{n}_{wg}^\top$  and  $\mathbf{n}_{wa}^\top$  are the random walk bias noises of the gyroscope and accelerometer respectively.

We now focus on quantization for measurement update. Assume the camera measures a feature  $\mathbf{f}$  at timestamp  $t_k$ , where  $\mathbf{f}$  denotes its 3D position in the global frame. The corresponding bearing measurement is given by:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k \simeq \mathbf{H}_x^* \tilde{\mathbf{x}}_{I_k} + \mathbf{H}_f^* \tilde{\mathbf{f}} + \mathbf{n}_k \quad (20)$$

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. Typically, it is easy to detect and track many environmental features, and transmitting and processing all the pixel measurements can be power-consuming and incur significant latency. This motivates us to quantize the raw measurements.

### A. zQVIO: Update with Quantized Measurements

As explained in Section III-A, the proposed measurement quantizer directly quantizes the raw pixel measurements using a given number of bits, which leads to the following linearized measurement model:

$$\mathbf{r}_k^m := \mathbf{B}(\mathbf{z}_k) - \mathbf{h}(\hat{\mathbf{x}}) \simeq \mathbf{H}_x^* \tilde{\mathbf{x}}_{I_k} + \mathbf{H}_f^* \tilde{\mathbf{f}} + \mathbf{n}_{z_k} \quad (21)$$

where  $\mathbf{B}(\mathbf{z}_k)$  is the multi-bit quantized measurement [see Eq. (5)],  $\mathbf{n}_{z_k}$  is the noise. Note that the number of bits is determined via trial and error or in an ad-hoc manner, which is not optimal. Once we have these quantized measurements, naively we can directly perform EKF updates with them, which however cannot compensate for the introduced quantization errors that could be significant in practice.

Therefore, to better incorporate the incurred errors due to the use of quantized measurements  $\mathbf{B}(\mathbf{z})$  instead of raw ones  $\mathbf{z}$ , we seek to properly model this quantization error in the proposed zQVIO. The key idea is to use a Gaussian distribution to approximate their difference (i.e.,  $\mathbf{n}_q = \mathbf{B}(\mathbf{z}) - \mathbf{z} := [\dots n_{q_i} \dots]$ ). Considering the scalar measurement  $z$ , the quantization error can be modeled as a uniform distribution with an interval  $[-r_q/2, r_q/2]$ , where  $r_q$  denotes the quantization resolution (e.g., for quantizing to range 0 to 1 with  $n$  bits,  $r_q = 1/2^n$ ). We can leverage a Gaussian distribution to approximate it by solving Kullback–Leibler divergence (KLD) and result as  $n'_q \sim \mathcal{N}(0, r_q^2/12)$ , which has zero mean and the same variance as the original uniform distribution. As such, for each scalar measurement, the quantization error can be modeled as an extra noise in addition to the raw measurement noise  $n_z$ , and the new measurement noise  $n'_z$  can thus be formulated as:  $n'_z = n_z + n'_q$  where  $n_z \sim \mathcal{N}(0, \sigma^2)$ , and thus  $n'_z \sim \mathcal{N}(0, r_q^2/12 + \sigma^2)$ .

It is important to note that this measurement quantizer entails a simple and clean communication protocol between the co-processor and host computer, as shown in Figure 1 (top), which could be appealing for modular design in practice. Specifically, the initial preprocessing of visual and IMU data remains unchanged. In the subsequent steps, only the quantized measurements are sent to the host processor, where both feature triangulation and estimation are performed with these quantized measurements. To account for the quantization errors, the approximate Gaussian noise is used to inflate the raw measurement noise.

