Visual-Laser-Inertial SLAM
Using a Compact 3D Scanner for Confined Space

Daqian Cheng[1], Haowen Shi[1], Albert Xu[2], Michael Schwerin[1], Michelle Crivella[3], Lu Li[1] and Howie Choset[1]

***Abstract*— Three-dimensional reconstruction in confined spaces is important for the manufacturing of aircraft wings, inspection of narrow pipes, examination of turbine blades, etc. It is also challenging because confined spaces tend to lack a positioning infrastructure, and conventional sensors often cannot detect objects in close range. Therefore, such tasks require a sensor that is compact, operates in short-range, and able to localize itself. In this paper, we introduce a miniature and low-cost 3D scanning system including an active laser-stripe triangulation hardware, integrated inertial sensors, and a Simultaneous Localization and Mapping (SLAM) software tailored for the sensor. The proposed system is capable of reconstructing photo-realistic 3D point cloud in real-time in spite of its compact monocular configuration. To achieve this capability, we propose an approach to capture both color and geometry using alternating shutter-speed on a single camera. A novel SLAM method is proposed to accurately localize the sensor by fusing laser, camera, and inertial measurements. Evaluation of localization accuracy and comparison on reconstruction performance against a significantly larger commercial off-the-shelf sensor demonstrate the proposed system's advantages in real-world applications.**

I. INTRODUCTION

Three-dimensional reconstruction is a fundamental problem in robotics and computer vision. Various sensor systems with wide-ranging capabilities (e.g. range and resolution), such as laser-stripe triangulators, RGB-D cameras, and LiDARs, and corresponding algorithms [1], [2], [3], [4] have made accurate 3D scanning possible in many types of spaces (e.g. indoor [5], [6], [7], outdoor [8], [4], [9], underwater [10], [11], [12], etc), revolutionizing both civil and industrial fields. These sensor hardware and software systems, in the authors' view, operate in wide-open spaces and are not well-suited, by design, for confined space operation. In fact, few, if any, systems for infrastructure-free 3D reconstruction have been developed for confined space operation. Such systems would be of great use for inspection applications, such as turbine inspection, where a unit is usually disassembled just to perform the inspection. The challenge in building such a sensor comes from the confined space constraints: the sensor must be 1) compact to fit into tight spaces; 2) able to operate at short-range; 3) able to localize itself without positioning infrastructure.

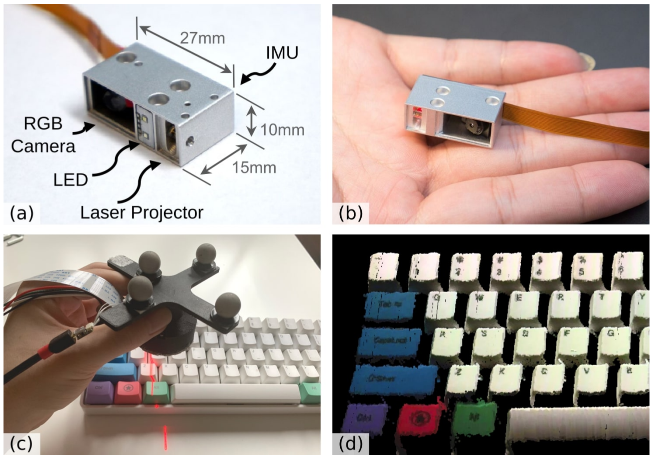

Fig. 1. (a), (b) The proposed sensor hardware prototype; (c) hand-held scanning with ground truthing experimental set up; (d) the reconstructed colored point cloud of a keyboard, scanned without external infrastructures.

Current commercial off-the-shelf (COTS) sensors for 3D reconstruction are either too large or dependent on external positioning infrastructure (e.g. robotic manipulators, motion-capture cameras, etc). Kinect (Microsoft, Redmond, WA, USA) and RealSense (Intel, Santa Clara, CA, USA) are two popular RGB-D cameras with self-localization capability, but even the smallest model size is $90 \times 25 \times 25$ mm with a minimum sensing range of 105 mm. Highly accurate and compact laser profilers such as optoNCDT (Micro-Epsilon, Raleigh, NC, USA) require external positioning devices to function. [13] introduced an ultra-compact 3D measurement sensor but also lacked self-localization capability.

In this paper we propose a compact and low-cost 3D scanning system including a hardware design and a tailored SLAM framework for self-localization. Based on active laser-stripe triangulation, the sensor consists of a monocular color camera, a laser stripe projector, and an Inertial Measurement Unit (IMU). The proposed sensor achieves a size of $27 \times 15 \times 10$ mm and a sensing range of 20-150 mm. Fig. 1 shows the proposed sensor hardware as well as the hand-held colored point cloud reconstruction of a keyboard.

