Zebedee: Design of a Spring-Mounted 3-D Range Sensor with Application to Mobile Mapping

Michael Bosse, Robert Zlot, and Paul Flick.

Abstract—Three-dimensional perception is a key technology for many robotics applications including obstacle detection, mapping, and localization. There exist a number of sensors and techniques for acquiring 3D data, many of which have particular utility for various robotic tasks. We introduce a new design for a 3D sensor system, constructed from a 2D range scanner coupled with a passive linkage mechanism, such as a spring. By mounting the other end of the passive linkage mechanism to a moving body, disturbances resulting from accelerations and vibrations of the body propel the 2D scanner in an irregular fashion, thereby extending the device’s field of view outside of its standard scanning plane. The proposed 3D sensor system is advantageous due to its mechanical simplicity, mobility, low weight, and operation in unstructured environments and interact safely and effectively with humans, other vehicles, and their environment. 3D data acquisition is also of vital importance for traditionally non-robotic (but in some cases automatable) tasks such as surveying, object scanning, mobile mapping, and gaming. There are a number of ways to acquire 3D data, such as scanning lidar or radar, flash lidar, stereo vision, imaging sonar, and structured light triangulation. The variety of solutions that are in common use is clear evidence that no single solution is suitable for all applications. The existing technologies differ in many ways, but the characteristics of primary relevance for robotics are typically accuracy, range, resolution, field of view, operating conditions, cost, weight, power consumption, and complexity. Aiming to achieve a good balance of these properties, we propose a new design concept for a 3D sensor system in which a 2D range scanner is coupled to a passive linkage mechanism, such as a spring (Figures 1 and 2). By mounting the device on a moving platform, such as an autonomous vehicle, accelerations and vibrations of that platform are transferred by the reactive linkage mechanism to motion of the 2D scanner with respect to the platform. The induced motion has the effect of rotating the 2D scanner’s scan plane, thereby increasing the field of view of the device with respect to the environment to achieve 3D coverage.

Allowing a range sensor to move about relatively freely and non-deterministically is counter to the traditional practice in robotics and mobile mapping. Typically, sensors are mounted rigidly to the moving platform, and motion is assumed to follow the trajectory of the platform (offset by a known and static rigid transformation). In some cases, suspensions or gimbals are used to stabilize the sensor mount and remove the effects of high frequency accelerations due to rough terrain, engine vibration, or other unwanted disturbances [1, 2]. In contrast, our design amplifies the effects of such accelerations and exploits the resulting motion to increase the sensor field of view along an additional scanning dimension.

Though our proposed design generally encompasses any type of range sensor, in practice we focus on 2D time-of-flight lidars. Two dimensional laser scanners are ubiquitous in the field of robotics: models that provide reasonable range, resolution, and accuracy in a small, lightweight package and operate in a wide range of environmental conditions are commercially available at modest costs. Thus, given that the...
passive linkage mechanism can be as simple as a spring, the hardware cost of the proposed device is essentially the price of a 2D laser scanner (plus, for some applications, an inertial measurement unit).

There are two common methods of 3D range sensing in mobile robotics that are closely related to our approach. Firstly, lasers can be actuated in a variety of ways, typically by periodic nodding [3] or continuous rotation [4, 5]. The resulting configurations can potentially have large fields of view, but are relatively mechanically complex. More recently, a more complex design which controls the rotation rate to increase the point density in an intended gaze direction has been proposed [6]. Secondly, a mechanically simpler solution is to rigidly attach one or more 2D lasers to a platform and rely on motion of the platform to produce 3D data, as is done in most commercial mobile mapping solutions [7]. Mobile mapping systems with fixed lasers require very precise knowledge of the vehicle trajectory, which is typically derived from high-grade GPS/INS solutions. When such high-grade solutions are not available—either due to their high cost or required operation in GPS-deprived environments—a wide field of view sensing system would be required to allow for correction online [5]. The proposed spring system sits between the above two configurations: it provides a mechanically simple solution with a large field of view, but requires platform motion in order to be maximally functional.

The mechanical simplicity, size, and weight of this sensor design make it suitable for developing practical solutions to a range of challenging applications, such as mapping from a micro-UAV, or infrastructure inspection in confined spaces. A lightweight handheld or human-wearable version of a spring-mounted laser would be ideal for use as a localization or mapping tool by first responders in emergency situations. While several handheld 3D scanners are commercially available (e.g., Z Corporation’s ZScanner, Leica’s T-Scan, and Mantis Vision’s MVC-F5), they are primarily intended for object scanning applications, often require modification to the environment, and have limited working volume not appropriate for large-scale mobile applications. Available structured lighting solutions are limited in their sensing range, precision, and the lighting conditions in which they can operate. For example, the Microsoft Kinect has an effective range of a few meters, precision of around 12 cm at 4 m range (the range resolution behaves approximately as $r^2 \times 0.0075$ m$^{-1}$), and fails in bright sunlight. Stereo cameras have similar precision characteristics and their performance is dependent on lighting conditions and textural appearance of the environment.

The contributions of this work are in the sensor system design, analysis of its behavior in a variety of configurations, and demonstration of its use for indoor and outdoor mapping. Novel enhancements enable an open-loop Simultaneous Localization and Mapping (SLAM) algorithm intended for actuated lasers to be adapted for the irregular motions of the proposed sensor design. A further contribution is in the application of the same SLAM algorithm to globally register a 3D point cloud and estimate a closed-loop trajectory.

The remainder of this article is organized as follows. In Section II, we describe a particular implementation of the proposed sensor system consisting of a 2D laser scanner and an IMU mounted on one or more springs (Figures 1 and 2), and analyze a number of different configurations of that general design. An example application of the sensor system is presented in Section III. Here, a handheld version of the sensor is utilized for SLAM in a number of different environments. Experiments demonstrating the SLAM algorithm for the handheld system are presented in Section IV for a number of environments and settings, including evaluation in a motion capture environment. A summary and discussion of the results, limitations, lessons learned, and future directions conclude the article in Section V.