### B. rQVIO: Update with Quantized Residuals

We now quantize the following residuals:

$$\mathbf{r}'_k = \mathbf{H}'_x \tilde{\mathbf{x}}_{I_k} + \mathbf{H}'_f \tilde{\mathbf{f}} + \mathbf{n}_{r_k} \quad (22)$$

where  $\mathbf{r}'_k$  is the quantized residual. Note that the new measurement Jacobians  $\mathbf{H}_x$  and  $\mathbf{H}_f$ , and quantized residual  $\mathbf{r}_k$  are computed based on Eqs. (13) and (14) given 1-bit residual quantization  $\mathbf{b}(\mathbf{z})$ . We then perform the MSCKF update [23], by first projecting Eq. (22) onto the left nullspace  $\mathbf{N}$  of the feature Jacobian  $\mathbf{H}_f$  and then recursively updating the state estimate and covariance [see Eqs. (15) and (16)].

In the following, we explain in detail the three main steps of this residual-quantized update (Figure 1 (bottom)).

1) *Data Preprocessing*: In this step, camera measurements and IMU readings are preprocessed. The co-processor, upon receiving images from the camera, performs feature detection and tracking to obtain bearing measurements from

these images. The timestamp corresponding to the camera measurement is sent to the host processor. In the meantime, the host processor collects inertial readings to propagate the state forward to the camera timestamp [see Eq. (19)]. Subsequently, the most recent IMU pose from this propagation is transferred back to the co-processor. Note that there are two reasons to communicate poses: (i) We need poses to calculate the measurement residuals, and (ii) the host processor cannot triangulate features with only 1-bit measurements.

2) *Data Transmission*: During this step, the tracked features are triangulated to get their initial guesses. The initial guesses and measurements are then communicated to the host processor to perform an MSCKF update [see Eq. (22)]. For the bearing measurements, both residuals for  $u$  and  $v$  are quantized to a single-bit measurement  $b(u)$  and  $b(v)$ . The initial feature guesses  ${}^C\mathbf{p}_f$  are also quantized to reduce the communication cost.

3) *Quantized Update*: In the final step, in the host processor, the quantized residuals from the co-processor are used for quantized estimation. The state correction  $\delta\mathbf{x}$  for the IMU poses is sent back to the co-processor and thus the co-processor has the updated IMU poses, which will help the feature triangulation.

In summary, the main steps of the proposed QVIO are outlined in Algorithm 1:

---

#### Algorithm 1 Quantized-VIO (QVIO)

---

##### Data Preprocessing:

- Co-Processor: Receives new image and timestamp from the image sensor. Perform feature detection and tracking, send measurement timestamp to the host processor
- Host Processor: Receives IMU readings for state propagation [see Eq. (19)], sends the latest pose to the Co-Processor

##### If do Residual Quantization (rQVIO):

- **Data Transmission:**
  - Co-Processor: Perform feature triangulation, send the quantized one-bit raw measurement and the quantized feature initial value to the host processor
- **Quantized Update:**
  - Host Processor: Perform quantized-based state estimation [see Eq. (20)]. Send the correction to the Co-Processor
  - Co-Processor: update the state estimates

##### If do Measurement Quantization (zQVIO):

- Co-Processor: Send quantized multi-bit measurements to the host processor
  - Host Processor: Perform general state estimation with the quantized measurement [see Eq. (21)]
- 

## V. NUMERICAL STUDY

We employ the OpenVINS [24] to produce realistic bearings and inertial measurements and implement the proposed QVIO estimators. In what follows, rQVIO denotes the residual quantization approach [See Section IV-B], zQVIO denotes the measurement quantization method [See Section IV-A], while zQVIO-D means that directly taking the measurement quantization approach without including the quantization error (using  $n_z$  instead of  $n'_z$  in Section IV-A). The number in the bracket for zQVIO (e.g., zQVIO(9)) denotes the number of bits used to quantize the raw measurements. In each timestamp, 200 bearing measurements are generated. Regarding the estimator configuration, we maintain a window size of 15. All features are MSCKF