Localization accuracy often determines reconstruction quality since individual laser scans are registered using localization. Monocular visual-inertial (VI) sensor setup is considered the smallest sensor-suite to perform SLAM with metric scale, and VI-SLAM methods have achieved promising results and are nowadays widely used in mobile robots, smartphone applications, and VR & AR. However, sensor

This work was funded by the Boeing Strategic University Program

[1]The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

[2]Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

[3]Boeing Research & Technology, North Charleston, SC 29456, USA

motion in confined spaces is often much slower and IMU measurements are much less excited, resulting in poor metric scale estimation and localization accuracy. Therefore, we proposed a SLAM method designed for active laser-stripe triangulators by fusing visual, laser, and inertial measurements. In [14] we briefly introduced the sensor design and high-level frameworks but the SLAM method was not described; in this paper, we improved our previous work and introduced a new window-to-map tracking method which enables consistent mapping under multi-pass scans. Experiments show higher localization accuracy of the proposed SLAM method compared to a state-of-the-art VI-SLAM method, demonstrate the SLAM framework's ability to maintain mapping consistency under repeated re-scanning, and show the proposed sensor's superior reconstruction quality to a COTS RGB-D camera. We summarize our main contributions as follows:

- A design of a compact and low-cost 3D scanner.
- An alternating-shutter approach to achieve colored 3D reconstruction using a monocular sensor.
- A novel visual-laser-inertial SLAM (VLI-SLAM) framework with 1) laser-based metric scale estimation and 2) window-to-map tracking for consistent mapping.

II. Related Works

Active laser-stripe triangulation has been one of the mainstream 3D scanning approaches for decades [15]. Usually consisted of a camera and a laser-stripe projector, the sensor detects the laser stripe and triangulates it into 3D space. Thanks to the simple hardware design and inexpensive components, laser-stripe triangulation is a popular choice for low-cost 3D scanning systems such as the DAVID Laserscanner [16]. Many high accuracy profilers such as Keyence Laser Profiler (Keyence Corporation, Osaka, Japan) and metallic surface scanners [17] also adopt laser-stripe triangulation due to its high accuracy and relative insensitivity to illumination compared to structured light. Although extensive work has been dedicated to reconstructing 3D models [1], very few have focused on localization using laser-stripe scanners to enable infrastructure-free capability, and positioning devices or localization aids [10] are often needed to register individual scans.

Structured light and time-of-flight are two core technologies behind today's 3D scanners. Structured light scanners project 2D patterns of light onto the scanned surface [18], [19], [20], while time-of-flight sensors measure distance using the travel time of light signals. Realsense and Kinect are two popular and relatively low-cost 3D scanners and are commonly referred to as RGB-D cameras since each pixel provides color and depth. A number of RGB-D SLAM algorithms with promising results have emerged in the past decade, including volumetric [2], [6] and surfel-based [7], [21] methods. However, compared to laser-stripe scanners, an RGB-D point cloud frame is generally able to account for 6 degree-of-freedom (DoF) motion via point cloud alignment. Therefore, these RGB-D SLAM methods cannot be directly applied to laser-stripe triangulators.

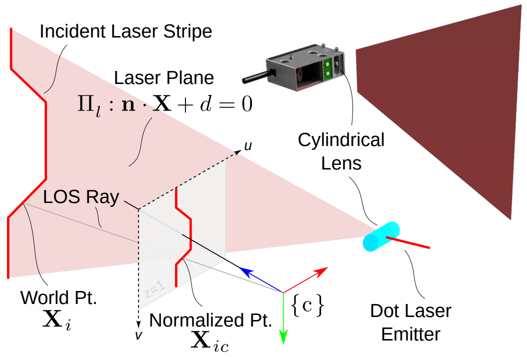

Fig. 2. Theory of operation: Laser depth is triangulated by projecting a camera ray out from the camera origin and finding its intersection with the laser plane.

The proposed SLAM method designed for laser-stripe triangulators is most similar to monocular visual SLAM, which has two main approaches: the feature-based approach [22], [23] uses visual features of images to estimate camera motion, while the direct approach [24], [25] directly utilizes pixel intensities. However, monocular SLAM is only able to recover the up-to-scale structure (camera motion and map). To overcome this scale ambiguity, an Inertial Measurement Unit (IMU) is often incorporated to recover the metric scale [23], [26]. Since laser triangulation is able to estimate the metric scale more accurately than an IMU, our SLAM method fuses visual features, inertial, and laser depth measurements to achieve high localization accuracy.

III. Sensor System Design

A. Hardware & Sensor Model

The proposed scanner hardware consists of an RGB CMOS camera, a MEMS-based 6-axis accelerometer & gyroscope, and a laser-stripe projector. A single laser stripe pattern is created by refracting a thin laser beam through a cylindrical lens, and is projected to the region in side camera field of view. The red laser stripe can be toggled on/off in synchronization with our image shutter trigger to enable the alternating-shutter technique described in Sec. III-B.

