

Continuous-Time Three-Dimensional Mapping for Micro Aerial Vehicles with a Passively Actuated Rotating Laser Scanner



Lukas Kaul

Autonomous Systems, CSIRO, Brisbane, Australia and Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Germany

e-mail: lukas.kaul@kit.edu

Robert Zlot

Autonomous Systems, CSIRO, Brisbane, Australia

e-mail: Robert.Zlot@csiro.au

Michael Bosse

Autonomous Systems Laboratory, ETH, Zürich, Switzerland

e-mail: mike.bosse@mavt.ethz.ch

Received 8 January 2015; accepted 18 May 2015

The ability to generate accurate and detailed three-dimensional (3D) maps of a scene from a mobile platform is an essential technology for a wide variety of applications from robotic navigation to geological surveying. In many instances, the best vantage point is from above, and as a result, there is a growing demand for low-altitude mapping solutions from micro aerial vehicles such as small quadcopters. Existing lidar-based 3D airborne mapping solutions rely on GPS/INS solutions for positioning, or focus on producing relatively low-fidelity or locally focused maps for the purposes of autonomous navigation. We have developed a general-purpose airborne 3D mapping system capable of continuously scanning the environment during flight to produce accurate and dense point clouds without the need for a separate positioning system. A key feature of the system is a novel passively driven mechanism to rotate a lightweight 2D laser scanner using the rotor downdraft from a quadcopter. The data generated from the spinning laser is input into a continuous-time simultaneous localization and mapping (SLAM) solution to produce an accurate 6 degree-of-freedom trajectory estimate and a 3D point cloud map. Extensive results are presented illustrating the versatility of the platform in a variety of environments including forests, caves, mines, heritage sites, and industrial facilities. Comparison with conventional surveying methods and equipment demonstrates the high accuracy and precision of the proposed solution. © 2015 Wiley Periodicals, Inc.

1. INTRODUCTION

The ability to generate accurate three-dimensional (3D) maps of real-world environments is a key capability for many surveying and robotic mapping applications. Though a local view of the environment may be sufficient for some robotic operations, robots often need an accurate global model of the environment to perform efficiently, or in some cases, producing a map is a primary goal. In addition to the relevance of 3D mapping to robotics, surveying, Geographic Information Systems (GIS), and reality capture are significant global industries, with a range of applications that include mapping streets, tunnels, and civil infrastructure; mines and industrial sites; buildings; cultural heritage sites; natural terrain, caves, and forests. Laser scanning technol-

ogy is commonly employed to capture precise range measurements for generating 3D point cloud models. Ground-based scanning has been the most prevalent modality for a number of reasons; but with the increasing availability of micro aerial vehicles (MAV), there is significant potential for a low-cost and lightweight airborne scanning solution. Scanning from a low-altitude airborne platform affords access to some sites not otherwise reachable and provides the ability to measure surfaces that cannot be viewed from ground level (Figure 1).

There are two key challenges that arise in the context of laser mapping from a MAV. The first is the need for accurate localization, even in scenarios where global navigation satellite systems (such as GPS) are not available—for example, in or near buildings, forests, or underground. A simultaneous localization and mapping (SLAM) solution is the preferred way to address this problem as it avoids

Direct correspondence to: Lukas Kaul, email: lukas.kaul@kit.edu

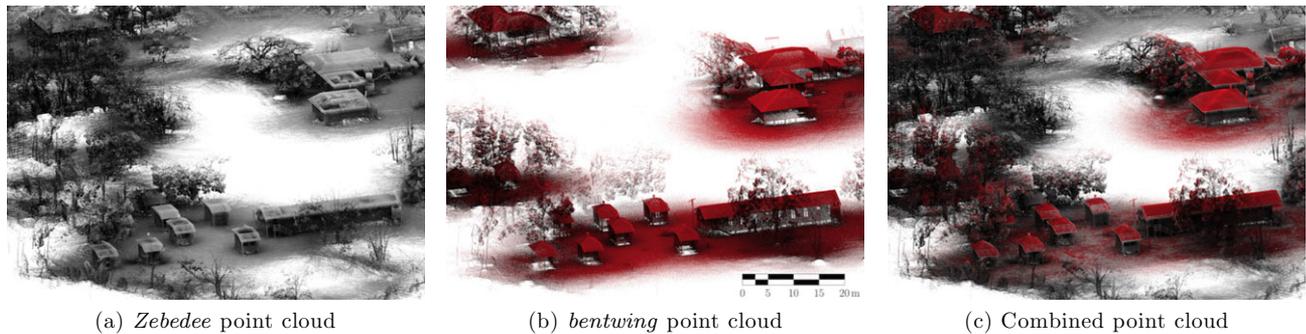


Figure 1. An example scenario illustrating the complementarity between ground-based and aerial laser scanning. (a) A section of the Peel Island Lazaret acquired from ground level with the *Zebedee* handheld 3D mapping system (Zlot et al., 2014). (b) Point clouds acquired in the same area with the *bentwing* aerial mapping system. The data were collected in multiple flights. (c) Combined point cloud from the *Zebedee* and *bentwing* data. The point clouds are colored using an ambient occlusion shader, with the *bentwing* points additionally shaded in red. Rooftops and tree canopies, which are not visible from ground level, are covered by the aerial scan. Conversely, the interiors, underspaces, and some other features of the buildings are not visible from the workspace safely accessible by the quadcopter.

the need for additional positioning infrastructure or a priori knowledge of the environment. A second challenge is the limited payload capacity of MAV platforms, which is often a few kilograms or less. This constraint in practice necessitates the utilization of laser scanners with a limited (often two-dimensional (2D)) field of view rather than heavier systems with an inherently large 3D field of view. A 3D SLAM solution, however, requires the scanner to repeatedly acquire measurements within a relatively wide 3D field of view at an appropriate rate (every few seconds) to facilitate reliable incremental motion estimation. For accurate modeling, the scanner trajectory must be represented as a continuous-time function, as it is not possible to rely on an aerial vehicle remaining sufficiently motionless at discrete poses while scanning with such a sensor configuration (e.g., Nüchter, Lingemann, Hertzberg, & Surmann (2007)). To meet these challenges, we propose the application of a continuous-time 3D SLAM algorithm to a rotating 2D laser scanning payload mounted on a quadrotor MAV. While the algorithms have been presented previously (Bosse & Zlot, 2009a; Zlot & Bosse, 2014b), to our knowledge this is the first reported time a continuous-time SLAM framework has specifically been applied to address the problem that aerial platforms are in continual motion. In addition, we introduce a novel sensor payload design in which a small, lightweight 2D laser scanner that is rotated at a rate on the order of 1 Hz via a novel passive mechanism that exploits the downdraft from the rotor blades.

Conventional airborne lidar surveying uses downward-pointing long-range laser scanners from full-scale airplanes or helicopters, typically flying at altitudes between 200 m and 1,000 m. The data resulting from these systems tend to be relatively sparse terrain maps, with meter-scale spacing between points (Baltsavias, 1999). Low-altitude flight is required for denser point

clouds and to measure surfaces that are not predominantly vertically oriented. Commercial products and demonstrators based on small rotary MAV platforms that fulfill these requirements have recently been introduced. These systems, such as the Riegl RiCOPTER¹ and products from Phoenix Aerial Systems², mostly rely on GPS/INS systems for positioning which limits the environments in which they can be deployed. An aerial lidar surveying system recently introduced by XactMaps³ claims not to rely on GPS and uses a Velodyne 3D laser scanner. Velodyne lasers have been used for mapping in scan-matching-based SLAM solutions (Moosmann & Stiller, 2011), and due to their high scan rate, a 360° scan can often be treated as instantaneous (or approximately corrected according to the current velocity estimate). As a result, a continuous-time SLAM solution is not strictly required in this case, but the downside of this sensor configuration is that the fields of view of Velodyne lidars are limited to 30–40° in the vertical. The sensors currently used by commercial systems tend to be relatively expensive, heavy, and high powered, necessitating custom-made, relatively large and expensive multirotor platforms. A lower-weight, lower-cost Velodyne sensor that has recently become available may help address some of these limitations (though not the restricted field of view).

In the research community, a number of works have been presented that deploy laser-range finders on small, low-flying aerial vehicles. Lin, Hyyppä, & Jaakkola (2011) have demonstrated aerial lidar mapping with a small-scale single-rotor helicopter, using comparatively heavy laser

¹<http://www.riegl.com/products/uasuv-scanning/new-riegl-ricopter-with-vux-sys>

²<http://www.phoenix-aerial.com>

³<http://xactmaps.com>

scanners. Their work does not focus on generating high-precision maps of the explored environment but rather on methods to detect features such as streets, trees, or light posts. More specifically, in robotics research, systems that deploy lidar-equipped MAVs usually do not focus on generating dense or highly accurate maps, but on such maps that are useful for autonomous navigation, for example, path planning and obstacle avoidance. Robotics research has also tended to explore the possibilities of using lightweight and low-power 2D lidar sensors on MAVs as opposed to heavier and more expensive 3D scanners. Some systems developed in this context include a 2D lidar with a horizontal scanning plane that is rigidly coupled to the airframe of a multi-copter MAV. Some of these systems only focus on methods for generating 2D maps of the environment that enable safe autonomous navigation, which is sufficient for navigation if the vehicle flies at a more or less constant height (Grzonka, Grisetti, & Burgard, 2009). The described setup can also be used to generate 3D maps when the vehicle not only moves horizontally but also vertically (Bachrach, Prentice, He, & Roy, 2011). A problem with this approach, when applied to environment mapping rather than to navigation, is the limited field of view and the lack of overlap between adjacent scans taken by the laser scanner. The laser scanner is not able to capture the parts of the environment that lie directly above or below the MAV. Also, the lack of repeated lidar coverage over short intervals precludes high-frequency 3D registration, necessitating a separate positioning system and generally resulting in reduced map quality. Shen and Michael (2012) augment a horizontal 2D laser scanner with mirrors to direct some laser rays vertically in a system where a separate Kalman filter (EKF-)based state estimation system is required for positioning. Here, the laser measurements contribute to the state estimate when they provide sufficient information for localization. The system is primarily focused on autonomous navigation, with 3D occupancy grid maps produced mainly for that purpose.

A lightweight 2D laser scanner can be actively spun on a multirotor MAV (Scherer et al., 2012; Droeschel, Holz, & Behnke, 2014a; Holz & Behnke, 2014) to expand the field of view and ensure significant scan overlap over short time intervals. Droeschel, Stückler, & Behnke (2014b) have presented a SLAM algorithm that can reconstruct the 6 degree-of-freedom (DoF) trajectory of the flying vehicle and produce 3D occupancy grid maps that can be used for indoor navigation. Holz and Behnke (2014) have presented algorithms that can produce relatively high-resolution 3D maps of structured environments based on data collected with the same platform. While these maps show more complete coverage and significantly higher resolution than other low-altitude airborne mapping approaches, they are still sparser and less accurate than maps generated with conventional ground-based laser surveying techniques. An off-axis spinning laser has been demonstrated for 3D mapping of river-

ine environments (Scherer et al., 2012). In this case, the pose of the system is determined by a separate state estimation algorithm that uses visual odometry, inertial measurements, and intermittent GPS. The reliance on additional hardware and software systems to estimate the laser scanner trajectory adds complexity to the mapping system, though in this case the additional sensors are required for other purposes as well. Registration of the laser data in their system could potentially improve the point cloud quality and state estimate using an approach similar to the one we propose. Our work aims at developing a simple and versatile lidar payload with a large 3D field of view that ensures sufficient scan overlap over short time intervals to facilitate high-frequency and high-precision registration, and that delivers 3D maps similar in quality and density to ground-based laser surveying in a wide range of environments.

The contribution of this work is a robust laser-based mapping solution for MAV platforms that can operate in a wide variety of environments and produce accurate and highly detailed 3D point cloud maps. While the core algorithms have been described previously, to our knowledge this is the first time an application of continuous-time SLAM technology has been applied to aerial platforms. Given that aerial platforms are constantly in motion while flying, a continuous-time representation of the sensor trajectory is essential in order to produce an accurate result when using a sensor that cannot be assumed to acquire data instantaneously. A further contribution is in the introduction of a passive-actuation mechanism used to rotate the laser scanner. The mechanism is designed to exploit the available downdraft from the quadrotor blades to impart motion rather than using a motor to spin the sensors. The system is demonstrated in a variety of field applications, demonstrating its suitability to mapping cultural heritage sites, caves, mines, forests, and industrial sites, and evaluating its performance in terms of point cloud quality and trajectory accuracy. The passive-actuation mechanism design concept is closely related to the *Zebedee* system previously developed by two of the authors (Bosse et al., 2012). *Zebedee* is a handheld 3D mapping system that uses a spring to convert the natural walking motion of the operator into rotations of a laser scanner in order to extend its field of view and to achieve an appropriate rate of repeated coverage. Similarly, the proposed platform, which we call *bentwing* (Figure 2), uses the existing quadrotor downdraft to rotate the sensor payload. These related systems, and more generally any suitable pair of ground-based and aerial-based mapping systems, have a complementary relationship in that the aerial platform is able to measure surfaces that are outside of line-of-sight from the ground-based system, and vice versa. As a result, it is often ideal to have both systems available in the field, and it is relatively straightforward to combine data from both sources to achieve more complete coverage of a site (Figure 1).

(a) *bentwing* 3D mapping platform(b) *bentwing* in flight

Figure 2. The *bentwing* aerial 3D mapping platform. (a) The sensing payload on the bottom consists of a laser scanner, an inertial measurement unit (IMU), and custom-designed wings, connected via a slip ring to the dedicated data-logging computer above. Lithium polymer batteries provide up to 15 minutes of flight time. (b) While in flight, the downdraft from the quadcopter acts on the wings to generate continuous rotational motion of the sensors.

2. CONTINUOUS-TIME MAPPING

To develop an airborne 3D laser mapping system capable of operating in GPS-deprived environments, a number of factors need to be taken into consideration. The first is that an airborne platform is always moving, even while attempting to hover. Second, to obtain a lightweight and inexpensive mapping payload, the most practical approach is to actuate a laser scanner to increase its field of view. As a consequence, a full 3D scan requires a non-negligible amount of time to capture, during which the platform is continuously moving. Therefore, in order to ensure an accurate, undistorted map, the sensor pose must be calculable for every time along the trajectory. In other words, the trajectory must be represented as a continuous-time function. In this section, we first summarize our approach to continuous-time SLAM that enables accurate airborne mapping. In the section that follows, the hardware design that produces a wide 3D field of view and enables reobservation of the environment at an appropriate rate for the SLAM solution is described.

Our continuous-time SLAM solution is based on a view-based approach focused on estimating the trajectory of the sensor payload. A *trajectory* is defined as a continuous-time function that specifies a 6 DoF pose at each time. For the sensor payload described here, the trajectory is estimated from laser range measurements, inertial measurements, and additional constraints based on the platform motion model. Given an accurate trajectory, it is straightforward to project the range measurements into a common coordinate frame to produce a globally consistent 3D point cloud map. A brief general description of our framework is provided in this section, as the approach has been published previously in more detail (Bosse & Zlot, 2009a; Bosse et al., 2012; Bosse & Zlot, 2013; Zlot & Bosse, 2014b) and

is applied to the *bentwing* platform without any significant modifications.

There are two main algorithmic components in our solution framework. The first is a non-rigid trajectory estimation and map registration algorithm (which we abbreviate as “non-rigid registration”). Non-rigid registration is used both for online incremental motion estimation (laser odometry), as well as global optimization of the overall trajectory (Bosse & Zlot, 2009a; Bosse et al., 2012; Zlot & Bosse, 2014b). For long-duration datasets in which accumulated drift error may be significant, a coarse registration step is necessary to provide a better initial trajectory estimate for the global optimization. These cases are handled by a place recognition solution (Bosse & Zlot, 2013), which is the second main algorithmic component of our system. Multiple datasets can also be automatically merged using place recognition followed by global non-rigid registration.

The data-processing steps in the general workflow are executed as follows. First, the open-loop trajectory of the sensor payload is estimated by applying the non-rigid registration algorithm incrementally to the last few seconds of the incoming stream of raw data falling within a sliding window. Global registration is then performed by first applying the place recognition algorithm to identify loop closure constraints, which are then incorporated into the trajectory through a pose graph optimization that coarsely registers the data. Since the battery capacity constrains the duration of *bentwing* datasets, the place recognition step is typically not required and can often be omitted. Fine registration is achieved through the non-rigid registration algorithm applied to the full trajectory, thereby producing a globally consistent closed-loop trajectory and point cloud. When processing multiple overlapping datasets, the above steps are applied to each set individually and then place

recognition is used to roughly align the datasets with respect to one another. Fine registration for the combined solution is achieved by applying the global non-rigid registration algorithm to all of the data. Each of the core algorithmic components is briefly described in the following subsections.

2.1. Non-rigid Registration

For non-rigid registration, the sensor trajectory is treated as a continuous function of time, and registration of the point cloud is achieved by applying varying corrections along the trajectory. In contrast, rigid registration methods such as iterative closest point, treat the sensor origins as a pair of discrete poses based on the assumption that measurements are captured instantaneously (or equivalently while stationary) and solves for a single transformation between the sensor coordinate frames that best aligns the points. Our non-rigid registration algorithm can generally be applied to data in the form of a prior trajectory and associated point cloud, which may be acquired by a variety of acquisition methodologies. In the *bentwing* context, the algorithm is used at two timescales: for incremental motion estimation using a sliding window containing a few seconds of measurements from the data stream; and for global registration in which the full trajectory containing minutes or hours of data is corrected based on areas of overlap in the point cloud.