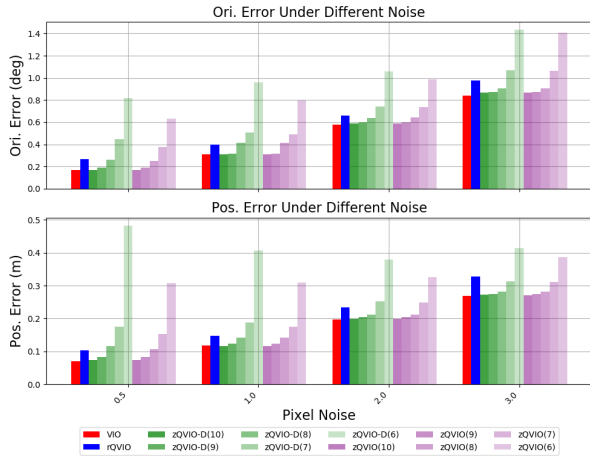


Fig. 2: Comparison of estimation errors with varying noise levels. VIO (red) is normal VIO, rQVIO (blue) denotes VIO using quantized residuals, zQVIO-D (green) and zQVIO (purple) denote using quantized measurements without and with accounting quantization error, respectively.

features and will be used to do updates when they lose tracking or reach the maximum window size.

One key communication overhead for the quantized residual method is the requirement of communicating feature linearization points for each set of measurements (i.e., the need to compute measurement Jacobian when performing estimation in the host processor [See Eq. (22)]). Therefore, we experiment with different numbers of bits used to communicate feature bearing and depth (i.e., the  $\mathbf{q}(\mathbf{f})$  in Figure 1 (bottom) contains its bearing and depth). For the bearing, we normalize its range to 0 to 1 by projecting it onto the image pixel coordinate and dividing it by the image size. A fixed-point number representation was then employed. Depth quantization was approached differently with a customized floating-point representation:

$$\text{Number} = s \times b \times 10^a \quad (23)$$

where  $s$  is a predetermined scaling factor,  $b$  is the fractional part, ranging from  $0 \sim 10 - \frac{10}{2^m}$ ,  $m$  is the bits assigned for the significant digit, while  $a$  is the exponential coefficient ranging from  $0 \sim 2^{n-1} - 1$ ,  $n$  is the bits assigned for the exponential component. We assign 0.01 for the scaling factor, 2 bits for the exponential component, and the rest for the significant digit. After transmission, all the transmitted data are converted back to the canonical floating point for estimator update.

We studied how many bits are needed for clear communication of features, as shown in Figure 3 (right). From the result, we can see when the number of bits for the depth is smaller than 9, the system shows a clear accuracy drop. Using less than 10 bits for bearing communication, the accuracy drop will continue until using over 9 bits for depth communication. Based on these, our final design incorporates 10 bits for bearing communication and 10 bits for depth.

Another major component in the communication for the rQVIO is the need to obtain state updates from the host processor [see  $\mathbf{q}(\delta\mathbf{x})$  in Figure 1 (bottom)]. For both the update of position and rotation, for each DoF, we allocate 1 bit for the exponential segment and another for the sign, the

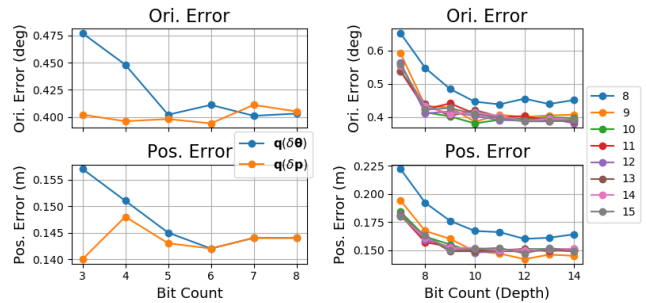


Fig. 3: The numerical study of the impact of accuracy with the number of bits used to quantize state update (left) and feature initial guess (right) (50 runs). For the right figures, the x-axis is the bit count for depth, and different curves correspond to different bit counts for the bearing.

scaling factor is set to be 0.001. The remaining bit is used to represent significant digits. A similar numerical study is performed, and from Figure 3 (left), we observe that it is feasible to use only 3 bits to communicate 1 DoF of position update (9 bits for one 3D position), and only 6 bits for 1 DoF of rotation update (18 bits for one 3D rotation) without a significant loss of accuracy.