3D points on the laser stripe are recovered from 2D images using triangulation. We model the projected laser stripe as a plane $\Pi_l : \mathbf{n}\cdot\mathbf{X}+d=0$ in 3D space, which intersects with the physical world. Depth of each image pixel observation $\mathbf{x}_i$ of the laser stripe is estimated using triangulation by solving a ray-plane intersection problem illustrated in Fig. 2 and described in (1), where $\mathbf{X}_i^c$ denotes the triangulated 3D point and π_c^{-1} denotes the back projection function that projects a pixel position onto the normalized image plane.

$$\mathbf{X}_i = \frac{-d}{\mathbf{n}\cdot\pi_c^{-1}(\mathbf{x}_i)}\pi_c^{-1}(\mathbf{x}_i) \tag{1}$$

We refer readers to our previous work [14] for the calibration of geometric parameters π_c^{-1} and Π_l.

B. Software Framework

Fig. 3 shows the software framework for processing sensor data, localizing the sensor, and reconstructing a photorealistic 3D point cloud.

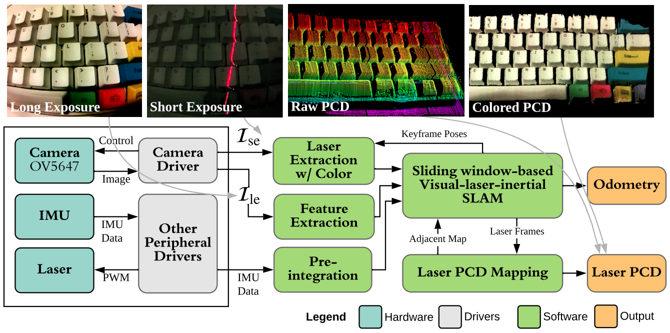

Fig. 3. Software framework and data flow visualization.

A highlight of our software is a custom designed sensor driver, which enables measuring two unique types of information using a single camera sensor by alternating the shutter between long and short exposure times. Thus, both bright images $\mathcal{I}_{\text{le}}$ for camera motion estimation and dark images $\mathcal{I}_{\text{se}}$ for laser depth triangulation can be captured at adjacent sample frames. Additionally, the laser stripe projector is synchronized with the camera shutter to switch off for $\mathcal{I}_{\text{le}}$ and on for $\mathcal{I}_{\text{se}}$. See Fig. 4 for a diagram detailing the interleaving timing sequence. The purpose of this approach is to allow the monocular camera to capture both color and geometric information with minimal time gap, in order to reduce sensor physical size that is critical for confined space requirements. Optimized for 3D geometry acquisition, $\mathcal{I}_{\text{se}}$'s are underexposed and exhibit a high laser-to-background contrast; $\mathcal{I}_{\text{le}}$'s are neutrally exposed, with no laser stripe, and are utilized for SLAM and point cloud coloring.

IV. VISUAL-LASER-INERTIAL SLAM

The proposed SLAM method fuses visual feature measurements, depth measurements from laser scan, and inertial measurements to achieve high localization accuracy. Each sensor plays a different role: visual features serve as the main source of camera motion estimation; IMU helps handle abrupt motion and estimate orientation thanks to its observability of roll and pitch angles; the laser points provide the metric scale for the visual odometry and help maintain mapping consistency via point cloud alignment. The proposed SLAM framework can be broken down to the following components: 1) a front-end that pre-process raw sensor data into visual features, colored laser points, and pre-integrated inertial data; 2) an initialization process bootstraps the optimization problem structure, 3) an odometry component estimates camera motion, 4) a mapping module registers laser points into a point-based map representation, and 5) a window-to-map tracking component aligns current measurements to the map to correct odometry drift.

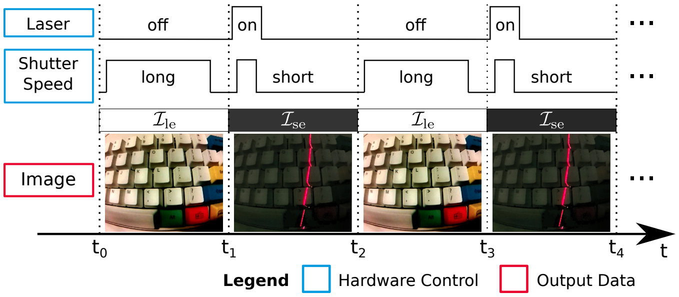

Fig. 4. The camera shutter alternates between long and short exposure ”on” times, keeping frame duration constant for all frames. The laser's state toggles synchronously with the alternating exposure, producing a sequence of images that provide both RGB and depth information.

A. Front End

Visual: 1) Visual features are extracted and tracked in each $\mathcal{I}_{\text{le}}$ image using KLT optical flow [27]: existing features in the previous frame are tracked and new feature points are extracted to maintain a minimum number of features. 2) We define *features-on-laser* $\mathcal{F}_l$ as the subset of feature points $\mathcal{F}$ close to the laser scan; for these features, the laser point cloud can help accurately estimate feature depths. A feature f_i is defined to be a feature-on-laser if any of its observations is close to the laser stripe pixels in adjacent $\mathcal{I}_{\text{se}}$'s, and the observation frame with feature's pixel position being the closest to the laser stripe is defined as f_i's primary observation frame $c^*_{f_i}$. For a feature $f_i \notin \mathcal{F}_l$, its $c^*_{f_i}$ is the first observation frame. 3) Keyframe selection: an $\mathcal{I}_{\text{le}}$ frame becomes a keyframe if the average feature parallax from the previous keyframe is sufficiently large or the number of tracked features from the previous keyframe is too small.