Non-rigid registration is expressed as a nonlinear optimization problem that takes a prior trajectory as input and computes corrections to that trajectory to minimize errors between the measurements and motion constraints. A trajectory is defined as a function $T(\tau)$ that specifies a 6 DoF transformation for any time value τ in the domain. In practice, the trajectory transforms a point \mathbf{p} measured at time τ from the sensor frame S to the world frame W as follows:

$$\mathbf{p}_W = T_W^S(\tau) \oplus \mathbf{p}_S \quad (1)$$

$$= \mathbf{r}_W^S(\tau) \oplus \mathbf{p}_S + \mathbf{t}_W^S(\tau), \quad (2)$$

where \mathbf{p} is a point in the specified frame, \mathbf{t} and \mathbf{r} , respectively, represent the translational and rotational components of the 6 DoF transformation, and \oplus is the transformation composition operator. The frame labels are often omitted for convenience when it is clear that the transformations are from the sensor to world frame. For the optimization problem, the trajectory is decomposed into a baseline trajectory ($\mathbf{t}_0(\tau)$, $\mathbf{r}_0(\tau)$) and a small correction: ($\delta\mathbf{t}(\tau)$, $\delta\mathbf{r}(\tau)$):

$$T(\tau) = (\delta\mathbf{t}(\tau) + \mathbf{t}_0(\tau), \delta\mathbf{r}(\tau) \oplus \mathbf{r}_0(\tau)) \quad (3)$$

The corrections $\delta\mathbf{x} = (\delta\mathbf{t}, \delta\mathbf{r})$ serve as the state in the optimization problem. Implementationally, trajectories (and corrections) are stored as samples at a reasonable frequency for the motion bandwidth (e.g. 100 Hz) and a spline interpolates transformations for times between the samples.

The optimization problem is linearized into a system of equations $\mathbf{A}\delta\mathbf{x} = \mathbf{b}$ and is solved using a robust iterative least squares method. Each row of the linear system represents a constraint that encodes either a prior assumption or measurement error. Prior assumptions model the platform motion, including smoothness of the trajectory corrections, initial conditions, and deviation from a reference velocity (i.e., from a prior trajectory estimate). Measurement errors aim to minimize direct differences between measured and predicted sensor readings (e.g. deviation between the trajectory and IMU readings), or registration errors between pairs of corresponding environment observations from an exteroceptive sensor (e.g. a laser scanner) taken at different times. Constraints are generated by differentiating the error vector with respect to the perturbations (corrections) and linearizing, thus the general form of each constraint is $J\delta T = e$, where J is the Jacobian of the nonlinear error with respect to the corrections. Each constraint is weighted based on its intended influence on the solution as determined by measurement and motion models. Constraints based on data correspondences are further weighted according to a Cauchy influence function in an M-estimator framework to provide robustness to potentially erroneous data correspondences. Finally, to solve for $\delta\mathbf{x}$, a sparse Cholesky factorization routine is applied to the modified system:

$$\delta\mathbf{x} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{b}, \quad (4)$$

where \mathbf{W} is the diagonal matrix of constraint weights. The system matrices \mathbf{A} and \mathbf{b} are premultiplied by \mathbf{A}^T for efficiency because there are many more constraints than state dimensions. This construction also allows $\mathbf{A}^T \mathbf{A}$ and $\mathbf{A}^T \mathbf{b}$ contributions from the various types of constraints to be summed to generate the full system.

The primary constraint types included in the non-rigid registration formulation for this application are as follows: minimizing correspondence errors in the laser data; minimizing deviations of the trajectory with respect to inertial measurements; minimizing deviations of the velocity with respect to the prior trajectory; ensuring smoothness and continuity in the trajectory. Definitions of these constraints are described in our previous publications (e.g. (Zlot & Bosse, 2014b)); however, we describe the data correspondence constraints in more detail here due to their importance in understanding the design requirements and flight limitations for the quadcopter platform and sensor payload. As laser measurements are actually samples from physical surfaces, we consider the error between corresponding surface patches that are estimated from local groups of raw laser points. In particular, we define surface elements, or *surfels*, to be approximately planar patches derived from a spatial and temporal neighborhood of points. Surfels can be efficiently generated by decomposing space and time into a multiresolution voxel grid and computing the mean position μ (with timestamp τ) and second-order moment matrix S of the

points in each grid cell. The second-order moment matrix S provides information on shape, including the surface normal \mathbf{n} . The timestamp τ of a surfel, computed as the mean time of the contained points, is associated with an instant in the trajectory. The range of times permitted in a surfel is limited in order to ensure the representation is temporally local (minimizing initial distortion)—multiple surfels can be considered in each spatial grid cell if the points therein exceed the defined temporal limit. Minimizing the error between a pair of *matched* surfels observed at different times constrains the trajectory at those two instants of time. The match error constraint for a correspondence between surfels i and j , expressed in the global frame is:

$$e_{ij} = \xi_{ij} \mathbf{n}_{ij}^T (\mu_i - \mu_j) \quad (5)$$

$$\xi_{ij} = \left(\sqrt{\sigma_r^2 + \lambda_1} \right)^{-1}, \quad (6)$$

where the surface normal \mathbf{n}_{ij} is the eigenvector that corresponds to the minimum eigenvalue of the sum of the moment matrices $S_{ij} = S_i + S_j$ for surfels with positions at μ_i and μ_j . The constraint weight ξ_{ij} is based on estimated measurement noise σ_r and ellipsoid thickness λ_1 (minimum eigenvalue of the combined second-order moment matrix S_{ij}). Correspondences between surfels from each level of the multiresolution voxel grid are identified by an approximate k -nearest neighbor (KNN) search⁴, where the search space is a weighted combination of the surfel position and normal vector. Additional steps filter out grossly distant matches and nonreciprocal matches. Because false positive matches are not completely eliminated, the optimization framework is designed to be robust to a moderate number of outliers via the M-estimator weighting described above.

A special case of surfel match error constraints are *fixed surfel* match error constraints, where surfels are aligned relative to a set of immovable surfels. The inclusion of these *fixed views* in the incremental open-loop trajectory generation context uses short (typically 1.5 s) snapshots of surfels from a few meters earlier in the trajectory to help reduce drift in a similar manner to keyframes in visual odometry algorithms. Fixed views can also be employed in the global context to match the acquired data against an existing prior map.

The optimization framework permits other relevant variables to be included in the state $\delta\mathbf{x}$ along with the trajectory corrections. Of particular relevance for the current application is the estimation of the temporal latency between sensors, IMU biases, and the physical offset (6 DoF transformation) between the laser and IMU frames. These

extra state variables augment the trajectory state and are estimated during the sliding-window trajectory optimization stage. As demonstrated previously (Bosse et al., 2012), the solution quality is sensitive to millisecond-scale timing latencies between the sensors. An extra time latency term is included in the Jacobian of the match errors. This term 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). Though the timing latency is not constant over time, its rate of change is relatively slow and therefore higher-order terms such as skew and drift do not need to be explicitly modeled. The 6 DoF IMU bias vector are also modeled with additional variables in the state vector. Though an estimate of the biases is computed while the platform is stationary before takeoff, not all DoF can be calibrated accurately this way, and the values drift over time, requiring regular updating. Variables for the estimation of the laser-IMU physical offset are also included in the state vector to account for any inaccuracies in calibration or disturbances that might have slightly perturbed the mechanical layout.

Solving the linear system $\mathbf{A}^T \mathbf{W} \mathbf{A} \delta\mathbf{x} = \mathbf{A}^T \mathbf{W} \mathbf{b}$ is an iterative process implemented as a nested pair of loops. A prior trajectory and the relevant measurements are provided as input. The outer loop forms the surfel matches and linearized constraints based on the current trajectory estimate. The inner loop repeatedly solves the system, updating Cauchy weights for the data correspondences on each iteration (without requiring that the entire system be regenerated). In typical cases, the inner loop runs to two or three iterations, while the outer loop can require five to ten, depending on the quality of the prior and the context of the implementation (local or global registration). The surfels can be transformed according to the latest trajectory estimate on each outer loop iteration and do not need to be explicitly recomputed from raw laser points at any time during the optimization.

The output of the optimization is an updated trajectory that has had the estimated corrections applied to it.

Our continuous-time SLAM solution has been continuously developed from an initial version originally published in 2009 (Bosse & Zlot, 2009a). Since that time, several research groups have proposed solutions in this area for incremental motion estimation (Dong & Barfoot, 2012; Anderson & Barfoot, 2013; Tong, Anderson, Dong, & Barfoot, 2014; Alismail, Baker, & Browning, 2014; Zhang & Singh, 2014) and global non-rigid registration (Elseberg, Borrmann, & Nüchter, 2013). One key difference between our solution and the others is that ours considers corrections to the trajectory in the state rather than the trajectory itself. Because the corrections can be adequately represented at a lower frequency, this allows us to reduce the number of variables (spline knots) required in the state while still maintaining high-frequency information from the trajectory. Our solution also uses a variety of constraint types in addition to

⁴We use a modified version of the `libnabo kd-tree` library (Elseberg, Magnenat, Siegwart, & Nüchter, 2012), in which we have adapted the search to ignore matches that are closer in time than a specified threshold.

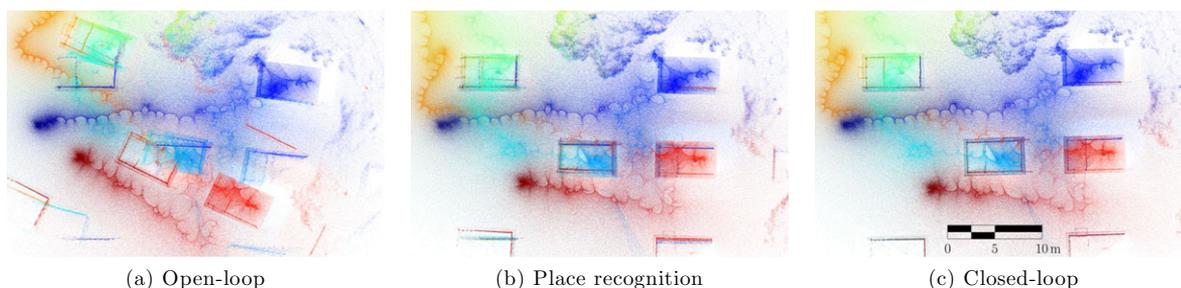


Figure 3. Example of the non-rigid registration and place recognition framework applied to an individual dataset scan of several huts on Peel Island (see Figure 1). In this view, the structures are seen from overhead and colored according to time (blue to red). (a) A portion of the point cloud generated during the initial open-loop trajectory generation phase. Because of drift error in the solution, the data from different passes are not aligned. (b) The same area in the point cloud after the place recognition phase. The data are now coarsely aligned. (c) The point cloud after global non-rigid registration using the coarsely aligned place recognition solution as input.

those based on feature matches and has been applied to solve trajectory estimation problems for both incremental generation and global registration. To the best of our knowledge, the other approaches have not been demonstrated at the speeds, scales, or variety of platform configurations at which our solution has been demonstrated (Zlot & Bosse, 2014a,b; Zlot et al., 2014).

2.2. Place Recognition

The *place recognition* problem is the problem of identifying regions in the environment that have been observed at distinctly different times during data acquisition and estimating the spatial relationships between their reobservations. Generally, place recognition can be used for identifying loop closures, global relocalization, and merging multiple overlapping datasets. In the *bentwing* 3D mapping context, the primary function of our place recognition algorithm is in merging multiple datasets, though it can also be used for loop closure detection in longer datasets in which the accumulated open-loop drift error is atypically large. In either case, the place recognition solution provides a coarse alignment that must be further refined by the non-rigid registration algorithm. A brief summary of the place recognition algorithm follows, with further details available in previous publications (Bosse & Zlot, 2009b, 2013). Figure 3 presents an illustrative example of the main steps in the place recognition framework.

The general approach taken in our place recognition solution is to divide the point cloud into a discrete set of *places*, which are defined as a collection of spatially and temporally contiguous local measurements. The term local refers to the requirement that the place should not include measurements taken over too long a duration such that the effect of drift could overly distort the appearance of the region (and in particular can affect multiple observations of that region in different ways). Each place is then represented by a set

of fixed-length descriptor vectors, each of which encodes the geometry of the neighborhood surrounding a keypoint selected from the point cloud. Places with many similar descriptors, identified through a k -nearest-neighbor voting mechanism, are postulated as being potential matches and then further verified based on geometric consistency.

For selecting keypoints and generating descriptors, a downsampled version of the point cloud is created by averaging measurements in a coarse voxel grid. This step removes local variations in the sampling density of the raw points that reflect the acquisition trajectory rather than inherent characteristics of the environment. Keypoints are then selected at random from the downsampled point cloud, though any that fall in locally planar regions are discarded due to the lack of saliency. (We have investigated more sophisticated methods for keypoint selection but have not encountered any that perform much better than this randomized approach.) The point descriptors are based on a 3D generalization of gestalt features (Walthelm, 2004; Bosse & Zlot, 2013), which encode the height distribution of points falling in radial bins surrounding the keypoint. As the KNN search assumes the similarity of descriptor vectors can be determined using a Euclidean distance, the descriptors are transformed according to a process similar to quadratic discriminant analysis (Bosse & Zlot, 2009b). The search for a given place proceeds by querying a kd -tree containing the global set of descriptors to find the nearest neighbors of each descriptor for that place. Each matching descriptor effectively casts a vote for the pair of places from which the descriptors originate, and the aggregation of the vote scores indicates the likelihood of place matches. Upon completion of the search for all places, vote scores over a threshold are retained and verified using a geometric consistency check based on RANSAC. The matches that remain are incorporated into a pose graph initialized with transformations between temporally adjacent places. A robust pose graph optimization algorithm, designed to be

tolerant of false positive links, is then applied to update the trajectory with the identified place matches (Zlot & Bosse, 2014b).

3. MECHANICAL DESIGN

To enable 3D perception of the local environment with a laser scanner that has an inherent 2D field of view, there are generally two options: allowing it to be swept across the environment in a push-broom fashion as the platform moves, or actuating it to generate a local 3D field of view independent from the platform motion. To deliver suitable input data for our trajectory estimation solution, the same parts of the environment must typically be reobserved within an approximate period of a few seconds, and therefore actuation is more practical than constraining the platform motion to meet this requirement. As described in Section 2, repeated observations of surface patches allow the 6 DoF scanner motion and 3D point cloud of the surroundings to be estimated.

3.1. Passive Rotary Actuation

In the vast majority of robotics applications that include articulated lasers, the actuation is driven by one or more dedicated electric motors through mechanisms and gears that increase overall system complexity, cost, power usage, and weight (e.g. Droschel, Holz, & Behnke (2014a)). This observation motivates the investigation of hardware solutions that focus on passive actuation of the sensor payload. Previous results with the *Zebedee* handheld mapping system in a variety of applications have established the feasibility and success of passively driven motion for 3D scanning with 2D lasers (Bosse et al., 2012). Although passively actuated systems are prone to produce relatively arbitrary and non-deterministic motion patterns (as is the case for *Zebedee*'s freely swinging spring-mounted laser scanner), our SLAM solution has been demonstrated to be capable of handling the resulting sensor data. We developed a mechanism that passively actuates the laser scanner's continuous rotation around the vertical axis on the quadcopter-based mapping system. Analogous to *Zebedee*, whose sensor motions are driven by the amplified walking motion of the operator that carries it, with *bentwing* we chose to investigate the use of the airflow generated by the quadcopter's four rotors as an energy source for the motion of the laser. However, in contrast to *Zebedee*, no spring is used as it would result in unbalanced forces that would need to be compensated by the attitude control mechanism to maintain a level flight, increasing power consumption. To convert the downdraft into rotational motion, two arms with aerodynamic vanes are connected to the sensor payload (Figure 4a). The downdraft acts on these vanes to generate a tangential force that propels the rotation. To maintain a desirable spinning mo-

tion, the vanes' angle of attack with respect to the airflow can be adjusted with two small servo motors in the center of the rotating unit that turn the two arms. A higher angle of attack leads to a higher tangential force, accelerating the rotation. Similarly, decreasing the angle of attack results in the rotation slowing down.

The sensor payload consists of a Hokuyo UTM-30LX-F 2D laser scanner (100 Hz scan rate, 270° field of view, 0.65° angular resolution, 210 g mass) and a MicroStrain 3DM-GX3-25 inertial measurement unit, rigidly coupled and mounted under the quadcopter frame. The payload rotation is enabled by the use of a single ball bearing that provides the mechanical connection between the sensor unit and the vehicle. A 12-wire slip ring provides electrical connections for power and data transmission between the sensor unit and the quadcopter platform. We manufactured most of the structural parts of the sensor assembly out of ABS plastic on a 3D printer. The booms that connect the vanes to the payload are made from very stiff and lightweight carbon fiber-reinforced plastic. A first prototype of the sensor payload was mounted under a small Ascending Technologies Pelican MAV to verify that the unique method of spinning the sensor unit works as expected and does not interfere with the quadcopter's normal flight behavior and dynamics. The initial vanes were based on a simple winglike design (Figure 4). These vanes in the downdraft of the four rotors provide sufficient tangential force to support a steady rotation of the sensor unit at a rate of about one to two revolutions per second while in hover flight. The rotation rate can easily be altered with different settings of the vane angles, controlled by the servo motors. As long as the spinning payload is well-balanced in the sense that its center of mass and the axis of rotation coincide it does not noticeably affect or destabilize the flight of the MAV.