To study the performance of zQVIO, we vary the number of bits used to quantize raw measurements and compare their performance with no quantization (VIO) and rQVIO. For the residual quantization method, the communication setup is shown in Table II. It is expected that as the simulated camera view is  $752 \times 480$ , thus using less than 10 bits for pixel measurements will result in a truncation of the original pixel readings, which will degrade system performance. As shown in Figure 2, the accuracy drops for measurement quantization are more severe as the number of bits allocated decreases (i.e., there is a significant accuracy drop when the number of bits reaches 6). Also, when the measurement noise is small, the accuracy drop of quantization is more significant, likely due to quantization noise being dominant. As expected, the residual quantization method always shows an accuracy drop due to the loss of information (i.e., only 1 bit is used). When the measurement noise is small, the performance of rQVIO is close to zQVIO using 8 bits, while when the noise is larger than 1, its performance is close to zQVIO using 7 bits, as the increase of pixel noise offsets the quantization effect. Comparing zQVIO and zQVIO-D when the same number of bits is used, zQVIO always outperforms zQVIO-D as it better captures the quantization error. The improvement of zQVIO is more significant when the pixel noise or the number of bits used is small, as in such cases the quantization error is more significant compared with the image pixel noise.

## VI. REAL-WORLD EXPERIMENTS

We further evaluate the two proposed quantized estimators on the EuRoC MAV dataset [25] using only the left camera. Our system, built on OpenVINS [24] as a baseline, extracted 300 sparse point features and managed 15 clones, processing them with MSCKF features when they lost track or exceeded the window size. For rQVIO, the communication setup is the same as the one we use in simulation, as shown in Table II.

One of the challenges in the real world is outlier rejection, which is typically addressed by the  $\chi^2$  test. However,

TABLE I: Average Absolute Trajectory Error (ATE) in degrees/meters and runtime in microseconds (ms) across 10 runs.

Algo.	V101	V102	V103	V201	V202	V203	MH01	MH02	MH03	MH04	MH05	Runtime
OpenVINS	0.53 / 0.07	2.10 / 0.06	1.68 / 0.08	0.88 / 0.08	1.24 / 0.07	1.09 / 0.21	2.18 / 0.17	0.90 / 0.20	1.18 / 0.17	1.36 / 0.27	0.70 / 0.37	3.5
rQVIO	0.71 / 0.07	1.90 / 0.06	1.97 / 0.08	1.60 / 0.14	1.25 / 0.08	1.37 / 0.21	2.84 / 0.21	1.02 / 0.20	1.24 / 0.19	1.01 / 0.19	0.74 / 0.32	2.8
zQVIO(10)	0.54 / 0.07	2.10 / 0.06	1.68 / 0.08	0.94 / 0.07	1.23 / 0.07	0.98 / 0.20	2.24 / 0.18	0.90 / 0.19	1.17 / 0.17	1.36 / 0.28	0.70 / 0.36	3.5
zQVIO(9)	0.51 / 0.07	2.06 / 0.06	1.72 / 0.08	0.82 / 0.09	1.24 / 0.08	0.93 / 0.20	2.20 / 0.17	0.90 / 0.20	1.34 / 0.18	1.42 / 0.29	0.74 / 0.39	3.5
zQVIO(8)	0.61 / 0.08	2.14 / 0.06	1.66 / 0.08	0.93 / 0.09	1.34 / 0.09	1.05 / 0.19	2.90 / 0.26	0.87 / 0.20	1.34 / 0.21	1.22 / 0.27	0.79 / 0.38	3.5
zQVIO(7)	0.58 / 0.11	2.10 / 0.07	1.68 / 0.08	1.23 / 0.09	1.43 / 0.11	1.11 / 0.17	2.62 / 0.24	1.02 / 0.18	1.65 / 0.25	1.45 / 0.28	0.80 / 0.42	3.6
zQVIO(6)	1.04 / 0.13	2.01 / 0.09	2.06 / 0.13	1.40 / 0.11	1.51 / 0.14	1.35 / 0.23	3.23 / 0.31	1.29 / 0.23	1.74 / 0.33	1.39 / 0.38	0.91 / 0.51	3.7
RVIO2	0.88 / 0.09	2.27 / 0.10	2.02 / 0.10	2.19 / 0.13	1.90 / 0.11	1.50 / 0.15	2.60 / 0.17	1.00 / 0.15	1.08 / 0.19	1.10 / 0.24	0.95 / 0.32	1.8
VINS-Mono	0.82 / 0.07	2.74 / 0.10	5.15 / 0.15	2.13 / 0.09	2.57 / 0.13	3.43 / 0.29	0.78 / 0.20	0.86 / 0.18	1.82 / 0.23	2.51 / 0.41	0.94 / 0.29	22.4

