

Method for 3-D Scene Reconstruction Using Fused LiDAR and Imagery From a Texel Camera

Taylor C. Bybee^{ID} and Scott E. Budge^{ID}, *Senior Member, IEEE*

Abstract—Reconstructing a 3-D scene from aerial sensor data creating a textured digital surface model (TDSM), consisting of a LiDAR point cloud and an overlaid image, is valuable in many applications including agriculture, military, surveying, and natural disaster response. When collecting LiDAR from an aircraft, the navigation system accuracy must exceed the LiDAR accuracy to properly reference returns in 3-D space. Precision navigation systems can be expensive and often require full-scale aircraft to house such systems. Synchronizing the LiDAR sensor and a camera, using a texel camera calibration, provides additional information that reduces the need for precision navigation equipment. This paper describes a bundle adjustment technique for aerial texel images that allows for relatively low-accuracy navigation systems to be used with low-cost LiDAR and camera data to form higher fidelity terrain models. The bundle adjustment objective function utilizes matching image points, measured LiDAR distances, and the texel camera calibration and does not require overlapping LiDAR scans or ground control points. The utility of this method is proven using a simulated texel camera and unmanned aerial system (UAS) flight data created from aerial photographs and elevation data. A small UAS is chosen as the target vehicle due to its relatively inexpensive hardware and operating costs, illustrating the power of this method in accurately referencing the LiDAR and camera data. In the 3-D reconstruction, the $1-\sigma$ accuracy between LiDAR measurements across the scene is on the order of the digital camera pixel size.

Index Terms—Bundle adjustment, image registration, LiDAR, multisensor systems, photogrammetry, remote sensing.

I. INTRODUCTION

TEXTURED digital surface models (TDSMs) are used in a variety of industries and are advantageous for determining the extent of large-scale effects over time such as agriculture, forestry, and natural disasters. The use of TDSMs in battlefield situational awareness can significantly aid critical circumstances. These 3-D representations of landscapes are constructed using a variety of methods, based on the type of sensors aboard an aerial vehicle and the desired level of accuracy to be achieved in the reconstruction process.

Highly accurate TDSMs are expensive to obtain, in terms of hardware cost, capture costs, and postprocessing

Manuscript received September 8, 2018; revised December 18, 2018 and April 11, 2019; accepted June 2, 2019. Date of publication July 10, 2019; date of current version October 31, 2019. (Corresponding author: Taylor C. Bybee.)

The authors are with the Center for Advanced Imaging LADAR, Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322 USA (e-mail: taylor.bybee@aggiemail.usu.edu; scott.budge@ece.usu.edu).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2019.2923551

computational demands. Much of the cost may be attributed to the high precision navigation equipment used on a full-scale aircraft in order to properly reference LiDAR measurements in space. On smaller aircraft with less-accurate navigation systems, postprocessing computational requirements become the primary cost factor. The costs associated with the use of LiDAR and photogrammetry for surface reconstruction, along with algorithms for doing so, are compared in [1].

Triangulation of the aerial images has been used for decades to produce both orthophotos and digital surface models (DSMs) [2], [3]. Photogrammetry-based methods do not make any direct range measurements of the surface and result in esthetically pleasing TDSMs, but the 3-D information may not be as accurate as LiDAR because scale information is lost in the image projections [4], [5]. Scale can be recovered to some degree using ground control points, camera location [6], [7], or human intervention [8], but models still infer, not measure, 3-D information. A contemporary overview of unmanned aerial system (UAS) photogrammetry technology is given by Colomina and Molina [9]. Recent studies of photogrammetry-based unmanned aerial vehicle surveys have also been done [10], [11].

LiDAR has an advantage over photogrammetry in that: 1) direct range measurements are made and 2) LiDAR beams can penetrate a foliage canopy allowing for bare-earth views. However, the methods using accurate LiDAR systems require precise knowledge of the location of the aerial vehicle using high-quality navigation equipment needed to reference the measurements [12]. A comparison between low-cost aerial LiDAR and terrestrial LiDAR was given by Roca *et al.* [13]. To make the LiDAR points more useful in several applications, one or more images can be overlaid on the point cloud. This overlaying operation is not a trivial process often requiring manual intervention [14]–[18].