After successful proof-of-concept testing, the system was migrated to a larger Skybit Systems Eagle MAV with sufficient payload capability to carry all the hardware components needed for 3D data acquisition (Figure 5). This additional hardware consists of a computer to log the sensor data and a microcontroller to connect the onboard computer to the servo motors, LEDs, potentiometers, and other peripherals. The single board computer we use for logging is an Intel Atom based system that runs Ubuntu Linux 12.04 and robot operating system (ROS) middleware. In the field, we can communicate with the logging computer through a WiFi-based interface to control the data collection process. The microcontroller is an Arduino UNO that is connected to the computer via a serial interface and to low-level peripherals through its digital and analog I/O-pins. Through the firmware on the microcontroller and a specific driver on the logging computer, the Arduino is seamlessly integrated into the ROS system. Besides data logging, the system also runs custom self-monitoring software and displays its status through a pair of LEDs. This feature enables the pilot to

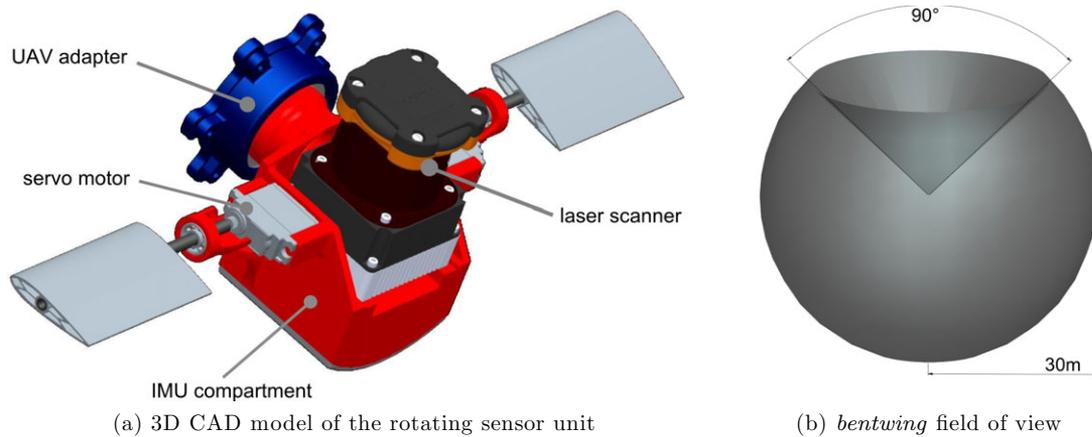


Figure 4. (a) Rendering of the 3D CAD model of the rotating sensor unit. A ball bearing and a slip ring allow rotation between the MAV adapter (blue) and the main body (red). One servo motor on each side controls the angle of the vane with respect to the airflow coming from the MAV. The IMU is mounted in an enclosed compartment under the laser scanner. (b) Illustration of the field of view of the *bentwing*-mounted rotating laser scanner. Oriented with the center ray pointing down, rotating around the vertical axis, and scanning up to a range of 30 m, the laser can cover an approximately spherical sector of up to 30 m in radius with a 90° blind cone above the sensor. Note that the field of vision is not exactly spherical due to the laser mount’s eccentricity.



Figure 5. The prototype of the sensor assembly and other hardware necessary for data acquisition mounted under the payload section of a modified Skybit Systems Eagle quadcopter system.

ensure that the data-logging system is fully operational before a flight is attempted, an important feature for efficient operation in the field.

3.2. Design Evolution

Although we were able to successfully acquire 3D data with the first version of the flying scanner, we did occasionally encounter reliability problems with the steady rotation and even stalling of the sensor unit, particularly in windy conditions and when descending. If the sensor rotation stalls for more than a few seconds, the 3D registration is likely to fail in this part of the recorded data, leading to misalignment in the generated map. To recover an undistorted map

from this data, the complete dataset could effectively be cut into two, discarding the data collected during the stall. The two parts of the dataset could then be merged thereafter (*i.e.*, aligned using place recognition and registered), leading to a consistent overall map. However, we had not implemented this solution at the time of the field trials, and thus focused on reducing the occurrence of sensor stalls during data acquisition. In doing so, we iteratively implemented several changes to the initial prototype setup, both in hardware and in the control software.

3.2.1. Arm Configuration

Observations of the airflow under the rotors indicate that it is relatively focused, so that virtually no vertical airflow acts on the two vanes when they are positioned in a gap between two rotors. In the initial mechanism design, the two arms supporting the vanes were spread at an angle of 180° (Figures 4a and 5). This configuration results in both vanes being in a low-airflow zone simultaneously, causing almost complete loss of propulsive torque four times during one rotation of the sensor unit. During some test flights conducted with the first operational setup shown in Figure 5, the rotation of the sensor unit stopped in such a position and could not reliably be restarted. In some cases, the payload rotation would not initiate after takeoff if the vanes were in the gaps between two rotors before starting the motors. To overcome these problems, we redesigned the mechanism with the arms angled at 135° as shown in Figure 6. In this design, the laser scanner position is shifted significantly off-center to counterbalance the eccentric weight distribution of the flaps. The angled configuration ensures that at

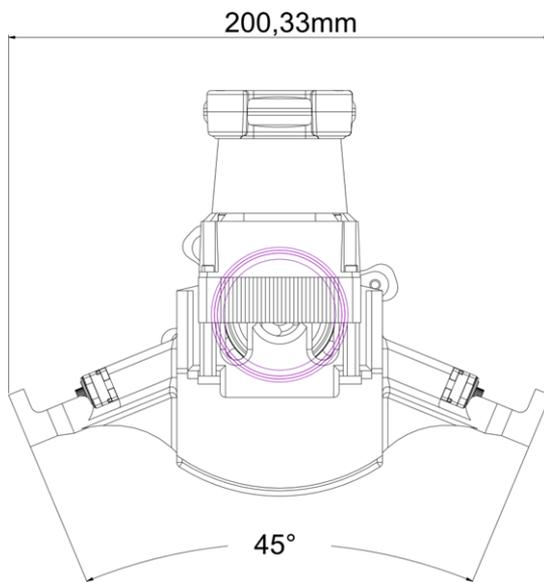


Figure 6. Bottom view of the redesigned sensor housing, spacing the two arms at an angle of 135° to ensure that at least one of them is affected by the rotors' downdraft. The main ball bearing is highlighted.

least one of the two vanes is in the zone of vertical airflow under a rotor at any time. At each position in which one vane is directly between two rotors and thus not affected by the downdraft, the other one is right under the center of one rotor disc. Through this design change, the reliability of the spinning motion was improved and it was assured that the rotation starts on takeoff, regardless of the mechanism's initial position. The angled configuration of the two wings partially inspired the name *bentwing* for this system. Bent-wings are also a native species of bats that roost in Australian caves, which is the other source of the name, as one intended application of the system is cave survey.

3.2.2. Vane Design

Despite the design change described in Section 3.2.1 and an increased reliability of the spinning mechanism, the rotation could still be stopped in flight by maneuvers or crosswinds that apparently significantly alter the aerodynamic conditions under the quadcopter. In addressing this problem, we designed and tested alternative vane concepts that make better use of the rotor downdraft and are less vulnerable to changes of the airflow. These changes include different airfoil shapes, both symmetric and asymmetric, as well as different arm lengths. The most successful vane designs include multiple small vanes per arm, enclosed in a common frame. The frame is asymmetrically shaped with a convex nose and concave back to take advantage of crosswinds in the same way that cup anemometers do. The

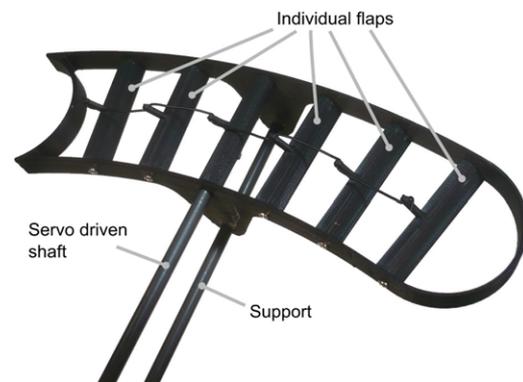


Figure 7. The individually actuated multiflap design contains five small vanes that can move individually from the enclosing frame. The angle of attack of all five vanes can be changed simultaneously without increasing the surface area that is subjected to crosswinds.

development of this vane type led from flaps that are rigidly coupled to the frame to a design that we call *individually actuated multiflaps* (Figure 7). The single vanes are replaced by five significantly smaller vanes that are spread out horizontally and are enclosed in an aerodynamically shaped frame. In the final design, the vanes are mechanically connected among one another and can move simultaneously to change their angle of attack, independent of their frame that always maintains a horizontal position. The position of all vanes in one frame can still be controlled by a single servo motor. This design serves two purposes: it homogenizes the effects of locally disturbed airflow under the copter through enlarging the effective area of the vane; and it exposes a significantly lower surface area to crosswinds than the initial, rather large, single flaps. Figure 2 shows *bentwing* in its final stage of mechanical design. The prototype payload assembly, including sensors and housing (615 g), multiflaps (94 g), logging computer (320 g), and microcontroller (63 g) has a total mass of 1.1 kg (not including batteries)⁵.

3.2.3. Rotation Speed Controller

Last, we implemented an automatic flap controller that enables the passively driven mechanism to maintain a relatively constant rotation rate even under the changing conditions that occur during different maneuvers and wind influences. Although our SLAM solution does not depend on a constant rotation speed, satisfactory results can be achieved more reliably when the rotation does not exceed a rate such that consecutive laser scanlines are too widely spaced

⁵The payload mass can be further reduced through design optimization and alternative computing platforms (the logging computer plus microcontroller mass can be reduced by a factor of three to 125 g with the components we are currently using).

(otherwise, measurements are too far apart to reliably estimate surfel parameters). Rotating the sensor payload too slowly could also be problematic, because the resulting slow scanning coverage of the surrounding environment would limit the acceptable flight speed of the MAV in order for the mapping solution to maintain convergence. A model-based derivation of the ideal range of rotation speeds is not straightforward to perform because it depends on a large number of parameters, some of which we can control (settings in our SLAM algorithm), some of which we can only partially control (flight of the MAV), and some that lie outside of our control (geometry of the scene to be scanned). However, with a very simple environment model, knowledge of exemplary SLAM parameters and basic assumptions about the flight, we can derive rough upper and lower bounds for the acceptable rotational speed of the scanner.

For the SLAM solution to operate reliably, the distribution of (matched) surfel normals over a processing window should span all three orthogonal coordinate axes of \mathbb{R}^3 with sufficient support. Over a given processing window, there is some minimum range ρ from the laser scanner in which the observable surfaces capture the variety of normal directions required. Assume that at distance ρ , there is a vertical wall surface that must be scanned at sufficient density to ensure that the span of surfel normals covers \mathbb{R}^3 . If the instantaneous sensor rotation speed at the time when this wall is scanned is α , the angle between consecutive scanlines intersecting this surface is $\alpha \cdot \Delta\tau$ where $\Delta\tau = 1/f_{laser}$, and f_{laser} is the laser scan frequency. For small angles ($\Delta\tau = 0.01$ s for *bentwing's* laser), the distance d between the scanlines at the surface can be approximated by the arc length at the given range and angle, $d \approx \rho \cdot \alpha \cdot \Delta\tau$. The value of d is dependent on the surfel size used in the online non-rigid registration algorithm and typically it is set to $d = 0.4$ m. With the other parameters selected or fixed, the inverse relationship between α and ρ represents a trade-off in flight path constraints: The faster the sensor payload is rotating, the closer the vehicle must generally be to the surfaces being scanned. We assume a conservative distance to maintain between the copter and the relevant surfaces (in a sparse environment) is around 5 m, which results in an upper bound value for the rotation speed $\alpha_{max} < 458^\circ/\text{s}$. Figure 8 illustrates the typical range measurement distributions in three environments, verifying that a significant proportion of the observed structure is maintained within 5–7 m of the platform. The upper bound would need to be reduced accordingly when intending to fly the platform at a longer range from, or higher altitude with respect to, the primary structures.

The trade-off between rotation speed and scan distance identified above suggests that maintaining the scanner rotation rate as low as possible would be the ideal scenario. However, in order to ensure the local environment is scanned more than once within a data processing window, typically 5 s in duration, the minimum average rotation rate would $\alpha_{min} > 180^\circ/5 \text{ s} \approx 36^\circ/\text{s}$ (a full sweep of the environ-

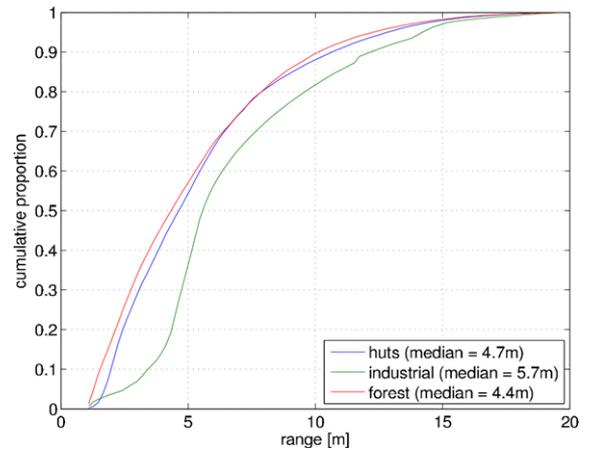


Figure 8. Empirical cumulative distribution functions of laser range values as measured in three different environments: a set of huts on Peel Island (Figure 1); an industrial compound (Figure 19a); a forest (Figure 13a). The median range value is also indicated for each environment.

ment is covered in 180° due symmetry in the scanner mount configuration). A value above this lower bound ensuring sufficient scan overlap is the absolute minimum, but for reliability purposes, it is more pragmatic to ensure at least two scanner rotations within a processing window, thereby doubling the minimum average rotation rate to at least $\alpha_{min} = 72^\circ/\text{s}$. Further taking the vehicle flight speed into account, and assuming flight speeds between 1 and 2 m/s are reasonable, we also need to ensure that reobserved surfaces are still within an acceptable range of the scanner to ensure adequate scan density to generate surfels. The minimum two scanner rotations achieved during a single processing window at the lower bound α_{min} above further suggests that, at the upper end of the desired speed range, the copter can move up to 5 m with respect to a surface before reobserving it, which is beyond our ideal observation range. Therefore, a more appropriate lower bound for rotation speed around $150^\circ/\text{s}$ is more realistic, though in terms of control, we choose a more conservative value of $180^\circ/\text{s}$.

To summarize, given the typical parameters of the SLAM solution, desired scanning distances of 5 m, and flight speeds up to 2 m/s, the sensor payload rotation rate should be maintained in the range $[150^\circ/\text{s}, 460^\circ/\text{s}]$. In practice, the mechanical system does not operate at the upper limit of this range—based on more than 50 flights, the median and maximum rotation rates are around $250^\circ/\text{s}$ and $350^\circ/\text{s}$, respectively—and the environments tend to be more structured than the bare minimum case assumed above, thus allowing some flexibility for the operator to increase maximum scanning distances from surfaces to 8–10 m, or flight speeds exceeding 2 m/s where appropriate. It is also possible to quantitatively analyze the current distribution

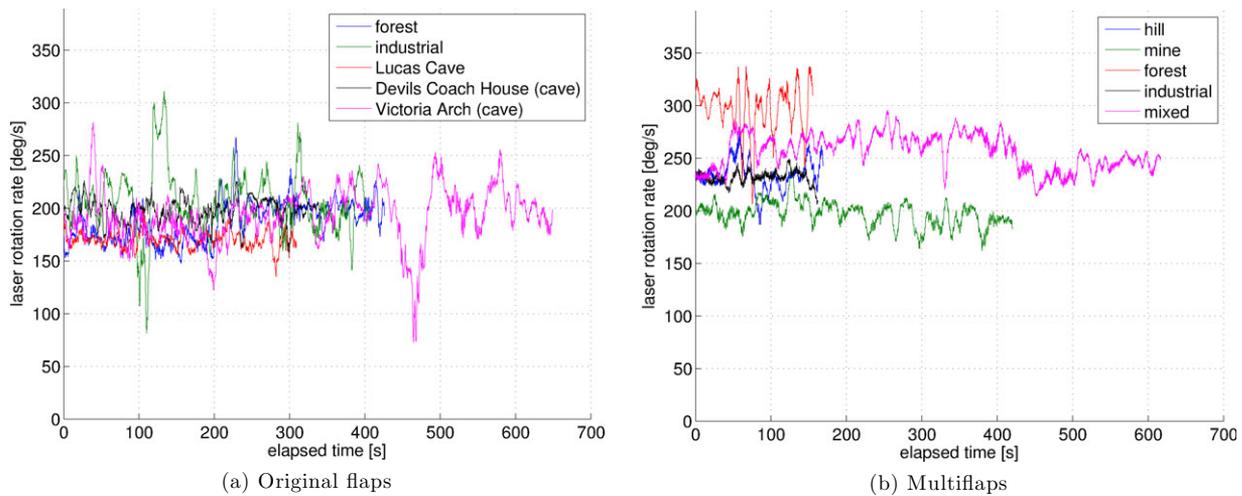


Figure 9. Measured laser spin rates from mapping datasets in various environments. Plotted are the rotation rates measured by the IMU averaged over a 5-second window. Mapping results from several of the datasets are presented in Section 4. (a) Results from five datasets using the original flap design. Total duration of the five datasets is 35.5 minutes. (b) Results from five datasets using the multiflap design. Total duration of the five datasets is 25.4 minutes.

of surface normals, and an onboard controller could adjust the flight behavior based on the available structure in the observable environment. Figure 9 shows the laser scanner rotation rates observed in mapping datasets in different environments (most of which are examples from Section 4). The rotation rates tend to be fairly uniform (within $150\text{--}210^\circ/\text{s}$) for the original flap design; however, the configuration is more susceptible to crosswinds that can occasionally reduce the rotation rate below the desired range (though not over a long enough duration or under conditions that caused any mapping errors). Higher variability generally occurs in the multiflap configuration, as here the pilot had the ability to remotely adjust the desired rotation speed and adapted it for each flight individually, according to wind speed and structure of the environment. This functionality was not available in the early flights with the original flaps, and a constant desired speed was used in the controller. In all of the multiflap flights, the rotation rates are maintained within the acceptable range, most importantly never dropping below $160^\circ/\text{s}$ in the 25.4 minutes of flight depicted.