in rQVIO, covariance is not available at the coprocessor end. We thus perform a naive test by ignoring the pose uncertainty, the  $\chi^2$  threshold is inflated by 2 and reject the measurements with residuals larger than 3. For the zQVIO, we vary the number of bits from 6 to 10. We only report the zQVIO with quantization error modeling since it shows superior performance in the simulation, and moreover, without taking care of those error, the system is unable to run when the number of bits is below 8 from our experience.

We also compare with the open-sourced RVIO2 [26], which is a square-root inverse filter VIO based on robot-centric state formulation and MSCKF-based feature processing, and VINS-Mono [27], which is an optimization-based sliding window VIO system [28]. The averaged Absolute Trajectory Error (ATE) values are reported in Table I. It is obvious to see that rQVIO and zQVIO with numbers for bits larger than or equal to 8 all achieve comparable performance with its baseline OpenVINS and achieve similar or better performance compared with RVIO2 or VINS-Mono. The minimal accuracy drop for zQVIO is as expected, which is also shown in the previous simulation sections. Surprisingly, the accuracy drop of rQVIO is not as significant in the real world as in simulation. Our speculation about two potential reasons behind this: (a) A different outlier rejection method is used for rQVIO, thus the measurements it uses can be quite different from others. (b) The loss of information is also applied to outliers, thus it mitigates the impact of outliers. Similarly, for certain cases zQVIO with only 7-bit achieves better accuracy. This might also because the inflated noise mitigates the influence of outliers.

An average communication statistics for the zQVIO on the V101 sequence can be shown in Table III. It is impressive that the average communication per measurement is only 3.7 bits while showing a minimal accuracy drop while quantizing raw measurements shows a significant accuracy degradation when the number of bits allowed is below 7.

Table I shows the runtime of the estimators. Ideally, quantization would minimally affect efficiency given it does not alter state or measurement sizes, and the extra computation complexities are at least an order smaller than the estimator complexity. As anticipated, zQVIO experiences a negligible overhead when the bit count is below 8, possibly stemming from extra time dedicated to 3D feature triangulation refinement with quantized bearing measurements. Notably, rQVIO is shown to be even more efficient than VIO, attributed to a simplified  $\chi^2$  test and a supplementary reprojection error check that may reject more measurements.

#### A. Remarks

First of all, the simulation and real-world experiment results have clearly validated the proposed first-of-its-kind

TABLE II: Communication setup for rQVIO. The bit size is the requirement for communicating 1 DoF of specific data.

	$C_{p_f}$ bearing	$C_{p_f}$ depth	$\delta p$	$\delta \theta$	$p, R$
Data Type	fixed	float	float	float	float
Bit Size	10	10	3	5	16
Min	0	0	-0.05	-0.0875	-65504
Max	1	99.6904	0.05	0.0875	65504

TABLE III: Average communication statistics for rQVIO for all the sequences in EuRoC MAV Dataset. The communication includes propagated latest pose, state updates, feature initial guesses, and 1-bit measurements. (Timestamp and specific communication package protocol are ignored.)