II. ZEBEDEE: DESIGN AND ANALYSIS

The Zebdeee 3D sensor system consists of a 2D laser scanner and an inertial measurement unit (IMU) mounted on one or more springs (Figures 1 and 2). The laser in our current model is a Hokuyo UTM-30LX, which is a 2D time-of-flight laser with a 270° field of view, 30 m maximum range, and 40 Hz scanning rate. The dimensions of the UTM-30LX are 60×60×85 mm, and its mass is 210 g, which makes it ideal for low weight requirements. The IMU we use is a MicroStrain 3DM-GX2, an industrial-grade IMU that contains triaxial MEMS gyros and accelerometers with an output rate...
of 100 Hz. Due to the expected magnitude of the spring oscillations, the IMU is required to have a maximum rotational rate of at least 600 °/s, which is an available but non-standard option for the 3DM-GX2. The IMU enclosure has dimensions of $41 \times 63 \times 32$ mm, and its total mass is 50 g. Springs with lengths between 50 and 150 mm and masses between 5 and 20 g are typically utilized to realize mechanical requirements. Some fine tuning is possible by adjusting the length of the spring and/or adding mass to the sensor head. The laser and IMU are mounted within a 150 g 3D-printed housing (Figure 2a left), giving the system a total mass of well under half a kilogram.

We define the sensor head to be the portion of the device on the range sensor side of the passive linkage mechanism, and the sensor base to be the rigid body containing the mount point on the platform side of the linkage. For instance, in the case of Zebedee, the sensor head contains the laser and IMU, and the sensor base may be a vehicle or handle on which the spring is mounted.

A. Spring Characterization

When considering the suitability of a particular spring, it is important to understand the behavior of the resulting device under a range of operating conditions. In order to empirically characterize the performance of a particular choice of spring, we measure the system response of the sensor head due to motions of the sensor base. In particular, we analyze the transfer of motions through the system using the rotational velocity of the base as input and the rotational velocity of the head as output. For these experiments, a second IMU is mounted to the sensor base in order to measure its motion while the system is excited over a range of frequencies. We consider two spring configurations for Zebedee: a short single-spring design for handheld applications; and a dual-spring design for a vehicle mount. The dual-spring design restricts the rotations so that the sensor head does not flip over with strong accelerations. The handheld sensor (Figure 2) is excited manually, while the dual-spring sensor is mounted on a Gator TE electric vehicle (Figure 2b). When the spring is in its neutral position, the orientations of the base and head frames are roughly aligned.

Manual excitation of the handheld unit provides roughly 10 Hz of input bandwidth. To achieve higher input bandwidths for analysis, we could use a vibration table or mechanical robot arm commanded on a pseudo-random trajectory, though typical handheld motions rarely exceed frequencies of 5 Hz. From the observed data, a 16th-order state space system is identified using an Observer Kalman Filter Identification technique [8]. The frequency response of the identified $3 \times 3$ MIMO system characterizes the resonant frequencies, amplification gains, and cross-coupling of the input to output rotational rates (Figure 3). In general, the springs tend to amplify the lower frequencies while suppressing higher frequencies. For the handheld Zebedee with a single spring, we have identified the dominant resonant frequency for pitch and roll rates at 1.4 Hz, and a secondary peak at 6.9 Hz for yaw rates. The vehicle-mounted Zebedee with dual springs has three different resonant frequencies at 1.6 Hz for the pitch rates, 2.25 Hz for yaw rates and 5.4 Hz for roll rates. The 5.4 Hz roll resonance is a residual vibration (note the relative magnitude of the gain) which does not contribute significantly to the sensor sweeping motion. The gains are about an order of magnitude greater than the handheld system, which is a result of the longer springs and is necessary to amplify the smaller input rotations present on a vehicle. The observed resonant frequency values are similar to actuation frequencies commonly used for nodding and spinning laser systems. Typical amplitudes of the oscillations result in a spread of approximately 100° for the handheld sensor and 70° for the vehicle mounted sensor, which provides a more than adequate field of view for many 3D perception applications.

Simple heuristic adjustments can be used to fine-tune the behavior of a particular configuration. For example, lengthening the spring—or more precisely, increasing the distance of the sensor head center of gravity with respect to the sensor base—decreases the resonant frequencies. Adding mass to the sensor head or decreasing the spring constant (e.g., increasing the pitch or decreasing the wire diameter) increases the gain.

1 On newer generations of Zebedee hardware, we use a MicroStrain 3DM-GX3-25-OEM, which has a mass of 11.5 g.
Timing accuracy, synchronization, and calibration has been recognized as a key factor in precision mapping and perception systems. The two main sensors comprising Zebedee are by default not synchronized, and due to timing jitter and non-constant latencies, the timestamps of the data arriving from the sensors cannot be naively accepted as accurate. The error in the projected position of a laser point is proportional to the rotational velocity, its range, and the timing latency error. Typical use of Zebedee induces rotational rates of around 600°/s, which implies that at a range of 10 m every millisecond of error in latency would result in approximately 10 cm of error in the laser point position. The sources of the timing errors include variable data buffering and/or dropped scans due to high system loads; motor speed adjustments internal to the Hokuyo laser; resolution of the system and Hokuyo clock (1 ms); temperature; and, to a lesser extent, gyroscopic effects on the rotating mirror in the laser when shaken vigorously.

As an example of the importance of timing accuracy, Figure 4 shows two series of point clouds from a mapping application. The first row shows the effect of errors due to the timing offset between the laser and IMU. The center point cloud uses an estimated timing correction calculated by the mapping solution, and to either side the point clouds have had extra latency errors applied to this correction. With the introduction of latency errors, the relative positions of surfaces measured at different sensor head rotational velocities are misaligned, even though the local surface structure is maintained. The second row of the figure shows the effects of increasing the timing jitter. Unlike the latency errors, timing jitter results in unrecoverable blurring of the local structure that increases with magnitude of the sensor head rotational velocity.