The automatic flap controller maintains the sensor payload rotation speed within the acceptable range without requiring the pilot's active intervention. The implemented controller receives a measured rotation rate from the inertial sensor that spins with the laser. The desired speed can be commanded by the operator on the radio transmitter, in the range from 180 to $540^\circ/\text{s}$. There is also the possibility that the rotation speed target can be automatically generated based on an estimate of the visible structure in the environment (e.g., the distribution of surfel normals), though that level of processing is currently not implemented on the

flight or logging computers. Environments rich in features can support faster, and thereby more steady and reliable, rotations when scanning; whereas, very open terrain necessitates a slower rotation to ensure dense laser coverage and maximum performance of the registration algorithms. Based on the actual speed and the set speed, the control algorithm computes suitable vane angles and derives control commands for the servo motors. The control algorithm consists of three individual terms: a map-based controller, a proportional controller, and a nonlinear stall prevention mechanism (Figure 10). The map-based controller computes an angle command proportional to the desired rotational speed with gain k_m , while the proportional control computes an angle command proportional to the difference in desired speed and set speed with gain k_p . The stall prevention constantly monitors the actual rotational speed. If the rate falls below $180^\circ/\text{s}$, the controller issues an angle command of k_{sp} . The three control commands are summed, constrained to feasible values, and then issued to the servo motors through the Arduino microcontroller.

The three controller parameters k_m , k_p and k_{sp} require initial manual tuning. The value for k_{sp} is set to 45° , which results in the flaps being set to an angle of 45° as soon as the rotation rate falls under the stall threshold of $180^\circ/\text{s}$. Consequently, the flaps are also set to this very high angle as long as the copter is on the ground, which results in a very reliable spinning-up of the sensor unit as soon as the rotors are started for takeoff. The map-based controller gain k_m can be set via a sliding switch on the pilot's radio transmitter. A laptop computer with a ROS-based ground station software and a wireless uplink to the MAV's onboard ROS-based computer systems is used to monitor and record data

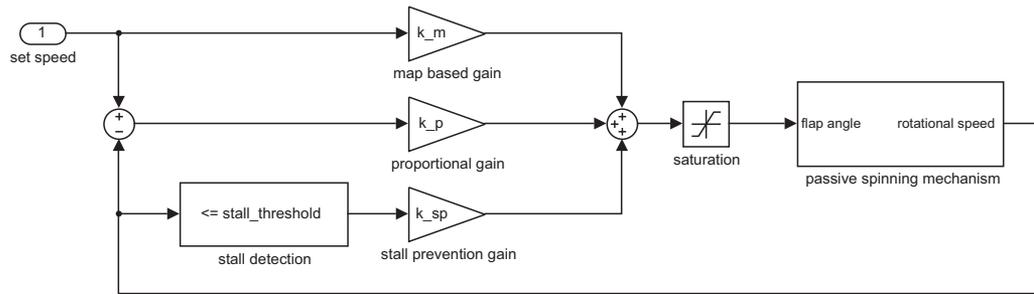


Figure 10. Block diagram of the rotation speed control algorithm. The three control actions, map-based, proportional, and stall prevention control are summed. The constrained result is issued to the servo motors that actuate the vanes.

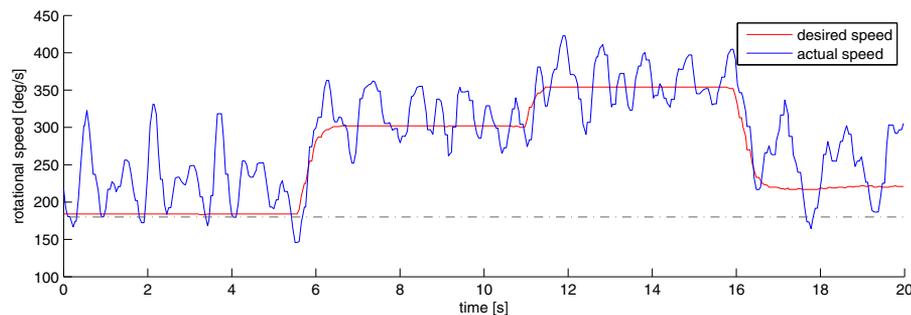


Figure 11. Evaluation data for the automatic rotational speed controller. The red curve shows the desired speed set by the pilot and the blue curve depicts the actual speed of the sensor rotation. Data was captured on the remote ground station during an outdoor flight. It is notable that the actual speed spikes after it falls under $180^\circ/\text{s}$ due to the stall prevention mechanism.

such as the sensor unit's rotational speed while in flight. While observing the rotational speed of the airborne sensors on the ground station, the k_m gain can be set so that it tracks the reference value reasonably well within the range of 180 to $540^\circ/\text{s}$ under no-wind conditions in stationary hover flight. The proportional controller is needed to compensate for changes in the airflow caused by crosswinds, headwind, and flight maneuvers. The controller gain k_p can also be set via the radio transmitter. It is tuned so that the desired rotational speed is tracked reliably under changing conditions while at the same time the actual rotational speed does not oscillate to an extent that decreases quality of the recorded 3D scans. Figure 11 shows 20 s of data recorded during an outdoor flight under windy conditions in normal scan operation with properly tuned controller parameters. The desired speed shown in red is commanded by the operator on the radio transmitter. The actual speed shown in blue is the rotational speed of the sensors as reported by the IMU. The plot reveals that the actual speed follows the desired speed, but oscillates around the set value. It can also be seen that the actual speed increases dramatically each time it falls under $180^\circ/\text{s}$ due to the stall prevention action of the control algorithm. Neither the oscillations nor the spiking have a negative effect on the mapping results. The actual rotation speed is confined within the interval between 150

and $450^\circ/\text{s}$, which is appropriate for collecting suitable data for our 3D mapping method.

3.3. Drag-Induced Power Consumption

To determine the power consumption induced by the drag of the individually actuated multiflaps, a stationary hover flight at about 4 m of altitude was performed with and without mounted flaps. An additional weight with the same mass as the flaps was attached to the MAV for the flight without flaps to ensure that any differences in power consumption relate solely to the drag of the flaps. The power consumption was recorded with the Eagle's own sensors that have a resolution of 100 mA for the current and 100 mV for the battery voltage. Figure 12 shows the Eagle's total power consumption for a 10-second segment of both flights. The results indicate only minor differences between the two scenarios. The mean values differ by about 4 W, which is less than 1% of the overall magnitude. Given the low resolution of the measurements, the sensitivity of the power consumption to wind influences and the possibility that the hover was not exactly stationary, it is hard to draw a definitive conclusion on the statistical significance of these results. However, it is evident that the drag of the flaps

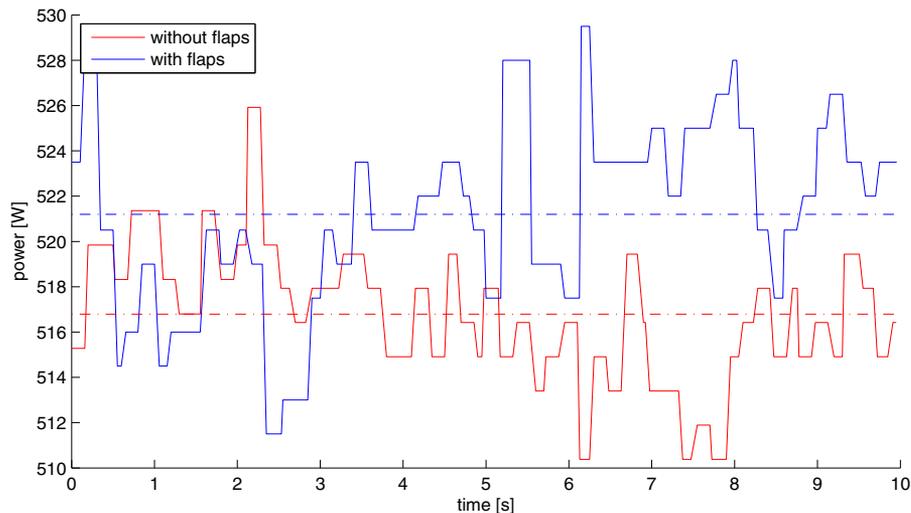


Figure 12. Power consumption of the Eagle quadcopter platform in hover flight with the actuated multiflaps mounted (blue) and with the flaps replaced by a dummy weight (red). The corresponding mean values are shown as dashed lines.

does not markedly increase the power consumption of the quadrotor platform.

3.4. Airspeed Experiments

With the final design of the spinning sensor payload, the system was successfully deployed under various conditions. The most important parameter that limits *bentwing*'s operability is the ambient wind speed. If the wind speed is too high or *bentwing* is flown at a high horizontal velocity, the rotation of the scanner occasionally stops. The rotation can be recovered quickly enough to avoid disrupting the collection of feasible data by a brief increase of motor thrust in most of these cases. But in some cases under strong wind influences, the stall can last more than a few seconds and thereby essentially end the data collection. To quantify the airspeed limitations under which the described passive actuation method can be deployed, we conducted two different types of tests. In the first set of tests, the MAV with spinning sensors is flown directly in the horizontal airstream of an industrial grade fan, thereby facing crosswinds much faster than wind speeds under which a scan flight could safely be conducted. In this scenario the rotation of the scanner did not stop and was not even noticeably affected by the artificial wind.

In the second set of tests, the MAV is securely mounted on two vertical booms about 1.8 m above an electric multipurpose ground vehicle. The four motors of the copter are powered and the ground vehicle is driven on a straight path while the speed of the headwind is measured with an anemometer. While we hypothesized this arrangement would better reflect flying conditions, the result of these experiments are ambiguous. During some of the runs, the sen-

sor system on the copter spun steadily even for the ground vehicle reaching its maximum speed and a measured headwind of 6 m/s. During a few other runs under seemingly the same conditions, the rotation stalled at measured headwinds as low as 3 to 4 m/s. As we have encountered stalls of the sensor rotation during flight trials at even lower airspeeds, the experiments seem to not accurately simulate the actual in-flight wind conditions. The stalls appear to be influenced by more complex factors, such as local wind disturbances caused by the local environment or quick changes in wind conditions; for example, when leaving the sheltered space between trees or during some of the MAV's maneuvers. Therefore, we cannot draw a general quantitative conclusion from these artificial scenarios as to the wind speed limit up to which the passively actuated sensor system can be deployed. Best results are achieved in low wind conditions and slow and smooth maneuvering.

4. MAPPING RESULTS

Over the course of several months, we fielded the *bentwing* aerial mapping system in a variety of applications and environments. Here, we first demonstrate the versatility of the system by describing selected results from some of these deployments in Section 4.1. We then present quantitative results from experiments evaluating the accuracy of the estimated trajectories and maps in Section 4.2.

For the presented experiments, the *bentwing* sensor payload was mounted on the Eagle quadrotor platform and used the 135°-angled vane arms. The vane design evolved from the simple wing shape to the multiflap version over the course of the deployments. While the earlier flap designs had a higher chance of stalling from crosswinds, these

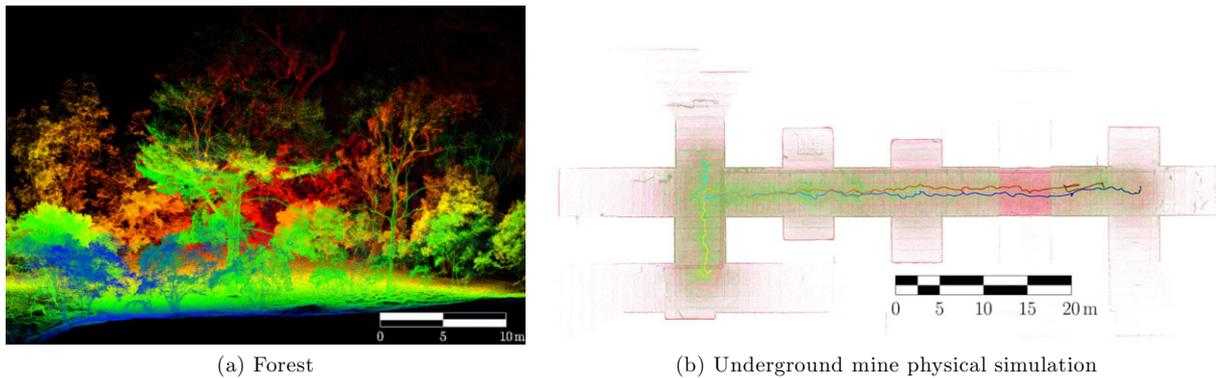


Figure 13. Results from *bentwing* in both natural and artificial environments. (a) 3D point cloud of a forested area containing eucalypt and pine trees. (b) Overhead view of a point cloud from a simulated underground coal mine. The *bentwing* trajectory is colored according to time (blue to red).

events were still quite rare and did not result in any significant problems in the field.

For the *bentwing* payload sensor configuration, the IMU is mounted on, and rotates with, the laser scanner. This design differs from the configuration deployed with spinning laser on a ground vehicle described in our previous publication (Zlot & Bosse, 2014b). The main reason for the change is that the quadcopter is constantly moving, which does not provide opportunity to re-estimate the IMU biases during periods where the platform is stationary. However, the *bentwing* sensor configuration allows for continuous IMU bias updates from the state estimation algorithm, in the same way it is performed for the *Zebedee* system (Bosse et al., 2012) as described in Section 2.1 (initial bias estimates are calculated during the stationary period before takeoff at the start of the dataset and updated via estimated corrections during flight). In fact, the software used for processing *Zebedee* data can be used unchanged for *bentwing* data, with the only minor modification being a parameter defining the physical dimensions of the platform to allow self-hit points to be filtered from the laser measurements.

The key parameters used for the continuous-time mapping algorithm are a 5-second processing window shifted by 1 second on each step; 0.4 m voxels as the base level of the multiresolution voxel grid, which has four levels (0.4, 0.8, 1.6, and 3.2 m voxels); and two sets of fixed view surfels containing 1.5 s of data, regenerated whenever the platform moves 1.5 m or rotates 120°.

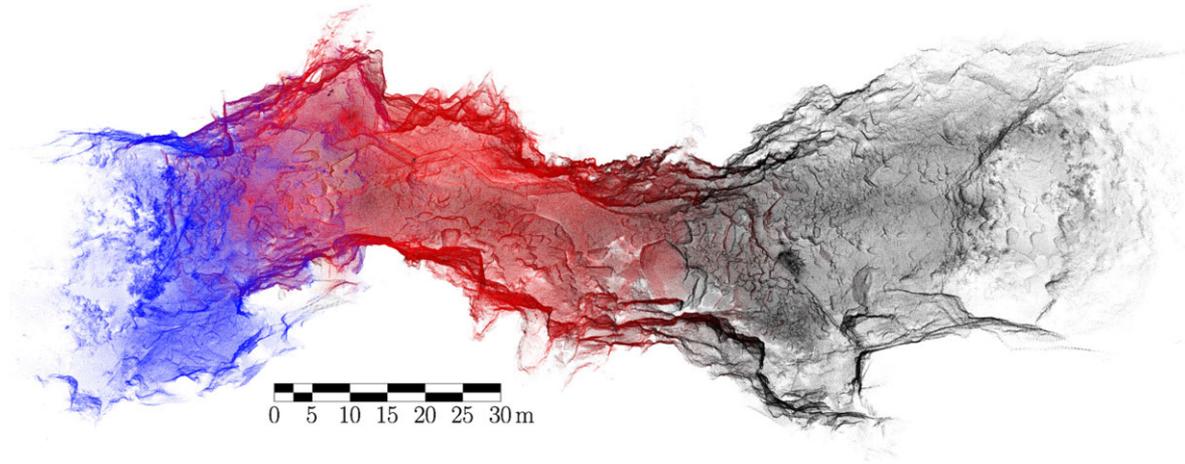
4.1. Demonstration of System Versatility

The mapping algorithms are designed to be general purpose, in that as long as there is a sufficient distribution of surface directions in the environment, it is not critical how structured those surfaces are. In particular, the system does not require planar geometry common in built environments and can operate across the spectrum between natural and ar-

tificial sites without any algorithmic or parameter changes. A range of sites along this spectrum are presented throughout this section and Section 4.2. Figure 13 illustrates this diversity with examples of results from a forest and an underground mine training facility.