Laser: For each $\mathcal{I}_{\text{se}}$, we detect the laser stripe pixels using the center-of-mass method [28] and triangulate these pixels into 3D points as described in Sec. III-A. Color information for each laser point is retrieved via projective data association using several temporally adjacent keyframes. Given the pose of this $\mathcal{I}_{\text{se}}$ (interpolated as described in Sec. IV-B) and the keyframes, each laser point is transformed into the global frame, then reprojected onto adjacent keyframes to find the average color of the associated pixels.

IMU: Preintegration is a commonly used technique to handle inertial integration efficiently by avoiding repeated computation. We perform preintegration following [29], [23].

B. Initialization Process

We initialize the sliding window-based SLAM framework by generating initial estimates of keyframe poses and feature depths in the sliding window using the following procedures. 1) First find two keyframes in the sliding window with enough parallax, such that the first frame is the primary observation frame of several features-on-laser. 2) The up-to-scale transformation between the two frames is estimated using the eight-point algorithm [30] with an arbitrary scale s_0. 3) Depth $\hat{d}$ of all the common feature points are estimated by triangulation. 4) The correct scale $\hat{s}$ is then estimated using each feature-on-laser's closest laser pixel's depth $\bar{d}$: $\hat{s} = (\sum_i^K \bar{d}_i/\hat{d}_i)/K \cdot s_0$. The two keyframes' poses and feature depths are then corrected using $\hat{s}$. 5) Given the initialized structure of the two keyframes, poses of other keyframes in the sliding window are estimated using the perspective-n-point algorithm [31], and other feature points in the sliding window are triangulated. 6) Finally, a bundle adjustment optimizes all camera poses and feature depths

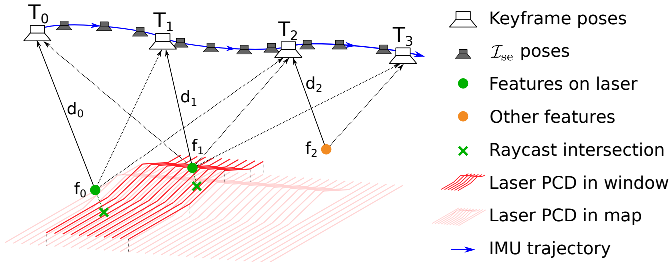

Fig. 5. Illustration of the sliding window-based visual-laser-inertial SLAM. The sliding window is consisted of several keyframe poses, features observed by the keyframes, laser point cloud observed in the time span of the sliding window, adjacent laser point cloud in previously built map (if revisited), and inertial measurements.

in this sliding window, and poses of $\mathcal{I}_{\text{se}}$'s are interpolated between poses of adjacent keyframes in order to register individual laser scans into a global point cloud.

Given an initialized camera motion trajectory and pre-calibrated extrinsic transformation between camera and IMU, we initialize the inertial-related variables including biases, velocity and gravity using methods described in [23].

C. Sliding Window-Based SLAM Formulation

We propose a tightly-coupled *visual-laser-inertial SLAM* (VLI-SLAM) formulation in a sliding-window of keyframes. Nonlinear optimization is employed to solve for state variables $\mathcal{X}$ consisting of keyframe poses $\mathbf{T}$, IMU states (linear velocity and biases) and inverse feature depths λ in each feature's primary observation frame. A combination of four types of residuals are minimized in the optimization problem: visual feature depth residual given laser point cloud, visual feature reprojection residual, inertial measurement residual, and window-to-map tracking residual (described in Sec. IV-D). An illustration of the proposed SLAM formulation is shown in Fig. 5.

Features-on-Laser Depth Residual: Depths of $\mathcal{F}_l$ can be accurately estimated using the depth prior from the registered laser point cloud. The depth prior $\bar{d}_i$ of a feature-on-laser $f_i \in \mathcal{F}_l$ is computed by first finding the 3D laser points near the feature viewing ray from $c^*_{f_i}$ using projective data association [32], then fitting a 3D plane to these points and intersecting the plane with the feature viewing ray to find $\bar{d}_i$. Using these depth priors $\bar{d}$, we introduce a residual for $\mathcal{F}_l$ described in (2).

$$r_l(\mathcal{X}) = \sum_{f_i \in \mathcal{F}_l} \left\| \frac{1}{\lambda_i} - \bar{d}_i \right\|^2 \tag{2}$$

Feature Reprojection Residual: For each feature $f_i \in \mathcal{F}$, reprojection residuals defined in (3) are evaluated between the primary frame $c^*_{f_i}$ and every other observation frame in the sliding window $\mathcal{C}$. In (3), $\mathbf{x}_i^j$ denotes the pixel observation of the ith feature in the jth keyframe; $\pi_c(\cdot)$ denotes camera projection function and $\pi_c^{-1}(\cdot)$ denotes back projection function; $\mathbf{T} \in \text{SE}(3)$ denotes a transformation matrix.

$$r_c(\mathcal{X}) = \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{C}} \left\| \pi_c \left(\mathbf{T}_w^{c_j} \mathbf{T}_{c^*_{f_i}}^w \frac{1}{\lambda_i} \pi_c^{-1} \left(\mathbf{x}_i^* \right) \right) - \mathbf{x}_i^j \right\|^2 \tag{3}$$

Inertial Measurement Residual: We follow the IMU measurement residual definition in [29], [23] to help estimate linear velocity, IMU biases, and camera poses; details are not elaborated for brevity. Since the laser point cloud provides metric scale information, IMU is not necessary for the scanner to function but is still desirable for directly observing roll and pitch angles and being able to handle abrupt motion.