Based on these observations, we conclude that for most applications, some means of time synchronization and smoothing must be built into the solution. Much of the timing jitter effects can be smoothed with knowledge of both the sensor internal and host system clock; however, timing latencies are not observable from the time sequences alone. We address the issue of latency estimation in our SLAM solution in Section III-C.

III. 3D SLAM with Zebedee

A simple, lightweight, 3D range sensor such as Zebedee is ideal for many mapping and localization applications. Useful accelerations, vibrations, and oscillations are naturally present in small rotary wing UAVs, off-road wheeled vehicles, and all-terrain legged robots, such as RHEx and LittleDog. A handheld or wearable sensor performing SLAM could also be useful in first responder applications, indoor mapping, and inspection of infrastructure or natural environment within confined spaces. We have developed an online SLAM solution in which we are able to accurately estimate the six degree of freedom (6DoF) motion of the sensor head and generate a 3D point cloud as Zebedee is transported through the environment. While we can apply this approach to produce globally registered trajectories and maps, we first focus on an open-loop incremental solution analogous to scan-matching.

The Zebedee SLAM algorithm is adapted from an algorithm previously developed for a continuously spinning 2D laser on a moving platform. Similar to the case of a spinning laser, over the typical time it takes for Zebedee to sweep over the scene, significant platform motion may occur and the resulting 3D scan cannot be treated as a rigid snapshot from a single location. However, the irregular and nondeterministic motion of Zebedee—in contrast with the controlled and repetitive nature of a spinning laser—introduces some significant new challenges into this SLAM problem.

At a high level, the SLAM algorithm estimates the 6DoF trajectory $T(\tau)$ consisting of a sequence of translations $t(\tau)$.
and rotation \( r(\tau) \) of the 2D sensor frame relative to the world frame as a function of time \( \tau \). For this view-based SLAM formulation, the trajectory is sufficient to describe the full state and is used to project the raw laser measurements into a registered 3D point cloud when necessary. As new data are acquired, the algorithm proceeds by processing a time-windowed segment of the trajectory, then advancing the window by a fraction of its length from the previous time step. Each segment is processed by solving a linearized system to determine the smooth trajectory that best explains corresponding surfaces in the associated 3D point cloud, subject to boundary conditions that ensure continuity with the previous segment.

The processing of a segment of data is an iterative process following a similar framework to the well-established Iterative Closest Point (ICP) algorithm commonly used for point cloud registration. A key difference of this approach from the standard ICP formulation is that instead of solving for a single rigid transformation, the solution estimates a continuous trajectory. Similar to ICP, there are two main processing steps that are iterated until convergence. The first step identifies corresponding surface patches from the laser point cloud, and the second step updates the trajectory to minimize errors between matching surfaces and deviations from the measured IMU accelerations and rotational velocities. The optimized state also includes parameters for estimating time synchronization and IMU biases, which may change slowly over time.

On the first iteration, the previously unprocessed trajectory segment is initialized by integrating the accelerometer and gyro measurements from the IMU. Since the processing window is advanced by a fraction of its length, the first section of the trajectory segment will have already been estimated from the previous time step, thus the IMU data are only required to propagate the trajectory for the remainder of the window.

**A. Correspondence Step**

Surface elements, or *surfels*, are extracted from the 3D point cloud projected from the initial trajectory estimate. Clusters of laser points that are both spatially and temporally proximal are identified and used to compute surface properties based on the centralized second-order moment matrix of the point coordinates. Provided there are enough points in the cluster, the surface normal is obtained from the eigenvector corresponding to the minimum eigenvalue of the second-order moment matrix. An estimate of the surface planarity, computed from the ratio of the eigenvalues, is used to discard surfels that are not approximately planar since the estimate of their surface normals would be unreliable. The timestamp associated with the surfel is computed from the mean of the timestamps of all the points in the cluster. Clusters with a small (less than \( \sim 15 \degree /s \)) rotational velocity component normal to the scan plane are discarded since the unknown translational velocities dominate the perceived local shape. When the normal component of the rotational velocity is high, then the initially unknown translational effect on the local surface geometry is minimized. However, for very fast motions, the surfels are automatically discarded due to undersampling (i.e., too few points per cluster).

The laser point clusters are determined by spatially decomposing the scene into a multi-resolutional voxel grid (similar to the strategy used in our previous work [5]). The temporally proximal points contained within each cube-shaped cell define each cluster from which a surfel is generated. There may be multiple clusters of points within the same spatial cell if the point timestamps span an interval larger than a predetermined threshold. The voxel resolutions are selected to double at each level, thus the cluster surface properties at coarser resolutions can be computed recursively from the finer resolutions. In order to reduce the sensitivity of the segmentation to the cell boundaries, the surfels at each level are computed from two grids offset by half of a cell width[^4].

Corresponding surfels are obtained from approximate \( k \)-nearest neighbor queries of a \( kd \)-tree in the six-dimensional space of surfel positions and normals. Matches are filtered by retaining only reciprocal surfel matches (i.e., the two matched surfels are both identified as near neighbors of one another). In contrast to our previous work [5], correspondences can arise from any two surfels provided the difference between their timestamps is larger than half of the nominal sweep period. These conditions help focus computational resources on quality matches over a larger time span, which provides more useful constraints to the trajectory. Each matched pair generates a match error to be minimized in the optimization step, which follows the correspondence step. Correspondences are recomputed in each iteration of the optimization procedure.