The accuracy and cost drawbacks of photogrammetry and LiDAR, respectively, make it difficult to obtain low-cost, high-quality data with the measurement errors on the order of a few centimeters without precision navigation equipment. This paper describes a bundle adjustment method using fused imagery and LiDAR to reconstruct a 3-D scene with 3-D distance accuracy approaching the image pixel size on the ground. This method is proven using the data acquired with a simulated UAS fitted with a texel camera (a low-cost digital camera and a LiDAR sensor) and a low-cost MEMS-based GPS/inertial measurement unit (IMU) navigation system, showing that highly accurate localization equipment is not necessary. This paper is a continuation of work presented in [19] but with

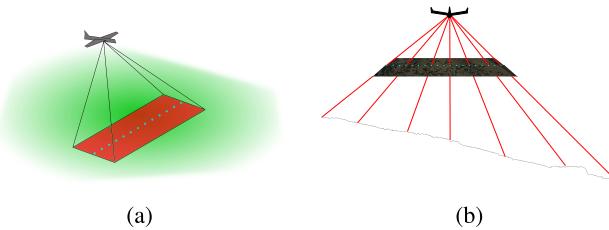


Fig. 1. Acquisition of texel swaths. (a) Acquisition of texel swaths with the field of view of the digital camera in red, and LiDAR shots in cyan. (b) Image and range measurements in a texel swath. The red lines show some of the many LiDAR rays from the camera center to the ground (surface profile). These rays intersect the normalized image plane at the calibrated projections, which lie along the dotted cyan line.

significant developments in texture registration and experimental results with additional scenarios having the ground truth data. There has been some work done integrating range sensing and photogrammetry (see [20] for an overview). Much of the work in this area still requires a postprocessing step for LiDAR-to-image fusion (see [21]–[32]) unlike a texel camera, while another estimates lever arms for bore sighting between the sensors [33]. Techniques similar to ours can be found in [34] and [35], but these differ because we assume the fusion of imagery and LiDAR at capture using texel camera assumptions.

The texel camera, introduced over a decade ago, is an instrument that combines LiDAR and imagery upon capture in a calibrated manner, such that the measurements from each sensor are fused in both time and space as described in [36]. The fusing is accomplished through a one-time factory calibration process that involves mapping each LiDAR measurement to a subpixel point on the digital image. Because the sensors are aligned in space and captured synchronously, this mapping does not change from capture-to-capture and is not subject to motion blur [37]. By virtue of the calibration, the imagery and LiDAR measurements can be used together without the drawbacks of computing a registration between the imagery and LiDAR measurements in each capture.

The LiDAR portion of the UAS texel camera typically sweeps from side-to-side along the cross-track flight direction, and a correspondingly wide digital image is acquired concurrently, shown in Fig. 1, along with information about the sensor pose in some world coordinate system (such as GPS/IMU). This skinny-shaped data set is termed a texel swath. With each successive texel swath, there is an overlap in imagery but not necessarily any overlapping LiDAR measurements. Although low-cost multiple-beam LiDAR systems are available, a single beam per sweep is assumed in this work. This is because a single-beam LiDAR can be designed with a much farther range using the same battery size on a small UAS, allowing for a higher altitude flight and correspondingly larger ground coverage. The method described in this paper can be extended to multiple-beam LiDARs in a straightforward manner. The use of a wide digital image is a design tradeoff as increasing the amount of imagery in the in-track direction will strengthen the block geometry but may require more capture time on the sensor itself. Our method works given there is sufficient LiDAR overlap into neighboring images.

Because the LiDAR points from each texel swath are not assumed to overlap, the point cloud registration techniques such as iterative closest point [38] or the normal distributions transform [39] cannot be used explicitly with these data sets. Classic photogrammetry methods do not have fused and calibrated range measurements available to supplement the 3-D scene reconstruction. Knowing where these LiDAR points are projected into several images allows for triangulation of texel swaths using both imagery and LiDAR. Together, this LiDAR-to-image and image-to-image matching allows for accurate TDSM reconstruction. There is a technique described by Habib *et al.* [40] where the LiDAR points are matched with other LiDAR points, but no known techniques matching LiDAR points to imagery using texel camera assumptions have been found. This texel swath triangulation method appears to be unique. The utility of this method is shown by reconstructing a TDSM of a scene created from aerial images and the measured LiDAR data using the simulated texel swaths. Variances of distance errors between points in the reconstructed scene and ground truth points are shown to decrease significantly, approaching the accuracy of the LiDAR sensor without the need for precision navigation equipment.