The next results presented were captured in natural caves⁶. Our previous research includes the first instances of laser-based mobile mapping of caves (Holenstein, Zlot, & Bosse, 2011; Zlot & Bosse, 2014a), and we believe the present work represents the first aerial 3D mapping inside caves. Figure 14 depicts results from Victoria Arch at Wombeyan Caves in Australia. Victoria Arch is a large, open-ended chamber that is 30 m high in places and contains complex geometry that would restrict line-of-sight measurement from ground level. Three overlapping datasets are illustrated (merged with the place recognition algorithm), containing 49 million points and representing 27 minutes of flight time. Devil’s Coach House, illustrated in Figure 15, provides another example of a somewhat higher open-ended chamber at Jenolan Caves in Australia. The map of the cave illustrated here was covered by *bentwing* in two flights (12 minutes flight time, 23 million points in total), one primarily in a relatively low section (10 m height), and another in a more open section around 45 m in height. The high sections of these two caves proved somewhat challenging for the pilot, as it becomes more difficult to judge the proximity to wall surfaces

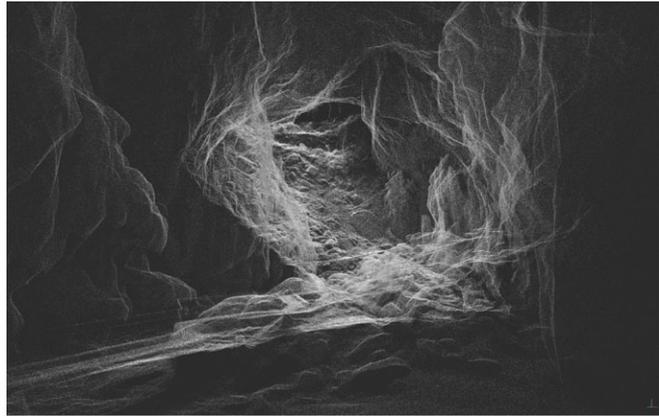
⁶The authors acknowledge that caves are delicate environments, formed over significantly long timescales. We took great care in approaching these sites and selected to fly *bentwing* only in large voids with limited speleothem growth to minimize the risk of damage from collisions and disturbance of materials. We worked very closely with cave management to ensure they were first comfortable with the technology, approved of the selected sites, and monitored the flights. We did not attempt any flights through narrow passageways or near delicate cave formations. Dust is also a consideration as disturbing significant amounts of settled dust can be damaging to the cave environment.



(a) Overhead view of Victoria Arch point cloud



(b) Photo of interior



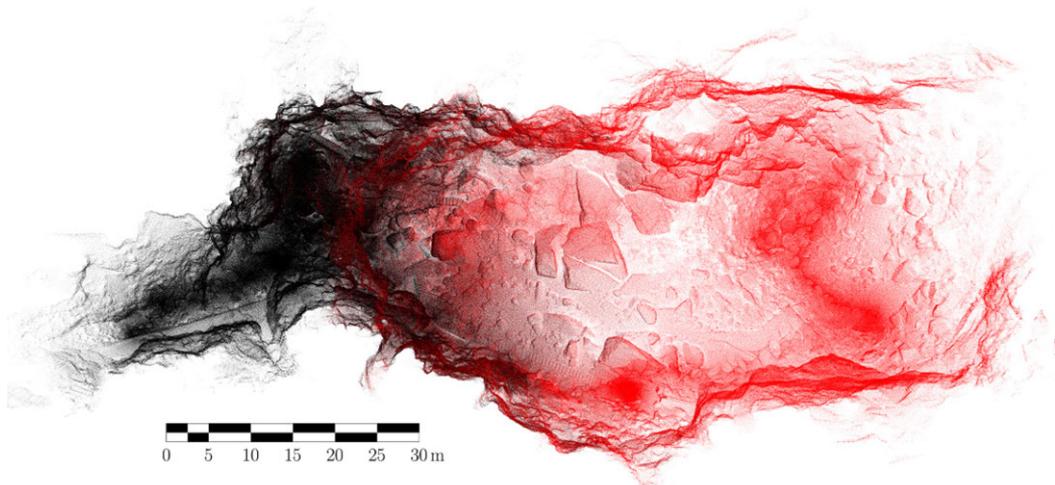
(c) Point cloud of interior

Figure 14. Results from Victoria Arch at Wombeyan Caves, New South Wales, Australia. Victoria Arch is approximately 120 m in length and 25–30 m high. It was captured by *bentwing* as three separate overlapping datasets, over a total of 27 minutes. (a) Overhead view of the *bentwing* point cloud, downsampled to 2 cm spatial density. Each of the three datasets is rendered in a different color. (b) A photograph of Victoria Arch facing due south (taken from the left side of (a), facing toward the right). (c) Rendering of the *bentwing* point cloud from approximately the same viewing location as the photograph.

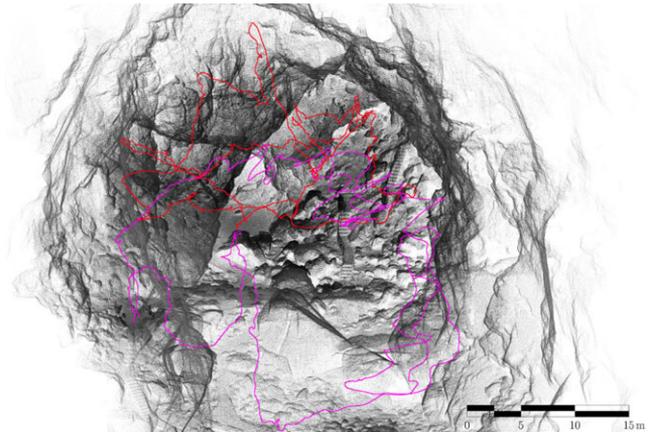
as the distance between the operator and aircraft increases. Figure 16 shows results from an interior chamber of Lucas Cave (Jenolan) called the Cathedral Chamber. The Cathedral has been scanned multiple times with *Zebedee* as part of scans of the larger cave; but, as is apparent from Figure 16a, the limited range of the laser scanner and occlusions from the surface geometry prevented the capture of the upper reaches of the chamber from ground level. Greater coverage of this part of the cave was achieved during a 5-minute flight with *bentwing* (Figure 16b). The void continues several meters higher, but the pilot restricted the quadcopter's altitude for safety reasons due to visibility limitations. The point cloud illustrated in Figure 16c demonstrates the

complementarity of *Zebedee* ground-based scanning and *bentwing* aerial scanning. Although *bentwing* was required to reach the upper areas of the chamber, the passages on either side of the Cathedral Chamber and throughout Lucas Cave (which continues on either side for hundreds of meters) are too narrow and delicate for the quadcopter but are suitable for an operator walking with *Zebedee*.

Further examples more clearly illustrate another benefit of combining ground-based and aerial scanning. Often there are high upward-facing surfaces, such as rooftops, that cannot be measured from ground level. In the scene from the Peel Island Lazaret, a historic former leper colony near Brisbane, Australia, shown earlier in Figure 1, the rooftops



(a) Overhead view of Devil's Coach House point cloud

(b) *bentwing* in the Devil's Coach House

(c) Point cloud and trajectory

Figure 15. The first *bentwing* cave mapping flights in the Devil's Coach House at Jenolan Caves, New South Wales, Australia. The Devil's Coach House is a large open-ended void approximately 45 m in height. (a) Overhead view of the *bentwing* point cloud of the Devil's Coach House. It was captured by *bentwing* in two separate runs composed of about 500 m of trajectory length over approximately 12 minutes of flight time. The rendered point cloud has not been downsampled, and as a result the scan/flight pattern is more apparent from density variations in the image (compare with Figure 14a). The two datasets are indicated in different colors. (b) *bentwing* in flight mapping the Devil's Coach House. (c) Rendering of the *bentwing*-generated map. The trajectories of the two datasets are indicated in two different colors. (The rendered view is not the same as the photo in (b)).

of various structures are not visible from ground level and thus are not included in the *Zebedee* scans. Conversely, viewing of some building walls, interiors, and other structures are not accessible from the quadcopter platform; but combined, the coverage is more complete. In the small marble quarry depicted in Figure 17, the upper surfaces of several of the benches cannot be measured from anywhere safely accessible by walking but are efficiently covered in a brief *bentwing* flight.

4.2. Evaluation of System Performance

To evaluate the accuracy of the *bentwing* trajectory, the quadcopter is tracked with a Leica Viva TS12 robotic total station to produce an independent trajectory estimate. The total station in tracking mode has a stated measurement position accuracy of 3 mm, with a typical measurement time of under 0.15 s. To enable tracking with the total station, a 360° mini-reflector prism (Leica GRZ101) is mounted on *bentwing*'s laser scanner. The quadcopter is flown in two environments:

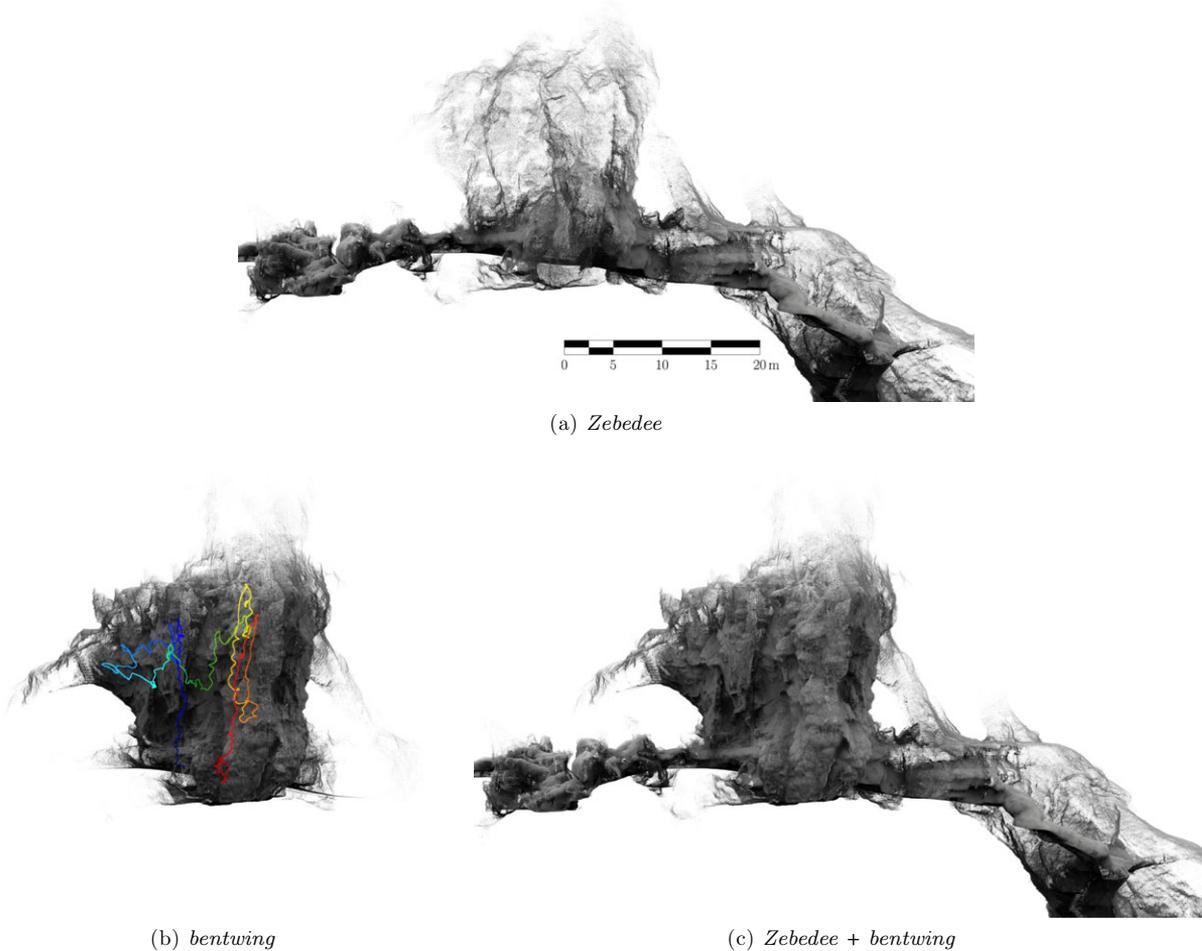
(a) *Zebedee*(b) *bentwing*(c) *Zebedee + bentwing*

Figure 16. External view of point clouds of Cathedral Chamber in Lucas Cave at Jenolan. (a) Cathedral Chamber with adjacent passages as captured on by *Zebedee*. The operator attempted to scan all visible surfaces with *Zebedee*; *i.e.*, the trajectory is not merely a straight walkthrough. Because of the limited range of the Hokuyo and line-of-sight factors, only the lower section of the chamber is densely captured. (b) The chamber captured during a 5.2 minute flight with *bentwing*. The quadcopter was flown to a maximum height of 20.6 m above floor level into several voids in the upper reaches of the chamber. The sensor trajectory, colored by time (blue to red), is superimposed on the point cloud. The chamber extends slightly farther vertically; however, because of range and visibility, the pilot assessed that it was not safe for either the cave or platform to progress any higher without additional protective features in place. (c) A combined point cloud showing data from both *Zebedee* and *bentwing* demonstrating the complementarity of the two sensing platforms (*Zebedee* can be used in the narrow passages while *bentwing* cannot; but the aerial platform is required to cover the upper reaches of high chambers in detail).

an industrial compound area with a concrete-surfaced courtyard surrounded by several large buildings (Figure 19a); and a small building on a grassy hill surrounded by a chain-link fence and vegetation (Figures 18 and 19c). The dataset durations are 2.8 minutes and 2.9 minutes, respectively, and flight speeds are typically between 1–2 m/s.

The tracking results are illustrated in Figures 19b and 19d. The differences between the *bentwing* trajectory and total station estimates are observed to be within a few

centimeters, with a smaller difference in the vertical compared to the horizontal direction. At first glance, if these values are interpreted as errors with the total station as ground truth, the results appear to suggest that *bentwing*'s positioning can be off by more than 5 cm. However, the total station is designed to measure stationary objects, and given the typical measurement time of up to 0.15 s, the instrument precision at the actual flight speeds can be on the order of tens of centimeters. This effect can be observed

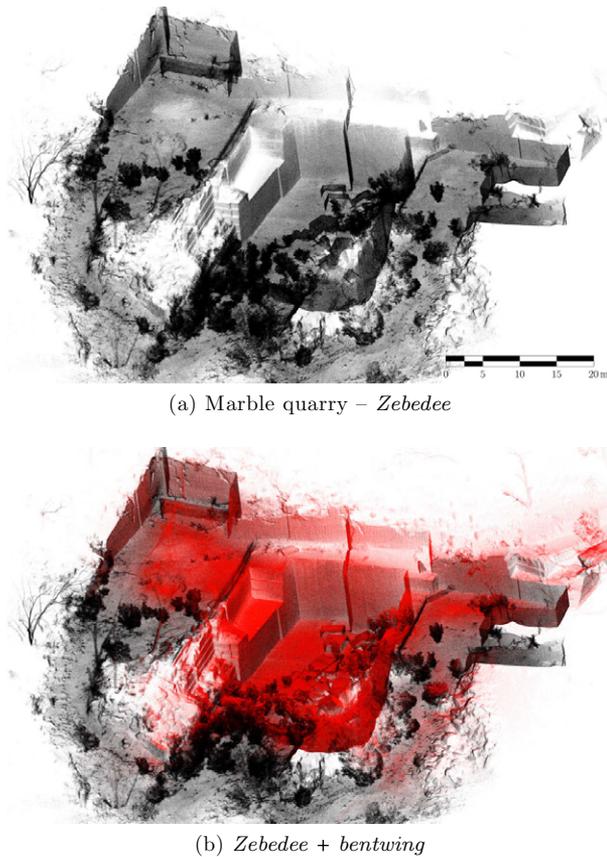


Figure 17. 3D scan of a former marble quarry. (a) *Zebedee*-generated point cloud. The high upper surfaces of the benches are not covered as they are not within line-of-sight of the ground-based scanner. (b) Combined *Zebedee* (black) and *bentwing* (red) point clouds. The high upward-facing surfaces are covered by data from a 7.6-minute *bentwing* flight to form a more complete scan of the site.

in Figure 20, where the positional measurement differences can be seen to be highly correlated with platform speed. As a result, the total station can be relied on to provide only coarse confirmation of the trajectory accuracy. That the biases are roughly zero-mean indicates that the error in the total station is roughly equally distributed in all directions with the given trajectories, though the typically slower vertical speeds result in tighter vertical distributions.