**B. Optimization Step**

The optimization step minimizes the surface correspondence error and IMU measurement deviations by adjusting the current estimate of the trajectory. As the underlying problem is continuous, we model the state \( x \) as a vector of stacked trajectory corrections \( \delta T(\tau_s) \equiv [\delta r(\tau_s), \delta t(\tau_s)]^T \) sampled at regular intervals \( \tau_s \), and interpolated for the times in between. The 6DoF trajectory corrections are applied to the current trajectory estimate in the following manner:

\[
T_{\text{corrected}} = T_i \oplus \delta T \oplus T_r
\]

\[
\equiv [\delta \mathbf{r}_i + \mathbf{r}, \delta \mathbf{t} + \mathbf{t}]
\]

where \( T_i \oplus T_r = T \) such that \( T_i \) consists only of the three translational degrees of freedom, and \( T_r \) the three rotational degrees of freedom, of \( T \) (the time notation has been omitted for clarity). This non-standard formulation is preferable since the corrections are expressed in the global frame and has one less term than the standard formulation \( \delta T \oplus T \equiv \)

[^4]: We use various representations of 3D rotations (quaternion, axis-angle, rotation matrix) in our implementation as appropriate. Quaternions are used for storage and interpolations; axis-angle is used for linearizations; and rotation matrices are used when projecting laser points (roll-pitch-yaw values are only ever used for user input/output). Unless otherwise specified, \( r \) indicates an abstract rotation entity.

[^5]: The offset grids approximate a body-centered cubic tessellation with cubic cells instead of octahedra.
decreased at each iteration leads to more reliable convergence.

There are three types of terms considered in the minimization: surfel match errors, IMU measurement deviations, and initial condition constraints.

Surfel match errors guide the trajectory to minimize the distances between surfel correspondences along a common surface normal. For a correspondence between surfels $i$ and $j$, the match error, expressed in the global frame, is:

$$e_{ij} = \xi_{ij} n_{ij}^T (\mu_i (\delta T(\tau_i)) - \mu_j (\delta T(\tau_j)))$$

where the common surface normal $n_{ij}$ is the eigenvector corresponding to the minimum eigenvalue $\lambda_1$ of the sum of the matched surfels’ moment matrices, and $\mu_i$ is the centroid of surfel $i$ (which depends on the trajectory correction at time $\tau_i$).

The coefficient $\xi_{ij} = 1/\sqrt{\sigma_i^2 + \lambda_1}$ is dependent on the sensor measurement noise $\sigma_i$ and the surfel thickness, captured by the eigenvalue $\lambda_1$.

Minimizing the IMU measurement deviations ensures that the estimated trajectory is smooth and matches the observed rotational velocities, translational accelerations, and gravity direction.

$$e_a(\tau) = \Sigma_a^{-1/2} \left( r(\tau) \otimes a(\tau) - \frac{d^2t(\tau)}{dt^2} - g \right)$$

$$e_\omega(\tau) = \Sigma_\omega^{-1/2} \left( \omega(\tau) - \frac{d\mathbf{r}(\tau)}{dt} \right)$$

where $a$ is the measured acceleration vector, $\omega$ is the measured rotational velocity vector, $e_a(\tau)$ is expressed in the global frame, $e_\omega(\tau)$ is expressed in the IMU frame, $\Sigma_a$ and $\Sigma_\omega$ are the IMU measurement covariances, and $g$ is the acceleration due to gravity.

The initial condition constraints enforce continuity with the previous trajectory segment by penalizing any changes to the first three trajectory correction samples in the current segment.

The above error terms are nonlinear with respect to the rotational corrections. Therefore, we linearize the errors about the current best estimate by taking a first-order Taylor expansion, and solve a linear system $Ax = b$, where $x$ is the stacked vector of trajectory corrections. Each error term forms a row of the system matrix $A$ according to its Jacobian with respect to $x$. For purposes of computing the Jacobian, we assume that the trajectory corrections are linearly interpolated in time between the trajectory samples. The vector $b$ is a vector of errors evaluated at the linearization point.

To mitigate the effect of correspondence errors (including inconsistent matches from dynamic objects), the linear system is solved by iterative reweighted least squares where the weights $w_i$ are computed within an M-estimator framework according to a Lorentzian function on the residual

$$w_i = \frac{1}{1 + ((A_{ij}x - b_{ij})/\bar{r})^2}$$

where $A_{ij}$ and $b_{ij}$ are the elements from the row corresponding to match error $e_{ij}$, and $\bar{r}$ is a soft outlier threshold. We have observed that starting with a large outlier threshold which is decreased at each iteration leads to more reliable convergence of the optimization.

C. Timing Latency Estimation

As noted in Section II-B the system is sensitive to millisecond-scale timing latencies between the IMU and the laser. Therefore, we augment the optimization state $x$ with an extra dimension to estimate a latency correction over the processing window. The Jacobians of the match errors then have an additional term for the latency that is dependent on the velocity induced at the surfel centroids (i.e., the velocity of the sensor transferred along the lever arm of the scan ray). Because the rate of change of the latencies is very slow relative to the trajectory segment window length, it is not necessary to model higher order terms such as skew and drift.

D. IMU Bias Estimation

Another significant source of error when working with MEMS IMUs is the bias on the raw acceleration and rotational rate measurements. The 6DoF bias vector is non-stationary and can change significantly over intervals of a few minutes; therefore, pre-calibration is not a feasible option. Rotational rate bias can be estimated when it is known that the sensor is stationary; however, in our application the sensor head is typically oscillating continuously after initialization. Accelerometer bias is typically only observed in the direction of gravity, which requires multiple orientations in order to measure all the components of the bias. Nonetheless, small updates to the bias can be maintained by including correction states in the SLAM optimization and modeling their effect on the IMU measurements.

E. Fixed Views

To reduce the accumulation of drift errors over processing windows, the algorithm maintains a set of surfels from a small number of recent past views, called fixed views, from which match errors are also minimized. For these match errors, the corresponding rows in the system matrix will only have terms for the corrections at times within the processing window (i.e., the trajectory corresponding to the fixed views is not further corrected). A fixed view is taken to be the surfels from the first section of an optimized processing window (which has already been finalized), saved at predetermined distance and angle intervals along the trajectory (i.e., as the trajectory grows, fixed views are generated whenever the growth is larger than either of a predefined distance or angle threshold). A small constant number of fixed views is buffered in order to avoid unbounded growth in computation from generating and processing the additional surfel match error terms.