The remainder of this paper is as follows. Section II describes the details of forming TDSMs from texel swaths using image processing techniques and triangulation. Section III describes the reconstruction results from the simulated data sets. Section IV offers a conclusion.

II. TEXEL SWATH TRIANGULATION AND TDSM CONSTRUCTION

The bundle adjustment algorithm uses both 2-D (image) and 3-D (LiDAR) data to form a TDSM. Because several types of data are manipulated in this algorithm, we describe the key mathematical notation in Section II-A. The construction of the TDSM is described in four basic steps.

- 1) Estimate pairwise image homographies (see Section II-B). These are used in two of the subsequent steps.
- 2) Create the DSM texture. The pairwise homographies are used to construct a registered image (texture) and the LiDAR-to-texture mapping. This step is described in Section II-C.
- 3) Select common LiDAR projection points. Section II-D outlines how the pairwise homographies are used to find the projection of each LiDAR measurement into the neighboring swath images.
- 4) Bundle adjustment of 2-D and 3-D information. The projections and LiDAR measurements are used to triangulate the point cloud. Triangulation is done by forming an objective function involving the LiDAR range measurements, their calibrated projections into the associated swath imagery, and their projection into neighboring swath imagery. The objective function is presented in Section II-E.

Minimizing the objective function recovers the capture poses, allowing the LiDAR points to be correctly referenced to one another, mitigating the inaccuracies of a low-cost navigation system. The optimized LiDAR point cloud together with