The *bentwing* system accuracy can alternatively be evaluated by focusing on the point cloud map produced. To assess the map quality, a comparison is performed between a point cloud generated from *bentwing* versus a point cloud generated from a Faro Focus 3D terrestrial laser scanner (TLS). The Faro scanner has a range bias of ± 2 mm and range precision of 0.6–2.2 mm (range, reflectivity, and configuration dependent) according to the manufacturer spec-

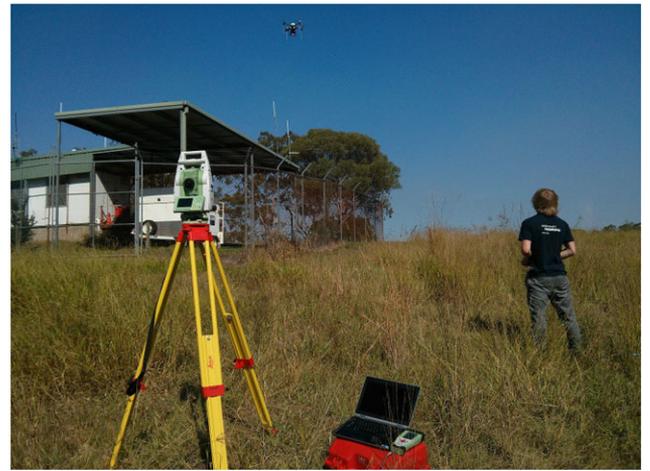


Figure 18. Setup for total station tracking experiments in the grassy hill environment. The total station is seen in the foreground and *bentwing* in flight above the building (top center).

ification, though the overall accuracy also depends on (to a smaller degree) how well calibrated the device is. The area selected for this comparison is the same open outdoor industrial compound considered in the previous experiments (Figure 19a), which has horizontal dimensions of approximately 50×35 m and building heights of 7.5 m and 11.5 m (Figure 21a). To capture this area, the TLS was set up to scan from six locations and the resulting point clouds were co-registered with a robust point-to-plane ICP algorithm (Figure 21b). The *bentwing* point cloud (Figure 21c) was acquired during a 6.8 minute flight at an average height of about 4.25 m and typical flying speeds between 1 and 2 m/s. Note that, due to availability of equipment, the TLS scans were acquired three months prior to the *bentwing* data. As a result, there are some objects in the scene that have been added, removed, or shifted.

Differences between the *bentwing* and TLS point clouds are illustrated in Figure 21d, where the TLS point cloud is taken as a reference. In the rendered map, each point in the *bentwing* scan is colored by the distance along the normal to surfels estimated from the TLS point cloud. Significant changes attributed to objects added or removed are filtered out by ignoring any errors larger than 30 cm (note that all such errors were due to physical changes in the environment). From the image, we can observe that the error is predominantly close to zero, with most of the major differences occurring along edges of objects that have been added or removed (the simple threshold classification described above does not eliminate errors at the boundaries of these objects). Point sizes have been increased when rendering this image to highlight the differences, but in some cases, this exaggerates the density of a few spurious negative errors (as these by definition are rendered in front of the other surfaces).

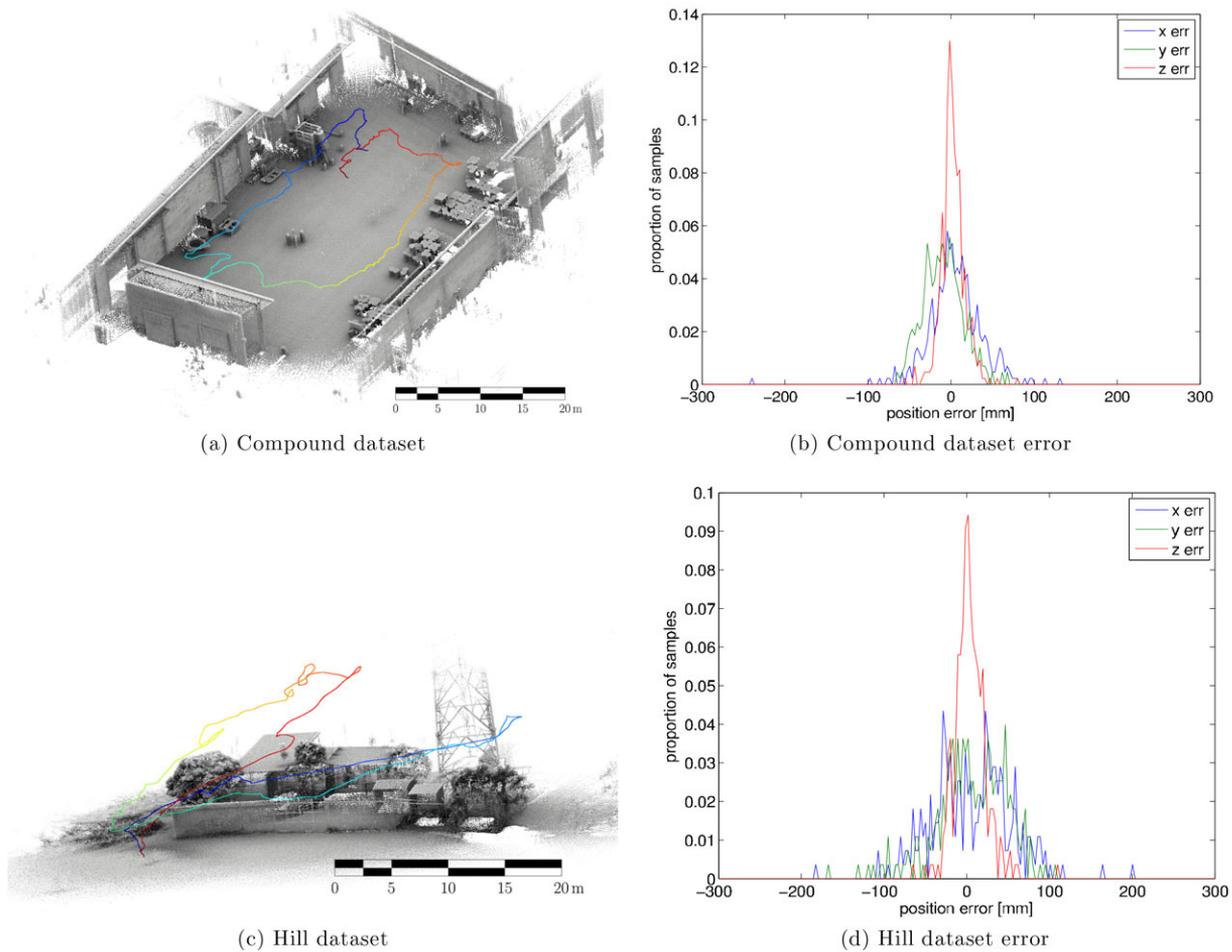


Figure 19. Comparison of the *bentwing* trajectory with the position estimated using a total station in two different environments. (a) *bentwing* point cloud map from an industrial compound area. (The mapped area is the same as the one covered in Figure 21 but collected at a different time.) (b) Distribution of differences between the *bentwing* and total station trajectories in the global x, y, and z directions for the compound area. (c) *bentwing* point cloud map of the exterior of a building on a grassy hill surrounded by a fence with trees and vegetation. (d) Distribution of differences between the *bentwing* and total station trajectories in the global x, y, and z directions for the hill area.

Areas with retroreflective tape (e.g. the “red” bollards at the center of the compound) also appear mismatched, but that effect is due to differences in how the two laser scanners deal with artificially reflective materials (the Hokuyo range measurement tend to be considerably noisier on these surfaces). The distribution of the errors are represented quantitatively in Figure 22. We observe that the overall error has little bias (close to zero mean), and a standard deviation of 1.4 cm. This result verifies that the *bentwing* point cloud accurately represents the environment (and that the visible errors discussed above truly are a small number of outliers), with the main differences between it and the TLS data occurring due to the lower precision of the Hokuyo laser scanner.

The point densities in the TLS and *bentwing* point clouds in the industrial compound area dataset are similar and illustrated in Figure 23. The densities are represented as distributions of point spacing, as determined by the nearest neighbor of each point. The median spacing in the TLS data is 8 mm, while in the *bentwing* data it is 10 mm. The point densities in the TLS data show less uniformity, with very high sampling close to the TLS stations and lower sampling farther away. This effect can be observed visually in the point cloud in Figure 21b, and also by the slight increase in mass above the tail of the TLS density distribution in Figure 23. As *bentwing* traverses through the environment continuously, it is likely to more evenly distribute its sampling as compared to the TLS scanner, which only takes

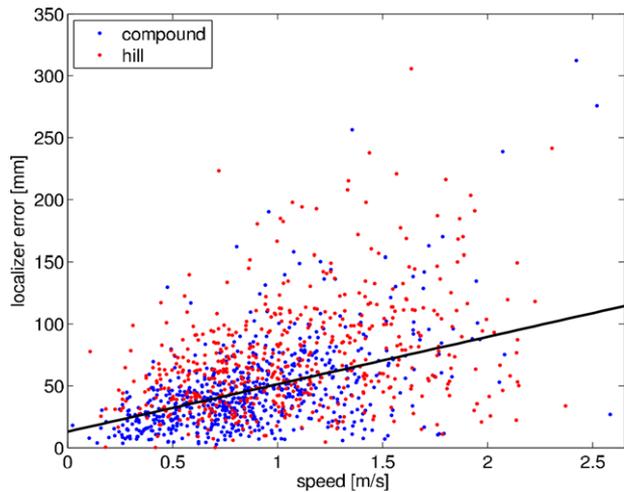


Figure 20. Distance error between the total station measurements and the estimated *bentwing* trajectory as a function of platform speed. The error is correlated with speed, as the total station is intended for stationary targets. A line of best fit to both datasets, based on robust least squares regression, is indicated in black.

measurements from a few discrete locations. In achieving similar average point cloud density, the *bentwing* flight at 6.8 minutes was considerably more efficient than the TLS acquisition, which required close to an hour to set up the equipment at the six stations and run the scan sequence.

The close agreement of *bentwing* with TLS point clouds and total station tracking is an indication of the accuracy of the closed-loop *bentwing* trajectory. Therefore, we can compare the open-loop trajectory, estimated in the first steps of the processing pipeline, to the closed-loop trajectory after global registration in order to quantify the amount of drift error. These observations provide an estimate of the drift that might be expected in cases where loop closures are not possible (or just not completed in the trajectory) in similar environments and flying conditions. For this comparison, we consider the drift that occurs after traversing a particular distance along the trajectory (e.g. 50 m), by aligning the open- and closed-loop trajectories at the beginning of the interval, and measuring the difference at the end. Statistics of the errors are accumulated for each DoF by moving a sliding window of the fixed segment length across the trajectory. The process is repeated for selected window lengths to illustrate the expected performance after traversing various distances.

The results of this analysis for two datasets, a built environment and a forest, are presented as a series of distributions in Figure 24. The built environment example is the same dataset used in the TLS comparison (Figure 21), and the forest environment is from the dataset depicted in Figures 13a and 25. These environments are sufficiently

large that the laser footprint covers only a small portion of the overall area at any given time, and thus the mapping algorithm is not simply matching against an initial map maintained in the processing window or fixed view surfels. The results for these environments are typical, and we observe similar performance on other datasets. In terms of translational drift (Figures 24a and 24b), we observe that the drift is unbiased in both the vertical and horizontal directions. We further observe higher accuracy in the vertical than in the horizontal, which is likely because there is nearly always dense laser data directly below the platform, and the vertical motions tend to be less extreme in speed and magnitude. Statistically, the translational drift is generally within 1 mm per meter traveled half of the time in the horizontal (based on the interquartile range), and an order of magnitude smaller in the vertical. The most extreme errors in the horizontal are around 5–6 mm per meter traveled and are slightly higher for the forest environment compared to the built environment. Over longer segment lengths, the variance is reduced as the random component of the error is to some degree canceled out.

For the rotational drift (Figures 24c and 24d), the error about the laser spin axis is significantly larger than in the other directions. This effect is largely because the platform orientation is typically upright with respect to the gravity vector, so the roll and pitch errors can largely be corrected based on this absolute reference in the IMU readings. Slight biases of a few thousandths of a degree are observed in the laser spin (platform yaw) direction, which is primarily due to residual errors in the laser to IMU calibration. This explanation can be verified by observing the reduction of these biases with improved calibration. The distributions of rotational errors are similar in the built and forest environments and are well within a one-hundredth of a degree half the time for the laser spin direction, an order of magnitude lower in the other rotational directions. The rotational drift is within a few hundredths of a degree for the most extreme observations of error in the laser spin direction (indicated by the boxplot whiskers over the shorter window lengths).

5. DISCUSSION

The aim of the presented work has been to develop a passively actuated mechanism to enable 3D perception with a 2D laser scanner on a quadrotor MAV and to use our existing SLAM solution to generate high-quality 3D models of the environment from the aerial scan data. This goal was achieved through the development of *bentwing*, which has been demonstrated to be a versatile and accurate small-scale aerial 3D mapping system. While the passive actuation approach can be advantageous compared to motorized designs—for example, in the case of the *Zebedee* design where a spring is lighter, simpler, and lower-power than a

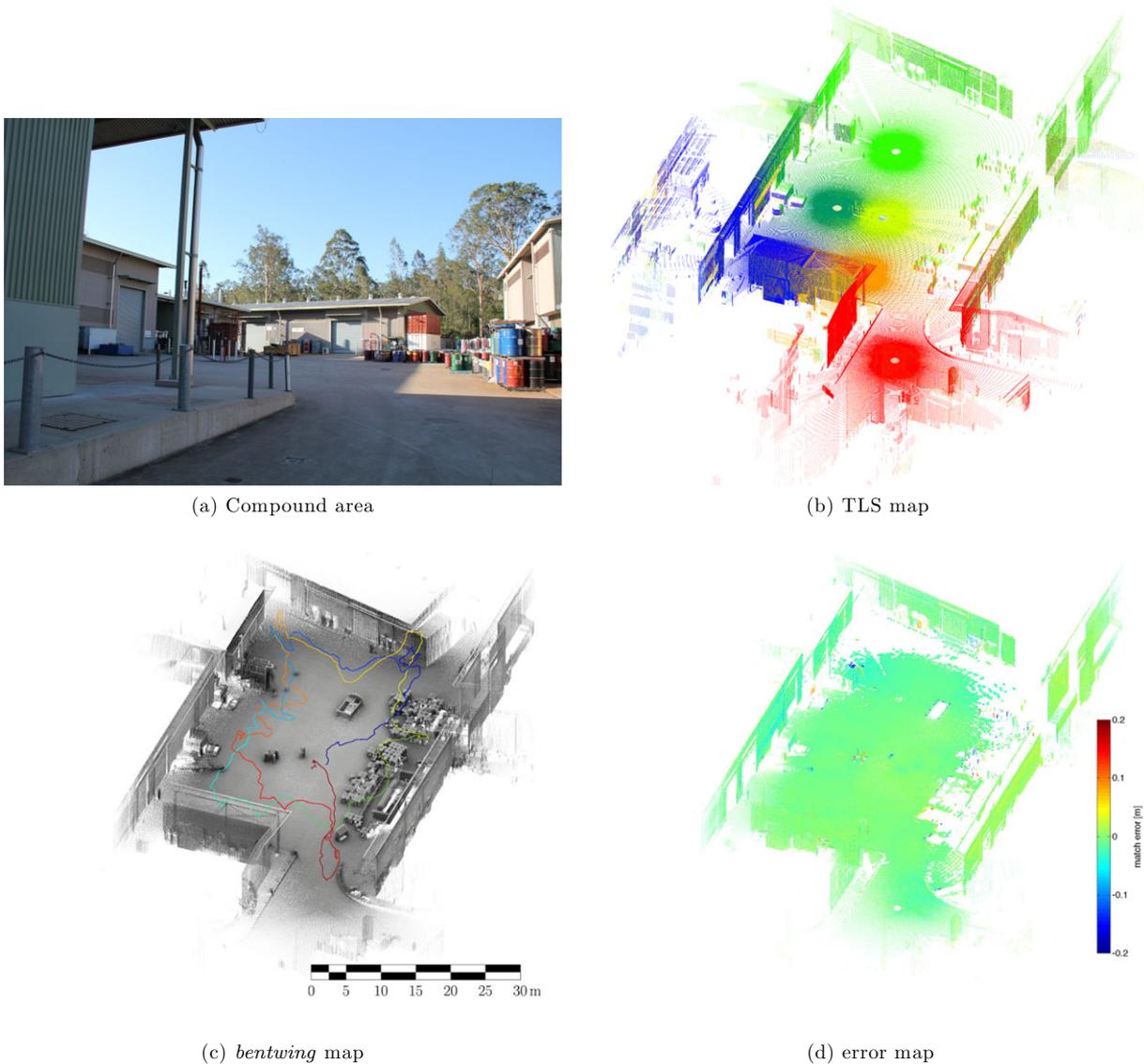


Figure 21. A comparison between *bentwing* data and terrestrial laser scan (TLS) data acquired with a Faro Focus 3D. Note that the data were acquired months apart, and various objects in the environment were moved between the scans. (a) Photograph of the compound area captured approximately from the center of the bottom-right (red) TLS scan in (b). (b) 3D point cloud of the compound composed of six stationary scans from a Faro Focus 3D TLS. The scans have been registered using the iterative closest point algorithm. Each of the six scans is rendered in a different color. (c) 3D point cloud map generated with *bentwing*, colored using an ambient occlusion shading routine. The trajectory of the quadcopter is indicated and colored according to time (blue to red). (d) The *bentwing* point cloud colored by error relative to the TLS data. Points from objects that have moved have been removed from the scan by thresholding the errors; however, some points near the boundaries of these objects still remain (and can be seen to have relatively high error values).

motor—those benefits could not easily and conclusively be replicated with the mechanism developed for *bentwing*.