F. Global Registration

The Zebedee SLAM algorithm can also be applied to globally register a point cloud given an initial guess for the trajectory (such as the open-loop solution described above), to produce a closed-loop trajectory. Rather than processing the trajectory in small increments, the global registration algorithm operates on the entire trajectory in one large window. The algorithm similarly generates surfels from the point cloud.
BOSSE et al.: ZEBEDEE: DESIGN OF A SPRING-MOUNTED 3-D RANGE SENSOR WITH APPLICATION TO MOBILE MAPPING

Fig. 5: Photos and open-loop results from three of the test environments. The trajectories are intended to follow a box shape, which were traced out by hand while shaking Zebedee.

IV. EXPERIMENTS

We evaluate the quality of the Zebedee SLAM solution by considering both the map and trajectory accuracies. For the first series of experiments, we do not have ground truth for the irregular trajectory of the sensor head. Therefore, we instead use a more conventional sensor on a wheeled pushcart that closely follows the position of the Zebedee sensor base (Figure 2b). A spinning SICK LMS291, which rotates the laser at 1 Hz about its middle scan ray, is mounted on the pushcart at a height of 750 mm. The spinning laser produces 13,500 3D points per second in a hemispherical field of view with its center oriented horizontally behind the cart.

The cart is pulled along a trajectory while a person holding and shaking Zebedee follows. As this was an early prototype of Zebedee, the sensor was tethered to the cart for power and data logging purposes—this also implies that the path followed by Zebedee was constrained to be within about a meter of the cart trajectory at all times. In newer versions of Zebedee the battery and logging computer are stored in a backpack. We are able to estimate the trajectory of the spinning laser accurately using an algorithm similar to the original approach upon which the Zebedee SLAM algorithm is based [5]. The global registration algorithm can correct the open-loop cart trajectory for long-term drift and close loops. Since the cart and Zebedee trajectories are not rigidly connected, quantitative metrics for evaluating the Zebedee solution analyze the error between the point clouds rather than the trajectories. Measurements from a Vicon motion capture system (Section IV-C) allow us to evaluate the trajectory accuracy under controlled settings.

It should be noted that when controlled manually, the sensor head oscillation is typically dominated by pitching motion, though some rolling motion still occurs naturally (or is sometimes induced by the operator). Other sensor head motions, such as a lasso-like rotational motion, are also acceptable and handled by the algorithm without modifications.

A. Pseudostationary Experiments

To demonstrate the local quality of the solution, we compare the point clouds generated by Zebedee and the spinning laser while keeping the cart stationary in a variety of environments. We refer to the spinning laser trajectory and point cloud as the reference trajectory and reference point cloud respectively. Zebedee is waved by hand and moved twice around a horizontal box pattern within arm’s reach within approximately 1 m of, and oriented in the same general direction as, the spinning laser. Point clouds are generated using the data from each sensor. The open-loop Zebedee SLAM algorithm (with no fixed views) is applied to the handheld sensor data. Since the cart does not undergo any motion, the raw points can be used directly to generate the reference point cloud.
two point clouds are then globally registered to one another using an iterative optimization procedure that minimizes the error between surfels extracted from the Zebedee points and the reference point cloud. We can then evaluate the local accuracy of the Zebedee point clouds by analyzing the error statistics of each point compared to the closest surfel from the reference cloud and visualizing the magnitude of errors in different environments. The surfels in the reference cloud are generated using a pair of single-level Cartesian voxel grids with resolution of 35 cm offset by half a cell width in each dimension. Since the surface normals are oriented to face away from the sensor, negative errors indicate a Zebedee point being too close to the sensor.

We performed the pseudostationary tests in an office environment with cluttered desks and an offset ceiling (Figure 5a); on an outdoor courtyard with many trees and a non-planar ground (Figure 5b); in an outdoor field with meter tall, bushy grass (Figure 5c); in a long, featureless hallway; in an outdoor road sided by bushes and trees; and in a stairwell (for brevity, photos of the last three environments are not included).

Figure 5 shows overhead views of the estimated trajectories from a sample of the environments. The trajectories follow a box-like pattern, as expected from the input motion. The trajectories estimated for the other three datasets (not pictured) similarly match the intended box pattern. The match error distributions for all six environments are illustrated in Figure 6a. The differences between the Zebedee point cloud and the reference point cloud is greatest in the “tall grass” environment, which is mainly due to variation between the performance of the SICK and Hokuyo lasers: the Hokuyo laser tends to penetrate deeper through the grass, resulting in differences between the point clouds. Despite the wider error distribution, Zebedee performs well in this environment as evidenced by the correct trajectory shape (Figure 5a) and the appearance of the resulting point cloud (not pictured). We observe the highest accuracy (2 –2.5 cm standard deviation of the match errors) in indoor environments containing more predominant large planar surfaces (“hallway” and “stairwell”), whose error distributions tend to have taller peaks (the “office” environment, while indoors, was more cluttered and therefore less dominated by planes as compared to the other indoor scenes). The remainder of the datasets have similar error distributions with standard deviations around 5 cm. Note that since our analysis relies on planar surfaces fit to the point clouds, a lack of strong surfaces in the environment may produce less reliable results (as has occurred in the “tall grass” environment).

The reported range accuracy in the laser manufacturer’s specification is between 3 and 5 cm, which is in agreement with the minimum standard deviation of errors among the environments tested. In general, we observe a slight bias of about 1 –2 cm in the errors towards the sensor, which is likely due to bias in the laser range readings. The scale of this observed bias agrees with the laser manufacturer’s range calibration data (shipped with each sensor); however, this information is only supplied as coarse samples (primarily at 5 m intervals) and the deviations are nonlinear. A negative bias of a few centimeters in range has also been reported by other researchers for the Hokuyo URG-04 [15].