D. Mapping & Window-to-Map Tracking

We use a point-based map representation similar to [21], [7], where each map point contains the following attributes: a position $\mathbf{v} \in \mathbb{R}^3$, a normal $\mathbf{n} \in \mathbb{R}^3$, an RGB color $\mathbf{c} \in \mathbb{R}^3$, and a weight $w \in \mathbb{R}$. Laser point cloud frames are added to the map after popped out of the sliding window. For each laser point to add, if there exist a nearby map point $\mathbf{p}$ with compatible color and normal, then the new point is merged into $\mathbf{p}$; if not, the new point is added to the map and its normal is estimated using nearest neighbors algorithm [33]. The weight attribute is the number of times that a map point is merged with a new point.

Accumulation of odometry drift will violate mapping consistency when user revisit a scanned region to fill reconstruction holes or to obtain a denser point cloud [20]. To account for this issue, many RGB-D SLAM methods have adopted a frame-to-map tracking approach instead of a frame-to-frame one [2], [21], [7]. However, laser points in a single frame are co-planar and geometrically insufficient to account for 6 DoF motion. Therefore, we propose a window-to-map tracking approach, where the registered laser point cloud in the sliding window is aligned to the map. Since odometry drift exists within the sliding window, a nonrigid Iterative Closest Point problem is formulated where laser points from the same $\mathcal{I}_{\text{se}}$ are treated as rigid, but transformation between $\mathcal{I}_{\text{se}}$'s are treated as nonrigid. This is achieved by incorporating per-point point-to-plane residual defined in (4) into the SLAM formulation. In (4), $\mathbf{v}_i$ is a laser point from an $\mathcal{I}_{\text{se}}$ in the sliding window, and c_k and c_{k+1} are the two temporally adjacent keyframes; $f(\cdot)$ denotes a pose interpolation function to estimate the $\mathcal{I}_{\text{se}}$ pose using its timestamp; $\mathbf{v}_i^g$, $\mathbf{n}_i^g$, and w_i are attributes of the closest map point to $\mathbf{v}_i$, which is searched for using KD-Tree.

$$r_{icp} = \sum_i w_i \left\| \left(\mathbf{v}_i^g - f\left(\mathbf{T}_{c_k}^w, \mathbf{T}_{c_{k+1}}^w, t_i \right) \mathbf{v}_i \right) \cdot \mathbf{n}_i^g \right\|^2 \tag{4}$$

V. EXPERIMENTS

The sensor's performance in hand-held 3D scanning is evaluated with real-world scanning experiments. Targeting localization and mapping benchmarking as main objectives, we first evaluated the localization accuracy of the proposed VLI-SLAM in Sec. IV against VINS-Mono, a state-of-the-art visual-inertial SLAM method, using the same sensor hardware [23], followed by a comparison of colored point cloud reconstruction against a popular COTS RGB-D camera, Intel RealSense D435. We also showcase the scanning of several industrial and household objects in Fig. 8.

The experiments were conducted by hand-holding the sensor to scan a keyboard. To mimic 3D scanning in confined spaces, the sensor was held at ∼3 cm above the keyboard facing downward, and the camera motion was kept slow to decrease the IMU signal-to-noise ratio (SNR). Because the laser stripe only covered three rows of keys at a time, we scanned the keyboard using a back-and-forth zigzag motion pattern consisting of six passes to incrementally cover the scene, visualized in Fig. 6. The total trajectory length was 185.4 cm and the average speed was 1.40 cm/s. Fig. 1c shows the experiment setup, where the sensor was mounted on a 3D-printed handle attached with motion capture markers for localization ground truth.

The sensor outputs $\mathcal{I}_{\text{se}}$ and $\mathcal{I}_{\text{le}}$ images of VGA resolution at 60 frames per second (FPS) combined and inertial measurements (linear acceleration and angular velocity) at 200 FPS. To achieve real-time SLAM, we used a sliding window size of 8 keyframes and extracted 100 visual features from each $\mathcal{I}_{\text{le}}$. On the testing PC with AMD Ryzen 3700x CPU, the average computation time was 29.8 milliseconds per frame.

A. Odometry Accuracy Evaluation

We evaluated the proposed VLI-SLAM method against VINS-Mono. A visual-inertial SLAM method is chosen as benchmark because it is the best choice available given the sensor suite. To evaluate the window-to-map tracking component described in Sec. IV-D, we experimented with two versions of the proposed SLAM method: one denoted as VLI-Odom with the window-to-map tracking component turned off, and the other as VLI-SLAM with it turned on. The ground truth trajectory was obtained using the Vicon motion capture system (Vicon Industries, Hauppauge, NY, USA).