The underlying concept of the laser scanner and IMU rotating as a sensor unit around the vertical axis under the quadcopter, driven by two vanes in the airflow, leads to

a simple and elegant design. This basic design, where the vanes are at a fixed angle and not actuated, does not need any motor, control, or additional power supply to produce a rotation that generates a large 3D field of view for the laser. Tests with an early prototype on a relatively small

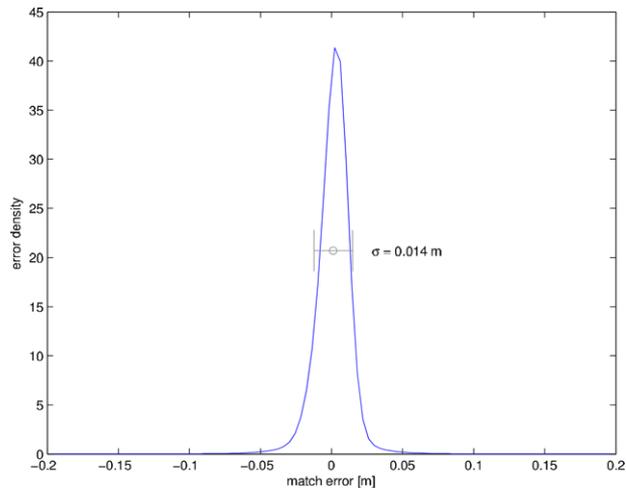


Figure 22. Distribution of errors of the *bentwing* point cloud with respect to the TLS scan data. Most points from objects that have moved have been ignored by thresholding the errors. Some points near the boundaries of these objects still remain as a result of this simple thresholding technique, which may contribute to inflate the error statistics.

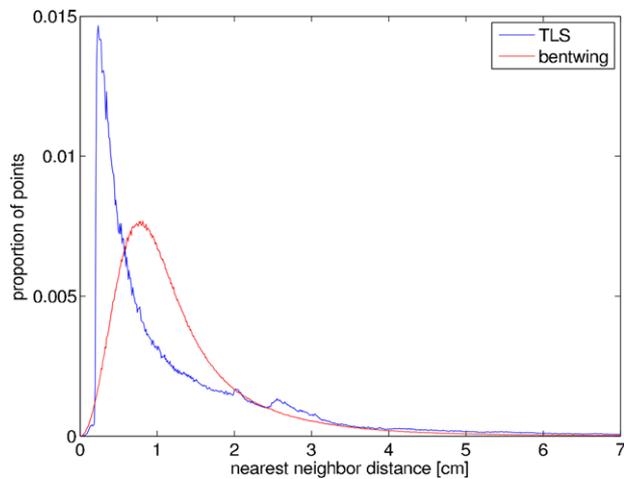


Figure 23. Density comparison between the TLS and *bentwing* point clouds from the compound area dataset. Distributions of the point spacing (nearest-neighbor distance for each point) for each method are illustrated.

quadcopter platform had indicated the potential of a mechanism based on this concept. These tests had also shown that this relatively large spinning payload does not noticeably affect the MAV's flight behavior other than through the added mass. However, two factors prevented the basic design from being deployable for general scanning operation. To carry all the components necessary for scanning, we required a larger quadrotor platform with larger gaps between the rotors. These gaps present a problem to the simple downdraft-

driven mechanism, as the airflow in some regions can be significantly affected by influences other than the rotors' downdraft. Furthermore, the rotation speed of the sensors cannot be arbitrarily high for the laser scanner to sample the 3D surroundings with high enough density suitable for our registration algorithm. Although it is difficult to define an ideal rotation rate for optimal sampling density because it depends on a large number of parameters associated with the MAV's flight, the type and scale of surroundings and internal settings of our SLAM solution, we generally achieve best results with a rotation rate close to 1 Hz. As the rotation speed of the sensor decreases, the influence of its inertia to counteract disturbances to the actuation also decreases, and the rotation becomes more prone to stalling. Our results have shown that a mechanism without active control can support a reliable rotation at a rate of approximately 2–3 Hz with appropriately set vane angles. At these relatively high rotational speeds, even moderate wind gusts or quick descents do not cause the sensor motion to stall. In this case, the rotation rate is determined by the airspeed of the rotors' downdraft, which is approximately constant since a scanning flight is typically a smooth hover. For slower rotations at rates close to 1 Hz, reliability is reduced and stalls become more likely to happen with the basic design. For these two reasons, we had to alter the design from the initial prototype and add a feedback control loop to dynamically adjust the vane angles, keeping the speed of the rotation in the desirable range. Although we could show that the so achieved passively actuated sensor unit is able to reliably deliver data suitable for high-quality 3D scan registration, the fact that we used two small servo motors and a relatively complex vane design make the advantages over a directly motor-spun solution in terms of elegance and simplicity debatable.

The passive approach presented here could be preferable to a motorized version if it did not rely on the servo motors to constantly adjust the vanes' angle of attack. In this case, there would be a chance that it could be lighter and simpler than a conventional actively actuated design. This goal could be achieved by a novel vane design that guarantees a slow, dependable rotation without the need of feedback control to compensate for changes in the airflow. However, we have not identified such a design yet. As mentioned earlier, a faster, reliable rotation on the order of 2–3 Hz can be achieved more easily in open-loop control (*i.e.* with fixed vane angles), than with the slower rotation rate (180–460°/s, or 0.4–1.3 Hz) required for the 100 Hz Hokuyo scanner. The use of a laser sensor with a higher scan rate would allow for faster rotation of the sensors while still being able to perform sufficiently dense point sampling. With the increasing popularity of MAV platforms, a larger variety of lightweight laser scanners are being introduced into the market, and it is conceivable that a suitable higher scan rate device could be available in the near future. Alternatively, a heavier rotating payload might also be worth considering, as the increased inertia would likely reduce the impact of

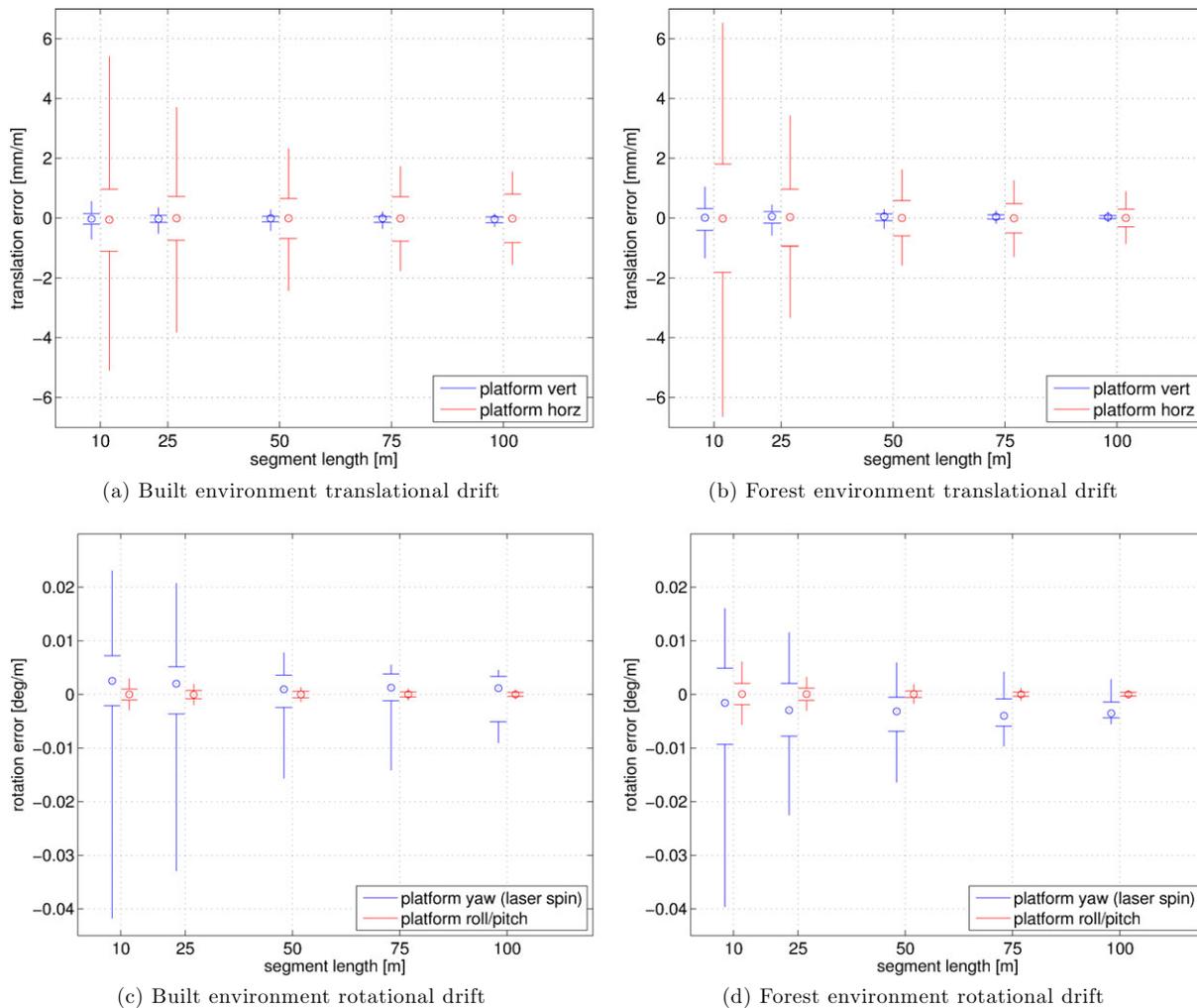


Figure 24. Distributions of drift error for the *bentwing* trajectory at various trajectory segment lengths visualized as boxplots. For each segment length, a sliding window is moved along the trajectory with the open- and closed-loop solutions aligned at the start and the accumulated error measured at the end. The measurements are expressed with respect to the platform (laser) frame. The horizontal translational and rotational axes have been combined because the platform is holonomic in the xy -plane. The circles in the centers of the boxplots are the median values, with the central region between the horizontal bars representing the interquartile range and the whiskers the 3rd and 97th percentiles. (a) Translational errors for the built environment (compound area pictured in Figure 21) as millimeters of error per meter traveled. The trajectory length is 324 m with maximum altitude of 8.8 m, and the typical platform speed between 0.6–1.3 m/s. The flight time is 6.8 minutes. (b) Translational errors for the forest environment (pictured in Figure 25) as millimeters of error per meter traveled. The trajectory length is 206 m with maximum altitude of 14.1 m, and typical platform speed between 0.5 and .1 m/s. The flight time is 5.2 minutes. (c)–(d) Rotational errors for the two environments as degrees of error per meter traveled.

crosswinds. The higher weight could be realized without increasing the system's overall weight by including the data-logging computer into the rotating payload.

There are several interesting lessons that we learned in the course of the design and research work with *bentwing* that will also find practical applicability in related research projects. As our existing SLAM solution was used

essentially unmodified, the key insights gained are mainly relevant to the mechanical and system design aspects. Limitations of the overall mapping system are discussed subsequently in Section 5.1.

Preventing pollution of mechanical parts. The airflow-driven actuation mechanism for the spinning sensors relies on low friction in the main bearing that allows the

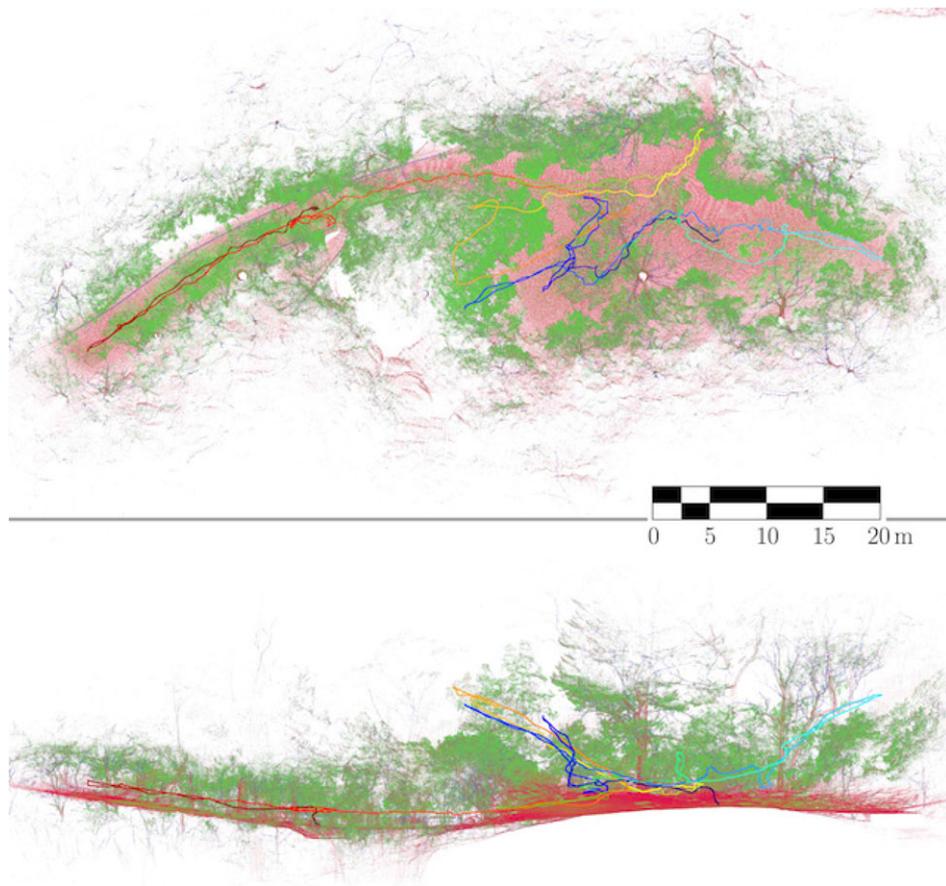


Figure 25. Point cloud and trajectory for the forest environment used for drift error analysis in Figure 24. The upper section illustrates the point cloud and trajectory from overhead, and the lower part shows a side view. The point cloud is colored according to local shape and the trajectory according to time (blue to red).

rotation. Any friction in this bearing causes the spinning mechanism to exert a torque on the MAV around its yaw axis. This torque can potentially change the flight behavior of the quadcopter or even cause undesirable yaw drift. To keep friction as low as possible, we chose an unsealed single row ball bearing and replaced the standard grease with very thin oil. The missing sealing rings facilitate intrusion of dust and other sources of contamination into the bearing that increase friction. We encountered this after repeated takeoff and landing on the very dusty surfaces of a sand island and in some of the very dry arch caves. The parameters of the vane control algorithm had to be adapted to the pollution level to let the mechanism achieve a reliable rotation at the desired speed. Future designs will have to incorporate suitable sealings that do not increase friction but provide sufficient shielding from dust and other pollution.

Compliant landing gear. The standard landing gear of the Eagle quadcopter platform had to be significantly

altered to provide space for the sensor payload with its rotating arms. The initial modified landing gear consisted of four stiff aluminum tubes, one mounted directly under each of the four motors (Figure 5). As it is not always possible for the platform to land perfectly level and with no horizontal velocity, the long and stiff legs exert a large torque on their adapter parts and the airframe in general on ground impact. After several imperfect landings, one of the adapters that connect the airframe and the landing gear legs broke. We were able to redesign the landing gear by changing the material in order to prevent this problem. With the four legs made from very flexible but durable, thin glass-fiber-reinforced plastic rods, the landing gear is sufficiently compliant to absorb large stresses caused by imperfect landings, preventing possible damage to the airframe and the sensitive sensor payload. The final version of the landing gear is depicted in Figure 2. This construction provides a lightweight and durable solution

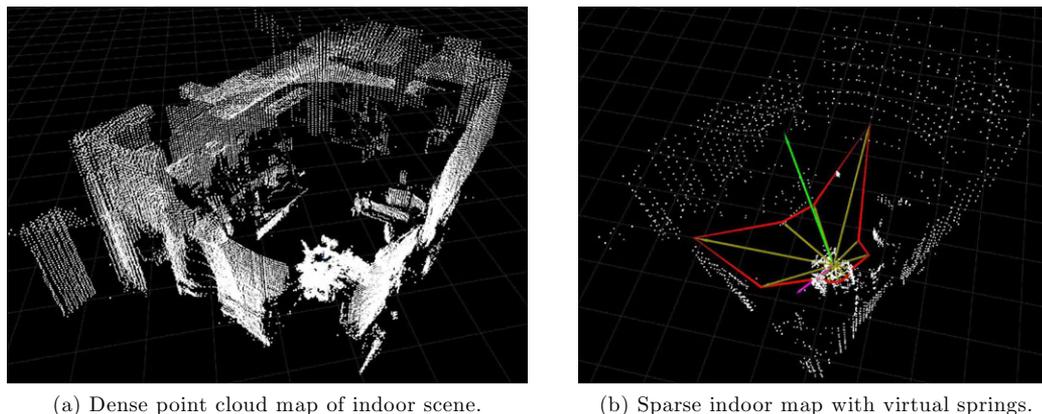


Figure 26. 3D point cloud maps shown on a 1 m by 1 m grid. (a) High-density point cloud generated by the onboard logging computer from laser and IMU data during half a rotation of the sensor unit in an office environment. (b) A map of the same office environment, generated from downsampled laser data for faster computation. The red boundary encloses the space that was labeled as unoccupied. The purple arrow indicates the platform's current heading. The yellow arrows indicate the virtual linear compression springs that exert forces between the environment and the platform (the shorter the spring, the higher the force). The green arrow represents the scaled sum of these forces and thus the momentarily preferred flight direction to increase the distance to close obstacles. All arrows start in the center of the quadcopter.

for compliant landing gears that also minimizes occlusions to the field of view of the laser scanner.

Collision avoidance. Safety aspects need to be taken into account when operating a quadcopter MAV in general and especially when flying in confined and fragile environments or at height. Although with sufficient training and experience it is not necessarily difficult to manually pilot a well-developed quadcopter platform, some special requirements of the scanning application complicate the usual operation. These are mainly flying at distance from the pilot to cover a large area (for example, at height along rock faces), flying close to obstacles in complex narrow spaces, and flying in poorly lit environments such as underground sites. The spinning payload can add additional confusion for the pilot when unsure about the quadcopter's orientation.