**B. Mobile Mapping Experiments**

For the mobile experiments, we pull the cart closely followed by a handheld Zebedee (which is tethered to the cart) through a loop in an indoor office and an outdoor courtyard. The open-plan office environment is crowded with cluttered desks and other furniture (Figure 5a), while the outdoor courtyard has tall vegetation in the center, building walls on three sides, and a staircase (Figure 5b). In each environment, the looped path is traversed twice. The spinning laser measurements are processed to estimate a globally registered reference trajectory and point cloud, as described above.

The Zebedee measurements are first processed to compute an open-loop trajectory using the incremental online SLAM algorithm, which then provides the initial estimate for the closed-loop trajectory optimization. Though the solution is processed offline after data collection, the open-loop trajectory is computed faster than real-time on a 3.2 GHz Intel Xeon CPU. The software is currently implemented in Matlab, with some of the computationally intensive operations coded in C++ with a MEX interface. When running with zero fixed
views, the online SLAM algorithm processes the data in approximately 62% of the acquisition time, while for two and five fixed views computation requires 71% and 73% of the acquisition time respectively. The processing time of the global optimization depends on the number of reobserved surfaces, which in turn depends on the nature of the environment and the trajectory followed. For the office environment (3.5 minutes acquisition time) the global optimization runs in under one minute, while in the outdoor courtyard (6.5 minutes acquisition time) it runs in under two minutes. Therefore, for both of these datasets, a closed-loop solution can be obtained from the raw data in less than the data acquisition time.

The closed-loop trajectory is used to project all the Zebedee points into a common frame and the resulting point cloud is compared with the reference point cloud in the same manner as in the pseudostationary experiments above. For the global registration, surfels are generated using multi-resolution Cartesian voxel grids at resolutions of 0.5, 1, 2, and 4 m (including a half-resolution offset grid at each resolution).

The trajectories from the indoor office environment are shown in Figures 7a and 7c. We note that both the open-loop and closed-loop Zebedee trajectories closely follow the spinning laser cart trajectory. The point cloud generated from Zebedee in this experiment is displayed colorized by its errors relative to the reference point cloud in Figure 7b. Though the slight inward bias is noticeable, the map is locally consistent and is a good quality representation of the office environment.

Figure 8 illustrates the trajectories and point cloud from the outdoor courtyard environment. Despite the reduction in the prevalence of planar structure, the Zebedee trajectory closely follows the spinning laser trajectory, and the point cloud is a locally consistent and good quality representation of the courtyard environment.

The growth rate of the open-loop trajectory errors for the outdoor courtyard environment can be seen in Figure 9. From repeating this experiment several times, we have observed that the envelope of the open-loop trajectory drifts between about 15 – 25 cm/min positionally from the closed-loop solution, and about 0.4 – 1.5°/min in yaw. This observed open-loop angular drift is considerably smaller than the measured IMU error rates.

The distributions of the point cloud errors for both environments are depicted in Figure 6b and are commensurate with the sensor noise. The indoor errors are slightly lower (standard deviation of 3.9 cm) and less biased than the outdoor errors (standard deviation of 4.1 cm), likely due to the reduced structure in the outdoor environment. The modeling error inherent to the point cloud comparison metric tends to broaden the error distribution in the presence of non-planar surfaces (such as foliage), resulting in an overestimate of the magnitude of the registration errors. Other possible sources of error include the range calibration of the Hokuyo, the beam divergence of the lasers, calibration errors in the spinning SICK (both range...
Fig. 8: SLAM results from the outdoor courtyard environment. (a) and (c) The trajectories estimated from Zebedee (both open-loop and closed) and the spinning laser (closed-loop). The Zebedee trajectories appear at a greater height as the device was held at a level higher than the spinning laser origin. (b) An overhead view of the point cloud generated from the closed-loop Zebedee SLAM solution with the trajectory overlaid. The points and trajectory are colored by error relative to the reference point cloud. The points from the ground and some of the tree canopy have been removed for clarity. (d) An oblique view of the point cloud colored by height. Present in the scene are various leafy and palm trees, a staircase, and building structure along three sides of the perimeter.

calibration and spinning mount calibration), and differences in performance of the two lasers on surfaces such as glass or vegetation.

Timing latency corrections between the Hokuyo laser and MicroStrain IMU clocks for all of the datasets described in this article are shown in Figure 10. We observe that the required correction must sometimes account for drift on the order of one millisecond per minute, which would have significant consequences (see Figure 4) if not properly addressed. In the longer sequences, additional timing information was recorded from the internal clock of the Hokuyo laser. The availability of this timing information enables better initial smoothing of the timing signal, and as a result the latency corrections do not vary as quickly (but can vary within a range well beyond a few milliseconds in longer datasets due to clock skew and drift). In practice, there are various strategies for dealing with the initialization of the timing latency (rather than initializing it to be zero). If the dataset is being processed offline, it can be restarted near the value at which it first settles. If the data is being processed online, the latency can be initialized with the last known estimate.

C. Motion Capture Experiments

Quantitative analysis can be performed on the trajectory provided a ground truth trajectory estimate is available. We therefore evaluate the accuracy of the trajectory in a motion capture environment in which we can track the position of the Zebedee sensor head with high precision. A Vicon system consisting of 14 cameras is capable of tracking the positions of tags containing multiple reflective targets to millimeter-
precision within a region of approximately 2 m by 2 m. The precision on the tag orientation is approximately 1°, mainly due to the small size of the tags. The environment containing the Vicon system is a 10 × 8 m room that includes a 6.5 × 4 × 3 m support frame in the center of the room, with computer workstations, furniture, as well as other infrastructure and office clutter around the periphery. A reflective tag is mounted to the top of the Zebedee sensor head to provide 6DoF tracking as it is oscillated by hand and carried through the motion capture region (Figure 2c). We process the Zebedee data three times using zero, two, or five fixed views (Section III-E) at spacings of min(0.5 m, 120°) along the trajectory, in order to evaluate their effect on the solution. Due to the limited precision in the Vicon-reported tag orientation estimates, we could only perform meaningful analysis for the positional errors of the trajectory. However, we can report that the Zebedee-computed orientations are within the limits of the Vicon system.