Fig. 6 shows the estimated trajectories in the top-down view of VINS-Mono, VLI-Odom, VLI-SLAM against ground truth, with absolute translational and rotational errors analysis. In the error plots, the background colors divide the time period into six segments corresponding to the six passes in the zigzag trajectory, starting from the bottom-left. The performance statistics comparison are listed in Table I, where drift is defined as maximum error over trajectory length.

Based on this experiment, VINS-Mono showed significantly larger translational drift compared to both VLI-Odom and VLI-SLAM, mainly due to inaccurate scale estimation, which was caused by high measurement noise of the low-cost MEMS IMU and low signal-to-noise ratio from the slow sensor motion. VLI-SLAM demonstrated slightly better translational accuracy than VLI-Odom thanks to the window-to-map tracking component, which reduced drift by registering current measurements with historic information. Since this drift-correction is map-centric rather than localization-centric, the mapping benefited more as described in Sec. V-B. All three methods showed similar rotation estimation performance.

TABLE I

ABSOLUTE LOCALIZATION ERRORS AND DRIFT RATES

Error metric	VINS-Mono	VLI-Odom	VLI-SLAM
t RMSE (cm)	5.1	0.39	**0.32**
t Max (cm)	8.6	0.86	**0.60**
t Drift (%)	4.6	0.46	**0.32**
r RMSE (rad)	0.022	**0.014**	0.023
r Max (rad)	**0.030**	0.033	0.035
r Drift (10^{-4} rad/cm)	**1.6**	1.8	1.9
t Drift$^{\text{Abs}}$ (%)	23.7	**0.65**	N/A
r Drift$^{\text{Abs}}$ (10^{-4} rad/cm)	4.3	**3.1**	N/A

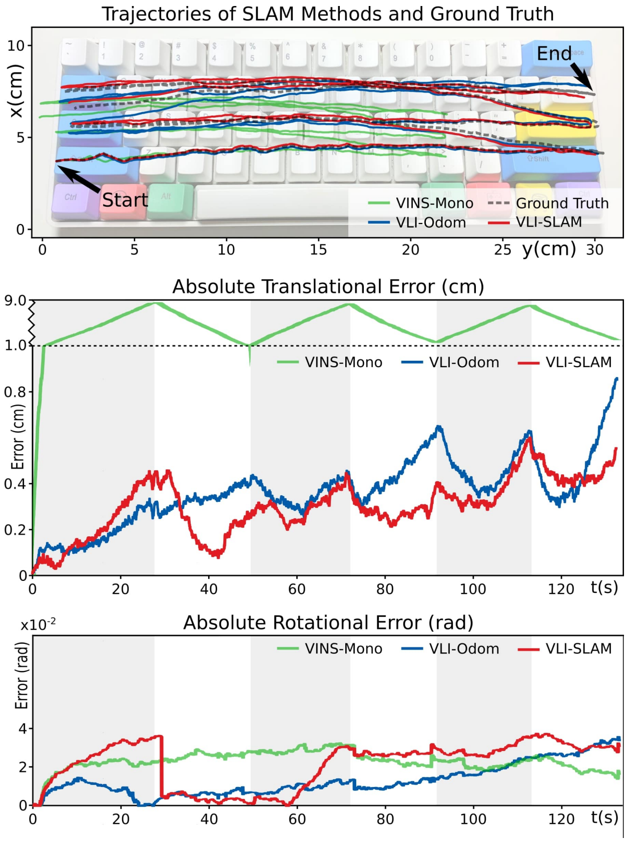

Fig. 6. Trajectories in top-down view of the proposed SLAM methods, VINS-Mono and ground truth and the associated translational and rotational errors. The background color of error plots indicates different passes in the zigzag trajectory. The top portion in the translational error plot is rescaled to accomodate the large error of VINS-Mono.

One key performance metric for real-world 3D scanning is the absolute drift. However, in this experiment, the drift growth often alternated between positive and negative as the sensor motion changed direction, thus the average drift is smaller than the absolute drift. Therefore, we segmented the trajectory into six passes. Within each pass, drift is zeroed at the beginning to evaluate the absolute drift, which is then averaged across the six passes. These translational and rotational drifts are denoted as Drift$^{\text{Abs}}$ in Table I. Since the window-to-map tracking would register the later passes to the first one, VLI-SLAM is not evaluated for absolute drift.

B. 3D Reconstruction Evaluation

The proposed sensor was compared against Intel RealSense D435 since it is one of the smallest low-cost and infrastructure-free 3D scanner although still significantly larger than the proposed sensor. RTAB-Map [3] was employed for SLAM using D435.