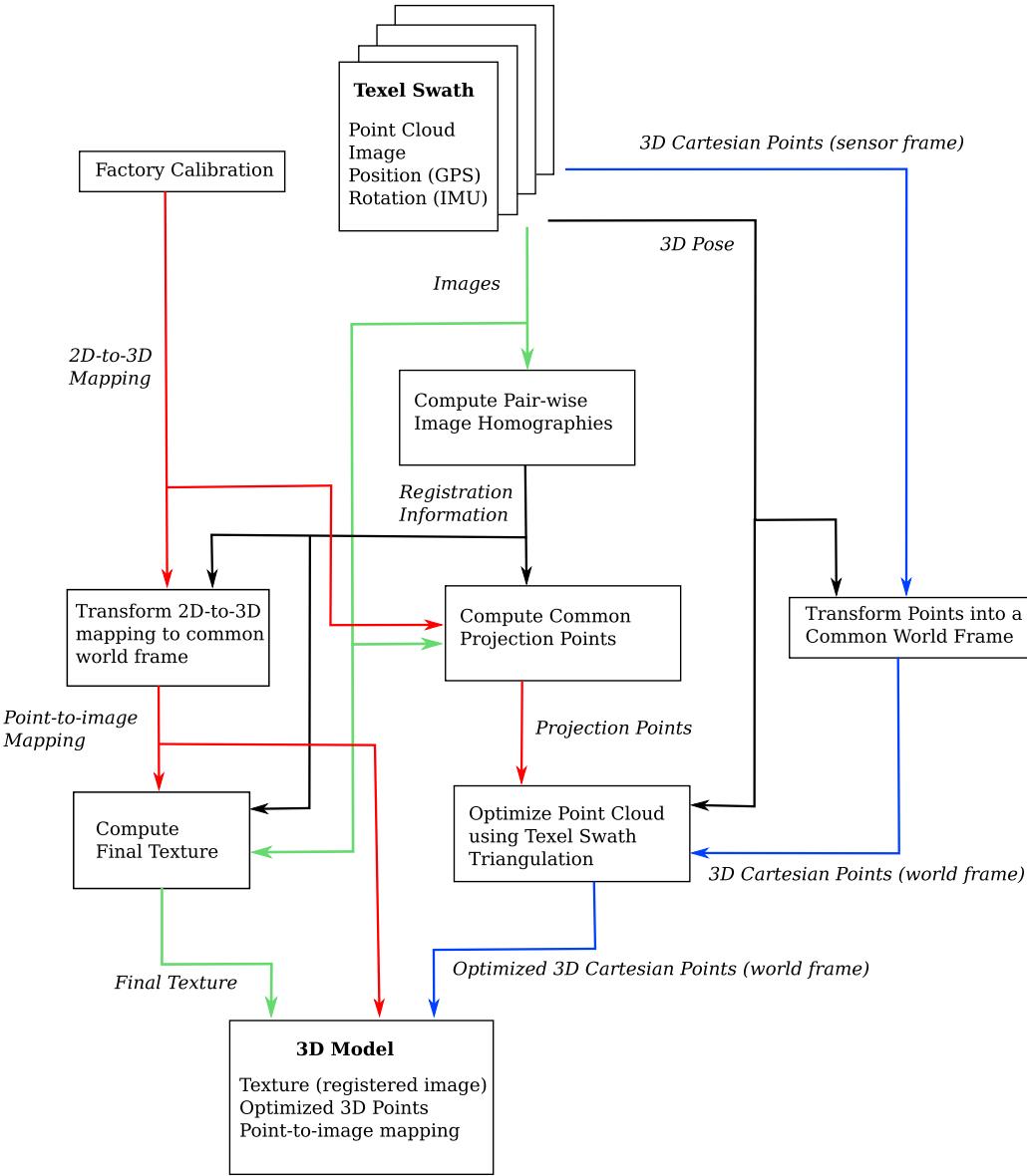


Fig. 2. Algorithm information flow described in this section. Red: 3-D-to-2-D projection information. Green: images. Blue: 3-D point information. Black: supporting information, such as pairwise homography registration and measured position and rotation information.

the reconstructed texture and measurement-to-texture mapping form the TDSM. A flowchart showing the information flow is shown in Fig. 2.

A. Mathematical Notation and Coordinate System Transforms

Each texel swath W_j (for $j = 1, \dots, M$) contains n_j LiDAR measurements, a set of 3-D points denoted as I_j , an image, and capture pose information \mathbf{a}_j . The pose information is described by a 7-D vector $\mathbf{a}_j = [q_{wj}, q_{xj}, q_{yj}, q_{zj}, t_{xj}, t_{yj}, t_{zj}]^T$, comprised of a quaternion rotation (see [46] for further discussion) and a 3-D translation.

Using the quaternion to create a rotation matrix R_j , the corresponding rotation is (1), as shown at the bottom of the next page.

The total number of 3-D points is $N = \sum_{j=1}^M n_j$, and the entire set of 3-D points is indexed using i . A 3-D point \mathbf{b}_i is acquired in texel swath W_j if $\mathbf{b}_i \in I_j$. Conversely, if swath

W_j did not acquire \mathbf{b}_i , this is denoted using $\mathbf{b}_i \notin I_j$. This is shown subsequently in Fig. 3.

Because each texel swath has its own coordinate system, we denote the i th point in the W_j coordinate system by $\beta_{ij} = [\beta_{ij_x}, \beta_{ij_y}, \beta_{ij_z}]^T$. Additionally, each point can be mapped into an arbitrary world Cartesian coordinate system \mathcal{O} using the information from a low-cost GPS/IMU. The i th point in coordinate system \mathcal{O} is denoted as $\mathbf{b}_i = [b_{ix}, b_{iy}, b_{iz}]^T$. The coordinate transform from \mathcal{O} to W_j is performed by

$$\beta_{ij} = R_j^T(\mathbf{b}_i - \mathbf{t}_j). \quad (2)$$

Because we assume each texel swath coordinate system has the camera projection center as the origin of that coordinate system and a pinhole camera model is assumed, we compute the projection of each point β_{ij} into the normalized image plane of the corresponding texel swath W_j by dividing the