Avg. Feats/frame	Avg. meas/frame	Avg. bits/frame	Avg. bit/meas
10.8	283.9	1063.2	3.7

quantized VIO. Comparing the two different quantizers, the zQVIO is simple and intuitive, requiring only one-way communication, which could be compelling in practice in terms of modular design and scalable maintenance. Despite its ad-hoc quantization, the proposed zQVIO is able to achieve minimal accuracy drop with only 8 bits per measurement. However, the accuracy drop is significant when less than 7 bits are available. In contrast, the proposed rQVIO is able to achieve comparable performance in the real world, but with an average of only 3.7 bits per measurement. This is especially appealing for edge computing, while its communication protocol is more complicated and might be less robust.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have, for the first time ever, designed the quantized visual-inertial odometry (QVIO) algorithms to reduce data transfer to improve energy efficiency. In particular, we proposed to quantize raw visual measurements with only a few bits and then perform EKF update with these quantized measurements, and moreover, in order to improve this ad-hoc measurement quantizer, we have also developed the rQVIO that instead quantizes the measurement residuals for update. Through extensive simulations and real-world experiments, we have shown that using only 8 bits for measurement quantization can still preserve good accuracy, while under the scenario data bandwidth is extremely limited, residual quantization is highly recommended as it can achieve comparable performance with only an average of 3.7 bits per measurement. In the future, we are interested in investigating multi-bit and iterative optimization for the residual quantization approach. Additionally, specific communication package protocols are ignored, we also plan to incorporate the analysis of their impact for future work.

## REFERENCES

- [1] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.
- [2] C. Chen, Y. Yang, P. Geneva, W. Lee, and G. Huang, "Visual-inertial-aided online mav system identification," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyoto, Japan., 2022.
- [3] S. Katragadda, W. Lee, Y. Peng, P. Geneva, C. Chen, C. Guo, M. Li, and G. Huang, "Nerf-vins: A real-time neural radiance field map-based visual-inertial navigation system," in *Proc. International Conference on Robotics and Automation*, Yokohama, Japan, May 2024.
- [4] "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/>.
- [5] 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>
- [6] C. Liu, A. Berkovich, S. Chen, H. Reyserhove, S. S. Sarwar, and T.-H. Tsai, "Intelligent vision systems—bringing human-machine interface to ar/vr," in *2019 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2019, pp. 10–5.
- [7] F. Zhou and Y. Chai, "Near-sensor and in-sensor computing," *Nature Electronics*, vol. 3, no. 11, pp. 664–671, 2020.
- [8] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "Soi-kf: Distributed kalman filtering with low-cost communications using the sign of innovations," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4782–4795, 2006.
- [9] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, "A set of level 3 basic linear algebra subprograms," *ACM Transactions on Mathematical Software (TOMS)*, vol. 16, no. 1, pp. 1–17, 1990.
- [18] N. Trawny, S. I. Roumeliotis, and G. B. Giannakis, "Cooperative multi-robot localization under communication constraints," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 4394–4400.
- [19] E. D. Nerurkar, K. X. Zhou, and S. I. Roumeliotis, "A hybrid estimation framework for cooperative localization under communication constraints," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 502–509.
- [20] E. D. Nerurkar and S. I. Roumeliotis, "A communication-bandwidth-aware hybrid estimation framework for multi-robot cooperative localization," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1418–1425.
- [21] —, "Hybrid maximum a posteriori estimation under communication constraints," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 4898–4902.
- [22] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, Jan. 2013.
- [23] 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*, 2007.
- [24] P. Geneva, K. Eickenhoff, 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](https://github.com/rpng/open_vins)
- [25] 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.
- [26] Z. Huai and G. Huang, "Square-root robocentric visual-inertial odometry with online spatiotemporal calibration," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9961–9968, 2022.
- [27] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [28] C. Chen, P. Geneva, Y. Peng, W. Lee, and G. Huang, "Optimization-based vins: Consistency, marginalization, and feJ," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.