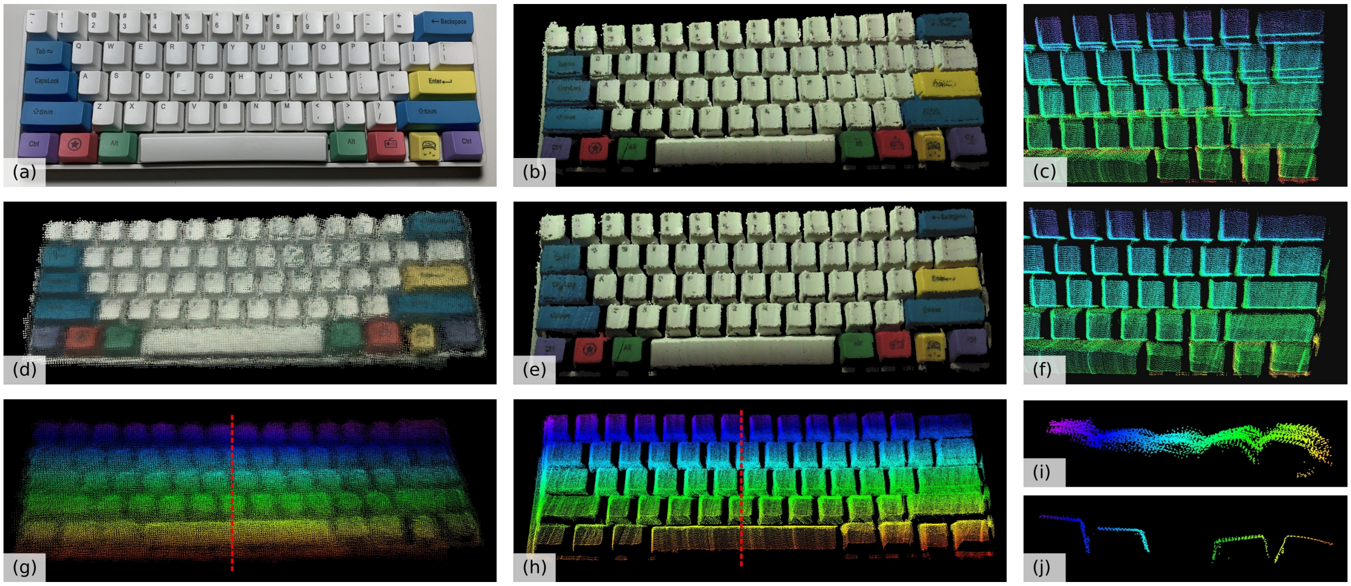

Fig. 7. Comparison of point cloud reconstruction. (a) is a photograph of the scanned scene. (d) and (g) are the photo-realistically and spatially colored reconstruction results by RealSense. Reconstructed using the proposed sensor, (b) and (c) show results using VLI-Odom, and (e), (f) and (h) are with VLI-SLAM. (i) and (j) are the sectional views of (g) and (h) respectively with the red dashed lines as the cutting planes.

The reconstructed point clouds were geometrically evaluated using a ground truth point cloud, which we obtained using a UR5e robot manipulator (Universal Robots, Odense, Danmark) to scan the keyboard with the proposed scanner. The point-to-point and point-to-plane RMSEs are shown in Table II, where the proposed sensor with VLI-SLAM showed the smallest error.

To qualitatively compare both the reconstructed color texture and geometrical shape, we present the photo-realistic colored point clouds as well as spatially color-coded point clouds in Fig. 7.

Based on the results, the proposed sensor system was able to achieve 3D reconstruction results with finer texture details as well as sharper geometries: comparing Fig. 7 (d) and (e), our sensor delivered superior reconstruction details on letters and patterns of the keycaps; geometric structures were also sharper in (h) compared to (g) which is more evidently shown in the sectional views (i) and (j). This confirmed the claim that laser-stripe profilers are often able to achieve higher reconstruction accuracy than structured-light based RGB-D cameras. The window-to-map tracking component in VLI-SLAM significantly improved mapping consistency under back-and-forth scanning motions. Fig. 7 (c) and (f) show the partial point cloud of (b) and (e) respectively in spatial color-coding. In (c) the point clouds from different passes were clearly separated from each other due to SLAM drift, and in (f) the point clouds were tightly aligned. We observe that although the window-to-map tracking only slightly reduces localization error in Sec. V-A, the mapping quality was drastically improved. This demonstrated the proposed VLI-SLAM's ability to maintain mapping consistency under back-and-forth coverage scanning, which is a common motion pattern for both laser-stripe profilers and other 3D scanners.

TABLE II

MAPPING RMSE STATISTICS

Error metric	VLI-Odom	VLI-SLAM	RealSense
Point-to-point (mm)	1.2	**0.97**	2.3
Point-to-plane (mm)	0.93	**0.76**	2.0

To comprehensively demonstrate the system's scanning capability, we also include the hand-held reconstructions of several other objects in Fig. 8.