For our flights in confined spaces, we equipped *bentwing* with a protective bumper to prevent damage in case of collision and mainly relied on the pilot to avoid collisions. However, the need for safe operation and damage prevention to both the MAV and the scanned environment suggests that more sophisticated and effective methods of collision avoidance would be preferable. We have started to implement autonomous obstacle detection and avoidance capabilities on *bentwing's* onboard computer in order to enable the MAV to assist the pilot in avoiding collisions. Based on the laser and IMU sensor data, *bentwing* is able to build local coarse 3D point cloud maps onboard and in real time, which are sufficient to detect potential obstacles. An occupancy grid-based approach (Elfes, 1989) is used to efficiently distinguish between free and occluded space in these maps. Virtual linear compression springs between the MAV and detected obstacles generate a virtual force that pushes

the MAV away from obstacles in close proximity. This virtual force is brought into effect by appropriately derived roll and pitch commands that are fused with the operator's commands (see Figure 26). The so created virtual bumper prevents the quadcopter from getting into contact with obstacles even if the pilot actively steers it toward them. Initial testing of this approach has been promising but occurred after the site deployments described in this article. The developed technique can be used as a very low-level safety feature that seamlessly integrates with higher-level control, be it the pilot's commands, GPS guidance, or any other method of flight plan generation.

5.1. Limitations

The *bentwing* aerial mapping system has proved successful in a variety of applications but nonetheless has several limitations that affect where and how it can be used. Some of these constraints relate to sensor or platform limitations and are largely beyond our control, while others are limitations of the SLAM solution, and yet others can be addressed in design improvements that will be implemented in the next-generation prototype.

Environmental structure. The SLAM solution requires a distribution of surface normals that spans the three orthogonal spatial axes. Flying in open, geometrically impoverished areas is likely to introduce a situation in which the sensor motion cannot be uniquely determined, and therefore, the system may fail in such environments. It is possible to switch over to an independent state estimates (GPS or other sensors) while temporarily traversing through feature-poor areas, and returning to the SLAM solution closer to more

structurally interesting areas. There is also a risk in some environments that flying at a high altitude above the primary structure would result in a similar situation where nearly all of the observable structure is from the top surfaces of objects with mainly upward-pointing normals, and very little structure is visible with horizontal surface normal components. Thus, some attention must be given in determining whether an environment is likely to be appropriate for *bentwing*, and if so whether there are some flight patterns that should be avoided to ensure sufficient 3D structure is always within the laser's field of view. Currently, the flight path considerations are up to the pilot's discretion, but by considering the distribution of visible surfaces, there is potential to alternatively integrate autonomous path planning or operator assist features to ensure the vehicle is flown in favorable areas.

Laser specifications. The lightweight Hokuyo laser has a maximum range of 30 m in ideal conditions, though outdoors in daylight the maximum range is typically closer to 15 m. This limitation restricts the altitude and ranges at which the platform can be flown and also the swath of the sensor coverage. As mentioned previously, the scan rate of the laser scanner affects the measurement density, which constrains the maximum payload rotation speed as well as the flight altitude.

Upward field of view. As described in Section 3, the current mechanism generates a generally spherical field of view for the laser scanner with a 90° blind cone over the center (Figure 4). This field of view is sufficient for scanning many outdoor environments, where any rays pointing upward toward the sky are unnecessary. However, this arrangement is not ideal for enclosed settings where there is a requirement to scan overhanging surfaces, such as ceilings or tree canopies. Some scan rays outside the blind cone are oriented in an upward direction, and therefore, the scanner can fill in coverage of overhanging surfaces where sufficient horizontal motion is possible. In areas where the overhanging surfaces are beyond sensor range above the quadcopter or horizontal movement is too constrained, the final result may still lack overhead coverage. An example of this situation occurred in Lucas Cave (Figure 16), where it was possible for the platform to ascend vertically in one section of the chamber, but the space was too confined for much horizontal motion. We are addressing this shortcoming in the next design of the sensor payload, where we will mount the laser with the center ray 45° below the horizontal (as opposed to pointing directly down), which results in the boundary ray pointing vertically upward. We will also mount the laser further off-center and with the scan plane non-vertical to ensure that upward-pointing rays are not blocked by the MAV itself. This modification will significantly reduce the blind spot, thus enabling *bentwing* to deliver more comprehensive coverage in confined spaces.

Offboard processing. The data from *bentwing* cannot currently be processed in real-time on the onboard

acquisition computer. The SLAM solution is capable of processing data in real-time, or post-acquisition at a rate faster than that at which it is collected, but this has only been demonstrated on relatively higher-grade computing hardware (mid-range consumer laptops and desktops are generally capable of doing so) as compared to the logging computer. While for manual-flight survey applications offboard processing is acceptable, an onboard real-time solution is required for more complex autonomous functions, including exploration, path planning, and interaction with the environment. We are presently working toward optimizing our software and, in parallel with improvements in computing power, anticipate that it will be possible to process the data in real-time onboard a MAV in the future.

Flight time. One of the clear limitations of *bentwing* is the duration of individual scanning flights. Whereas the low power consumption and operator-carried batteries of *Zebedee* allow for extremely long scan times of up to several hours, the high power demand and limited payload capacity of the *bentwing* platform results in strict restrictions on flight and scan time. The power consumption of *bentwing* in hover flight amounts to approximately 550 W. We use lithium-polymer batteries with 148 Wh capacity that allow for approximately 10–12 min of flight operation with a sufficient safety margin. After a mapping flight, the discharged batteries can quickly be exchanged for charged ones and scanning can be resumed. Because our solution enables the straightforward alignment of multiple datasets, the limited individual flight time does not limit the area that can be scanned, provided that enough spare batteries (and charging time) are available.

Situational awareness before takeoff. Prior to takeoff, *bentwing*'s sensor payload is stationary and does not start spinning until it leaves the ground. Thus, data are available for mapping or situational awareness within a second or two into the flight but not in advance. In cases where an autonomous or teleoperated system requires further situational awareness in order to decide whether it is safe to take off in the first place, the current sensor configuration is not adequate for that purpose.

Georeferencing. The trajectories and maps currently produced by *bentwing* are in an arbitrary coordinate frame based on the starting pose. For outdoor runs, it is reasonably straightforward to incorporate loose GPS measurement constraints into the global registration in order to georeference the data. This enhancement has so far been demonstrated on *Zebedee* but not yet tested on *bentwing* (and GPS data were not logged in the deployments thus far).

6. CONCLUSIONS

Accurate, efficient, and dense 3D mapping from aerial platforms has the potential to benefit a wide variety of applications. We have developed and presented *bentwing*, an aerial laser mapping system capable of generating

point cloud models of environments at typical scales of tens to hundreds of meters (or even larger when combining multiple scans). *bentwing* incorporates our existing continuous-time SLAM framework that has been successfully applied to ground-based hardware platforms. *bentwing* also introduces a novel method to passively actuate the sensor unit's rotation, exploiting, through adjustable vanes, the naturally present downdraft of the quadcopter's four rotors. The system has been demonstrated in a range of environment types including caves, forests, heritage sites, underground mines, and industrial facilities. The mutual complementarity of aerial and ground-based sensing has also been illustrated by merging data collected with *bentwing* and the *Zebedee* handheld mapping system.

The mechanism developed for passively rotating the sensor payload, although generally successful, has some shortcomings that may limit its utility for some applications. In particular, using this method for consistently generating the relatively slow rotation that optimizes mapping results (given the current sensor configuration) has proved to be challenging. The main difficulty is that the airflow under the copter is not as steady and homogeneous as we initially assumed but very much influenced and disturbed by crosswinds, headwinds, and flight maneuvers. Although very rare, on some occasions the disturbances can invoke an unrecoverable stall in the sensor rotation, leading to a lack of useful data for the SLAM solution to estimate the platform motion and generate maps. Though the developed mechanism is capable of providing the desired motion in most flight conditions, its mechanical complexity and need for sophisticated feedback control do not necessarily render it a preferred option compared to motor-spun designs. However, we found that the presented method of passive actuation is well-suited for generating rotations at higher speeds above 720°/s, where the inertia of the rotating unit becomes significant and effectively counteracts changes in rotational speed caused by external factors. At these higher speeds, adjustable vanes and feedback control are not needed, making passive actuation simpler than a motor-driven design would be. While our particular application could only showcase the possibility but not the full advantages of passive sensor actuation on a quadrotor MAV, other applications or configurations that allow slightly faster sensor rotation will likely be able to benefit further from this approach.

We have sought to experimentally evaluate the robustness of our actuation mechanism against external disturbances, namely, through artificially generated headwinds and crosswinds. However, we could not observe a close similarity between the effects encountered in actual flight trials and the results of these simulated conditions, nor have some of the experimental results been reliably reproducible. This outcome suggests a need for further work to be carried out to appropriately understand and model the aerodynamic

conditions under a quadcopter MAV and its interaction with the rotating vanes.

The performance of the system as a localization and 3D mapping solution was evaluated quantitatively through comparison with conventional surveying equipment, including a total station and a terrestrial laser scanner. Comparison with the total station verified the accuracy of the estimated *bentwing* trajectory, which was observed to have zero-mean error and variance within the total station tolerance (which is effectively at centimeter-scale levels for moving targets) in both a highly structured industrial site and a more vegetated environment. The 3D point cloud generated from *bentwing* matches closely with a point cloud of the same environment generated from the more precise TLS measurements and has comparable density. The error of the *bentwing* point cloud with respect to the TLS data is zero-mean and with a standard deviation of 1.4 cm, which can be explained by the increased range noise inherent to *bentwing's* Hokuyo laser. Finally, an analysis of the drift error by comparing open-loop and closed-loop *bentwing* trajectories, indicates that the translational drift is on the order of one-tenth of a percent of distance traveled in the environments and conditions considered. Rotationally, the error about the laser spin axis is on the order of one-hundredth of a degree per meter traveled and in the platform horizontal direction about one-thousandth of a degree per meter traveled.

Future work will primarily focus on improvements to the hardware design, reliability, efficiency, and increased platform autonomy. In the next version of *bentwing*, the laser scanner mount will be modified to ensure greater coverage of the environment (including above the platform), and components will be optimized or replaced to reduce the overall mass of the payload. Integration of other onboard sensors such as GPS can alleviate temporary situations in which the environment is too sparse for the SLAM solution to estimate platform motion. In terms of autonomy, we plan investigate improvements to the MAV's collision avoidance capabilities to increase safety of manual operation and will further develop *bentwing* into a fully autonomous exploration and mapping platform. For some autonomous functionality, onboard real-time processing will be needed, which will require further development and optimization of the mapping software. An eventual goal is to have the ability to send the platform beyond line-of-sight into a complex, unknown space and have it return with a comprehensive 3D map.

APPENDIX: INDEX TO MULTIMEDIA EXTENSIONS

A video demonstrating *bentwing* in flight and illustrating 3D mapping results in a variety of environments is available as Supporting Information in the online version of this article.

Acknowledgments

The authors would like to thank the following persons and organizations, and acknowledge their contributions to the presented work:

CSIRO: Paul Flick for providing invaluable support with *bentwing* hardware development. Farid Kendoul for sharing his expertise on UAV operation. Peter Reid and Lance Munday for providing advice and help with operating and modifying the Eagle quadcopter platform. Navinda Kottege for his assistance with operating the total station for the tracking experiments. Jonathan Roberts for his enthusiastic support of the *bentwing* project and assistance in some of the fieldwork.

Site support: Dan Cove and Jenolan Caves Reserve Trust for providing support and access to areas of Jenolan caves (Jan 2014). Michael Lake, Jill Rowling, Bruce Welch, and Rowena Larkins of the Sydney University Speleological Society for guidance at Wombeyan Caves, and the Wombeyan Caves management for making our flights in these caves possible (Jan 2014). The ML3 Quarry and Wombeyan Caves fieldwork was financially supported in part by a grant of the Australian Speleological Federation. Rowland Dowling of the Queensland Parks and Wildlife Service and Kelly Greenop of the University of Queensland for their support during our mapping flights on Peel Island (Feb 2014). The staff of the Mastermyne Myne Start Underground Training Facility in Brisbane for granting us access and allowing us to operate *bentwing* in the mine simulation environment (Mar 2014).

This work was partially supported through a Dr.-Ing. Willy-Höfler-Foundation scholarship grant.

REFERENCES

- Alismail, H., Baker, L. D., & Browning, B. (2014). Continuous trajectory estimation for 3D SLAM from actuated lidar. In IEEE International Conference on Robotics and Automation, Hong Kong, May/June 2014.
- Anderson, S., & Barfoot, T. D. (2013). Towards relative continuous-time SLAM. In IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 2013.
- Bachrach, A., Prentice, S., He, R., & Roy, N. (2011). RANGE - Robust Autonomous Navigation in GPS-Denied Environments. *Journal of Field Robotics*, 28(5), 644–666.
- Baltsavias, E. B. (1999). Airborne laser scanning: Existing systems and firms and other resources. *Journal of Photogrammetry and Remote Sensing*, 54, 164–198.
- Bosse, M., & Zlot, R. (2009a). Continuous 3D scan-matching with a spinning 2D laser. In IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.
- Bosse, M., & Zlot, R. (2009b). Keypoint design and evaluation for place recognition in 2D lidar maps. *Robotics and Autonomous Systems*, 57(12), 1211–1224.
- Bosse, M., & Zlot, R. (2013). Place recognition using keypoint voting in large 3D lidar datasets. In IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 2013.
- Bosse, M., Zlot, R., & Flick, P. (2012). Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5), 1104–1119.
- Dong, H., & Barfoot, T. D. (2012). Lighting-invariant visual odometry using lidar intensity imagery and pose interpolation. In International Conference on Field and Service Robotics, Matsushima, Japan, July 2012.
- Droeschel, D., Holz, D., & Behnke, S. (2014a). Omnidirectional Perception for Lightweight MAVs using a Continuously Rotating 3D Laser Scanner. *Photogrammetrie, Fernerkundung, Geoinformation*, 5, 451–464.
- Droeschel, D., Stückler, J., & Behnke, S. (2014b). Local Multi-Resolution Representation for 6D Motion Estimation and Mapping with a Continuously Rotating 3D Laser Scanner. In IEEE International Conference on Robotics and Automation, Hong Kong, May/June 2014.
- Elfes, A. (1989). Occupancy grids: A probabilistic framework for robot perception and navigation. PhD thesis, Electrical and Computer Engineering Department/The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.
- Elseberg, J., Magnenat, S., Siegwart, R., & Nüchter, A. (2012). Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. *Journal of Software Engineering for Robotics*, 3(1), 2–12.
- Elseberg, J., Borrmann, D., & Nüchter, A. (2013). Algorithmic solutions for computing precise maximum likelihood 3D point clouds from mobile laser scanning platforms. *Remote Sensing*, 5, 5871–5906.
- Grzonka, S., Grisetti, G., & Burgard, W. (2009). Towards a Navigation System for Autonomous Indoor Flying. In IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.
- Holenstein, C., Zlot, R., & Bosse, M. (2011). Watertight surface reconstruction of caves from 3D lidar data. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, USA, September 2011.
- Holz, D., & Behnke, S. (2014). Mapping with Micro Aerial Vehicles by Registration of Sparse 3D Laser Scans. In International Conference on Intelligent Autonomous Systems (IAS), Padova, Italy, July 2014.
- Lin, Y., Hyypä, J., & Jaakkola, A. (2011). Mini-UAV-Borne LIDAR for Fine-Scale Mapping. *IEEE Geoscience and Remote Sensing Letters*, 8(3).
- Moosmann, F., & Stiller, C. (2011). Velodyne SLAM. In IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, June 2011.
- Nüchter, A., Lingemann, K., Hertzberg, J., & Surmann, H. (2007). 6D SLAM—3D mapping outdoor environments. *Journal of Field Robotics*, 24(8/9), 699–722.
- Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., & Singh, S. (2012). River mapping from a

- flying robot: State estimation, river detection, and obstacle mapping. *Autonomous Robots*, 32(5), 1–26.
- Shen, S., & Michael, N. (2012). State estimation for indoor and outdoor operation with a micro-aerial vehicle. In *International Symposium on Experimental Robotics*, Quebec City, Canada. June 2012
- Tong, C. H., Anderson, S., Dong, H., & Barfoot, T. D. (2014) Pose interpolation for laser-based visual odometry. *Journal of Field Robotics*, 31(5), 731–757.
- Walthelm, A. (2004). Enhancing global pose estimation with laser range scans using local techniques. In *International Conference on Intelligent Autonomous Systems*, Amsterdam, Netherlands. November 2004.
- Zhang, J., & Singh, S. (2014). LOAM: Lidar Odometry and Mapping in real-time. In *Robotics: Science and Systems Conference*, Berkeley, USA. July 2014.
- Zlot, R., & Bosse, M. (2014a). Three-dimensional mobile mapping of caves. *Journal of Cave and Karst Studies*, 76(3), 191–206.
- Zlot, R., & Bosse, M. (2014b). Efficient large-scale three-dimensional mobile mapping for underground mines. *Journal of Field Robotics*, 31(5), 758–779.
- Zlot, R., Bosse, M., Greenop, K., Jarzab, Z., Juckes, E., & Roberts, J. (2014). Efficiently capturing large, complex cultural heritage sites with a handheld mobile 3D laser mapping system. *Journal of Cultural Heritage*, 15(6), 670–678.