D. Dependence on Operator Control

Clearly, the field of view of the handheld Zebedee system is dependent on the motion and control of the operator. A valid question is then: how robust is the system to poor instantaneous oscillatory motion, and how dependent is its performance on the operator’s skill? To investigate this question, we conducted a set of experiments in which the sensor head oscillatory motion is intentionally stopped for several seconds at a time, to simulate operator “error”. In our experience,
such gaps occasionally (but rarely) occur in rugged terrain where the operator may have difficulty maintaining sensor head oscillation when concentrating on balance and footing; or in more benign environments when an inexperienced operator must temporarily slow down; for example, to open a door. The first dataset presented for this evaluation was collected in a stairwell between the first and third floors of an office building (Figure 16b). The operator started at the first floor landing, ascended the stairs to the third floor, then returned to the bottom. At several points during the data collection, the operator intentionally prevented the sensor head from oscillating while continuing to traverse the stairs or landings between staircases. Each stoppage lasted for five to ten stairs, or several strides on a landing (including yaw rotations), and generally covered three- to six-second time intervals.

For this experiment, the second-generation version of Zebedee was utilized (Figure 2d). The main technical differences from the original Zebedee are the use of a MicroStrain 3DM-GX3 IMU (rather than a 3DM-GX2) which is mounted on the back of the laser (rather than the bottom), as well as a spring with slightly different physical characteristics.

Figure 14 illustrates the general sensor head and operator motion over the course of the experiment: the IMU roll and pitch data indicates when the sensor is oscillating, and the sensor base translational velocity indicates how quickly the operator is walking. During the oscillation stoppages, the sensor head pitch and roll are close to zero, while the sensor base (operator) translational speed remains significant.

We expect that the use of fixed views should increase the robustness of the algorithm to momentary sensor head oscillation stoppages since they maintain a longer history of valid 3D surfs to match against. Open-loop trajectories are therefore estimated with and without fixed views for this dataset and illustrated in Figure 15 together with a closed-loop reference trajectory (generated using five fixed views). We observe a significant vertical error (about 5 m) near the beginning of the open-loop trajectory in the case with no fixed views. The instance where this error occurred involved a relatively long break (approximately six seconds) where the operator walked up ten stairs with the sensor head oscillations effectively stopped (the first circled region in Figure 14). We also note an error in yaw appearing towards the end of the zero-fixed view trajectory; though in general the solution is locally consistent aside from the large vertical error event. The results demonstrate that the addition of fixed views greatly improves robustness of the algorithm: the two-fixed-view open-loop trajectory does not contain any major errors and drifts minimally from the reference trajectory. The open-
loop trajectory is close enough to the true solution that the registration algorithm is able to produce a globally consistent closed-loop trajectory that is similar to the reference trajectory. The positional RMS error of the two-fixed-view closed-loop trajectory relative to the reference trajectory is 1.7 cm. We have further analyzed the performance of the algorithm in the presence of anomalies in both the outdoor courtyard and stairway environments. Large error events, such as the one seen in the zero-fixed-view stairway example, were particularly rare, and only occurred under extreme conditions of sensor head motion stoppage. We do not observe any noticeable negative effects on open-loop error when moderate sensor head oscillation stoppages occur or if a few moving pedestrians are present in the scene.

We further consider removing the operator’s ability to manually control the sensor head oscillations by fixing the sensor base directly to the backpack with the center of the sensor field of view facing behind the operator (Figure 24). Here, the natural accelerations from walking are the only inputs to the sensor motion. The primary drawback to mounting the sensor in this way is the loss of the operator’s ability to fully control the sensor field of view if complete coverage of all surfaces in a complex environment is desired. However, the benefit of having both hands available is critical in some applications such as first response. We collected a dataset in the stairwell, following a similar path to the previous experiment, and observe that there is sufficient sensor head motion to reliably and accurately estimate the trajectory. The drift accumulated from the open-loop trajectory (using two fixed views) is illustrated in Figure 17 with the most significant components in height (approximately 10 cm/min) and yaw (approximately 2°/min). These errors are similar to those that have been observed in the handheld experiments and well within the limits of the global registration convergence region. The drift can also be visualized by comparing the open-loop and closed-loop point clouds (Figure 15). We have tested the backpack-mounted configuration in other environments, both indoors and outdoors, with similar results. Therefore, we conclude that a hands-free Zebedee configuration (e.g., mounted on a backpack, shoulder, or helmet) is a viable option for deploying the sensor on a human operator. In contrast to the backpack mapping system proposed by Chen et al. [14], the Zebedee design is lighter and simpler (one lidar as opposed to three), requiring less cost and calibration, and the Zebedee system is capable of SLAM in arbitrary environments (as opposed to being restricted to highly planar indoor environments).

V. Conclusions and Discussion

We have introduced a novel design of a 3D range sensor consisting of a 2D lidar scanner mounted on a flexible spring useful for 3D perception in robotics and other mobile applications. We have demonstrated a particular realization of the general design concept, which we call Zebedee. Analysis of the frequency response and oscillation amplitudes of the device hardware can be performed to determine suitable component
specifications for a given implementation. A specialized simultaneous localization and mapping algorithm has been developed that can be used both for incremental motion estimation and global point cloud registration. In fact, the same code base can be (and has been) used for a variety of mobile 3D laser configurations including spinning, nodding, and trawling 2D lidars. The solution has been evaluated in a variety of environments, both indoor and outdoor, and quantitatively analyzed using a motion capture system for ground truth position estimates. The results show that the localization errors are within acceptable limits for many applications, and the precision of the point clouds is generally commensurate to a solution from more traditional spinning laser hardware. The motion capture experiments demonstrated that the Zebedee trajectory can be estimated with a sub-centimeter accuracy. We find that for all of the experiments considered (with the exception of the zero-fixed-view solution when demonstrating long oscillation stoppages), the open-loop trajectories are sufficiently accurate to initialize the closed-loop optimizations, without the need for additional coarse registration constraints (e.g., from place recognition). The results further demonstrate that with the inclusion of fixed views, the SLAM algorithm is robust to irregular operator performance including brief stoppages in sensor head oscillations on the order of several seconds duration. We conclude that the proposed sensor would be a reasonable choice for applications where a relatively inexpensive, lightweight, and mechanically simple 3D range sensor is required. In general, the performance of a passively actuated 3D sensor would be dependent on the quality and mechanical properties of the range sensor employed.