VI. CONCLUSIONS

In this paper, a miniature 3D scanner for confined spaces with close sensing range and infrastructure-free self-localization is introduced. A framework including alternating shutter data generation and a visual-laser-inertial SLAM method are designed to achieve photo-realistic 3D reconstruction using the monocular sensor. This framework can be generalized to any camera-based laser triangulators. Experimental evaluation on localization demonstrated our SLAM method's performance compared to a state-of-the-art visual-inertial SLAM method, and the reconstruction results suggest our sensor is able to capture finer details and sharper geometric shapes against a popular but larger COTS RGB-D camera. To the best of the authors knowledge, the proposed sensor and software framework is the most compact RGB-D photo-realistic reconstruction system for hand-held infrastructure-free 3D reconstruction, which provides a disruptive solution for a wide range of 3D scanning applications where sensor form factor and ultra short sensing range are critical.

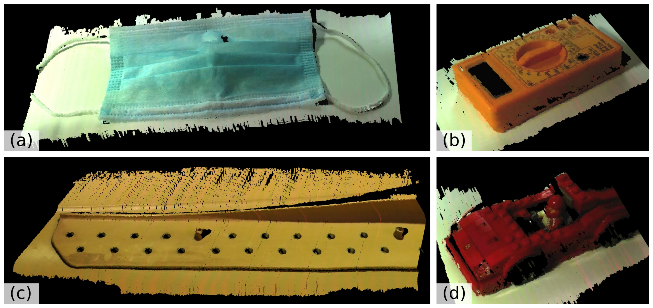

Fig. 8. Reconstruction using the proposed system of (a) a face mask, (b) a multimeter, (c) a industrial aerospace part, and (d) a toy car.

REFERENCES

[1] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 127–136.

[3] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[4] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[5] M. Klingensmith, I. Dryanovski, S. S. Srinivasa, and J. Xiao, "Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields." in *Robotics: science and systems*, vol. 4. Citeseer, 2015, p. 1.

[6] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[7] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[8] F. Dai, A. Rashidi, I. Brilakis, and P. Vela, "Comparison of image-based and time-of-flight-based technologies for three-dimensional reconstruction of infrastructure," *Journal of construction engineering and management*, vol. 139, no. 1, pp. 69–79, 2013.

[9] Y. Ling and S. Shen, "Real-time dense mapping for online processing and navigation," *Journal of Field Robotics*, vol. 36, no. 5, pp. 1004–1036, 2019.

[10] M. Massot-Campos, G. Oliver, A. Bodenmann, and B. Thornton, "Submap bathymetric slam using structured light in underwater environments," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, 2016, pp. 181–188.

[11] A. Palomer, P. Ridao, and D. Ribas, "Inspection of an underwater structure using point-cloud slam with an auv and a laser scanner," *Journal of Field Robotics*, vol. 36, no. 8, pp. 1333–1344, 2019.

[12] M. Massot-Campos, G. Oliver-Codina, and B. Thornton, "Laser stripe bathymetry using particle filter slam," in *OCEANS 2019 - Marseille*, 2019, pp. 1–7.

[13] S. Katayose, Y. Kurata, K. Watanabe, R. Kasahara, M. Itoh, D. Watanabe, K. Matsuo, and K. Hanano, "Ultra-compact 3d measurement module using silica-based plc," in *2019 24th Microoptics Conference (MOC)*, 2019, pp. 84–85.

[14] D. Cheng, H. Shi, M. Schwerin, L. Li, and H. Choset, "A compact and infrastructure-free confined space sensor for 3d scanning and slam," in *2020 IEEE SENSORS*. IEEE, 2020.

[15] G. Sansoni, M. Trebeschi, and F. Docchio, "State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation," *Sensors*, vol. 9, no. 1, pp. 568–601, 2009.

[16] S. Winkelbach, S. Molkenstruck, and F. M. Wahl, "Low-cost laser range scanner and fast surface registration approach," in *Joint Pattern Recognition Symposium*. Springer, 2006, pp. 718–728.

[17] P. Fasogbon, L. Duvieubourg, and L. Macaire, "Fast laser stripe extraction for 3d metallic object measurement," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 923–927.

[18] J. Geng, "Structured-light 3d surface imaging: a tutorial," *Advances in Optics and Photonics*, vol. 3, no. 2, pp. 128–160, 2011.

[19] S. Zhang, "High-speed 3d shape measurement with structured light methods: A review," *Optics and Lasers in Engineering*, vol. 106, pp. 119–131, 2018.

[20] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, "State of the art on 3d reconstruction with rgb-d cameras," in *Computer graphics forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 625–652.

[21] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3d reconstruction in dynamic scenes using point-based fusion," in *2013 International Conference on 3D Vision-3DV 2013*. IEEE, 2013, pp. 1–8.

[22] 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 multi-map slam," *arXiv preprint arXiv:2007.11898*, 2020.

[23] 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.

[24] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[25] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[26] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[27] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[28] R. Fisher and D. Naidu, "A comparison of algorithms for subpixel peak detection," in *Image technology*. Springer, 1996, pp. 385–404.

[29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[30] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[31] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[32] G. Blais and M. D. Levine, "Registering multiview range data to create 3d computer objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 820–824, 1995.

[33] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, "Comparison of surface normal estimation methods for range sensing applications," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3206–3211.