The mobility of the Zebedee sensor as a handheld device allows access to most environments accessible to humans, including rough, natural terrain and stairways. Further to the experiments presented here, we have deployed the system in applications including large-scale assembly plants and indoor spaces, as well as mapped natural environments that would otherwise be inaccessible to standard 3D mapping systems. For vehicle-mounted versions, the passive linkage concept can be applied to a variety of range sensors of varying performance and mass specifications depending on the application requirements. Simple mechanical modifications can also be made to the design to limit the scanning range of the sensor when required.

A. Limitations and Lessons Learned

There are, of course, some limitations to the system. Common to all laser-based SLAM algorithms, there exist pathological environments where the motion is weakly or non-observable, such as long, smooth tunnel-like spaces, or large, featureless, open areas. Such cases are identifiable by examining the distribution of surface normals, but extreme cases can be difficult to recover from without additional sensors. Environments dominated by moving objects can also challenge the algorithm since it becomes difficult to distinguish true outliers. In some applications, the Zebedee design requires fairly continual excitation in order to induce motion of the sensor head. For example, the device may not be appropriate for electric ground vehicles operating with infrequent accelerations on smooth terrain. However, there are many potential excitation sources in typical applications such as rough terrain, vibrating engines, legged platforms, wind disturbances for air vehicles or fast-moving ground vehicles, or wave motion for marine surface vehicles. It should also be noted that the trajectory computed by the SLAM solution tracks the sinuous trajectory of the sensor head: further processing would be required to also estimate the trajectory of the platform on which the sensor base is mounted. Since the compression of the spring is small compared to the bending of the spring, it behaves approximately like a rigid link with a universal joint. Thus, a simple transformation approximates the position of the sensor base frame relative to the sensor head. An additional IMU can be mounted on the base platform to further estimate the relative platform orientation using the knowledge that the two systems are coupled. More accurate results can potentially be obtained by observing and tracking structure on the base platform within the local point cloud. Finally, a limitation of the SLAM solution we present is that it does not compute an uncertainty estimate, though ongoing work is investigating this aspect. The primary challenge in estimating a consistent covariance is the complexity of accurately modeling the many sources of uncertainty, including: surface normal directions, match sensitivity, linearization and interpolation of constraint coefficients, fixed surfel views, sliding window initial conditions, IMU prior modeling, IMU biases, etc.

There have been several lessons learned while developing and testing the Zebedee device. An important observation is that timing accuracy is critical: even millisecond errors between the IMU and laser scanner can cause significant distortions of the point cloud data. In early Zebedee models, we experienced some problems with wear on the IMU USB cable, which would eventually necessitate replacement of the cable. Using a stronger cable solved this problem, though in the latest version we have employed a coiled cable to further reduce strain. Cabling passing through the spring can also have adverse effects in terms of dampening the sensor motion, so extra care must be taken in determining cable length, diameter, and tension.

B. Future Work

Future work will involve further development of the sensor design and algorithms, as well as experimentation in additional applications. The mechanical design process would benefit from a parameterized model of the mass-spring system relating the input spring constant, sensor head mass, and geometry, to the resonant gains and frequencies of the resulting motion. Ongoing efforts are focused on extending typical helical spring models, which tend to consider in-axis motion (compression or tension) rather than out-of-axis motion (bending). We continue to improve the performance of the SLAM algorithm to increase its precision. Along these lines, we are investigating methods for calibration of the range bias of the laser, as well as modeling the beam divergence of the laser (which can be on the order of 0.6°–0.8° for lasers commonly used in robotics). We are also investigating the importance of the IMU quality to
determine how low-cost an IMU can be used while still producing accurate results. The current SLAM software is written in Matlab, with some components written in C++ and called through MEX wrappers. This implementation is able to run faster than real-time on a desktop computer; however, our goal is to enable online operation on a lightweight laptop. A C++ version of the software is currently being developed to run the incremental SLAM algorithm and global optimization online faster than real-time on a desktop computer; however, our goal is to enable online operation on a lightweight laptop. A C++ version of the software is currently being developed to run the incremental SLAM algorithm and global optimization online in real-time on a portable computing platform. An ongoing incremental SLAM algorithm and global optimization online faster than real-time on a desktop computer; however, our goal is to enable online operation on a lightweight laptop. A C++ version of the software is currently being developed to run the incremental SLAM algorithm and global optimization online in real-time on a portable computing platform. An ongoing area of research is investigating various scene decomposition methods for generating surfels. While the current solution works sufficiently well, this step is one of the computational bottlenecks in the system. We believe that a more sophisticated technique could result in fewer, but higher quality, surfels enabling computational savings while still maintaining a high degree of accuracy and robustness. Further applications we are considering are the use of Zebedee for obstacle detection on an off-road vehicle (e.g., using larger SICK lasers on stronger springs) or a small rotary wing UAV, and using local point cloud views for place recognition within a larger known map. Other implementations of the general concept may use other range sensors or mechanical linkages, such as floating the device in a viscous fluid, use of elastic cables, or hanging a sensor on a rope as a pendulum.

ACKNOWLEDGMENTS

The authors would like to thank Steven Martin, Timothy Gurnett, and the Queensland University of Technology for use of their equipment and assistance with the Vicon experiments. The design of the second-generation handheld Zebedee device (Figure 2d) was developed with assistance from CMD Product Design and Innovation (http://cmd.net.au). A patent application associated with the technology described in this article has been filed by the CSIRO in 2010.

REFERENCES