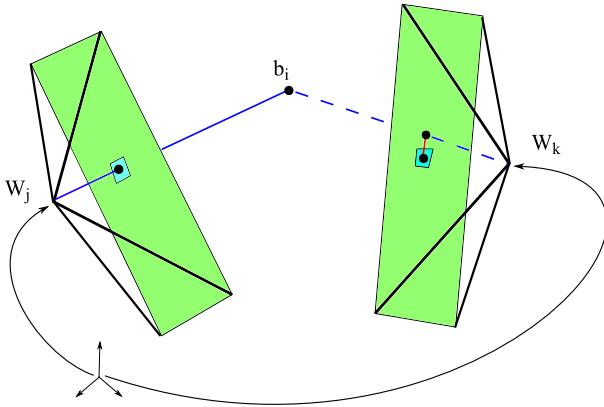


Fig. 3. Projection of a 3-D point \mathbf{b}_i into texel swaths W_j and W_k . Green areas: normalized image plane of each swath. The blue patches in each swath represent matching image patches, with the patch in W_j used as the reference patch. The patch in W_k is found using the techniques described in Section II-D. The point \mathbf{b}_i is captured with swath W_j (i.e., $\mathbf{b}_i \in I_j$). Blue solid line: measured range λ_{ij} . The calibrated projection point (x_{ij}, y_{ij}) is the dot in the image patch of W_j . The common projection point (x_{ik}, y_{ik}) is the dot in the image patch of W_k . The intersection of the blue dotted line with the normalized image plane is the projection point $(\hat{x}_{ik}, \hat{y}_{ik})$. The red line segment shows the error ϵ_{ik} . Because the poses and point locations are adjusted in the optimization, the relationship between \mathbf{b}_i and W_j is not constant, introducing $(\hat{x}_{ij}, \hat{y}_{ij})$ and $\hat{\lambda}_{ij}$, which are not explicitly shown in this figure.

coordinate perpendicular to the normalized image plane, we use the z -direction. This projection is denoted as (x_{ij}, y_{ij}) . The range from camera projection center of the W_j swath to the point β_{ij} is λ_{ij} . The projection and range information is combined into a vector given by

$$\mathbf{B}_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ \lambda_{ij} \end{bmatrix} = \begin{bmatrix} \frac{\beta_{ij_x}}{\beta_{ij_z}} \\ \frac{\beta_{ij_y}}{\beta_{ij_z}} \\ \frac{\beta_{ij_z}}{\sqrt{\beta_{ij_x}^2 + \beta_{ij_y}^2 + \beta_{ij_z}^2}} \end{bmatrix} \quad (3)$$

which is another way to represent a 3-D point in space. There are singularities and hemisphere ambiguities associated with this representation, but we ensure all points lie in the same hemisphere and away from the singularities.

The mapping from (x_{ij}, y_{ij}) to discrete pixel coordinates is known by virtue of the intrinsic camera calibration. We assume that any distortion caused by optics is removed. The registered image overlaid on the point cloud I is called the final texture, and it is formed from the images in each swath. A projection point in the final texture representing the 3-D point \mathbf{b}_i is denoted as ρ_i and is maintained in continuous image pixel

coordinates. A discrete pixel in the final texture is denoted generally as \mathbf{p} .

B. Estimating Pairwise Homographies

Pairwise homographies are used for both creating the DSM texture and in the selection of common projection points. Because the texel swaths are ordered by virtue of successive captures in time, pairwise homographies can be computed between successive digital images in the texel swaths. From these homographies, the approximate location of any image pixel in a swath in any other swath is computed by a concatenation of the homographies between them. Since the swaths are acquired very close together in space and time, the concatenated estimates are fairly accurate.

The pairwise homographies are estimated by finding and matching feature points in each swath image and applying least-squares techniques to estimate the 2-D homography between each adjacent pair of swath images. Various techniques can be used to find and match the necessary feature points for this algorithm; in this work, we use a Harris corner detector (see [42]) to find putative correspondences because there is little or no rotation and lighting changes between pairs of images. We have also used a speeded up robust features detector [43] and it achieves comparable results. Normalized cross correction (see [44]) matches the features, and the random sample consensus algorithm (RANSAC) filters outliers (see [45]). We choose to use a homography defined by a scale-invariant projective transform and solved in a total-least-squares manner.

Note that the pairwise homographies (and the associated features used to find them) are not used in the texel swath triangulation bundle adjustment and do not directly affect the accuracy of the optimized 3-D points. The pairwise homographies are used to create the final DSM texture and to seed the selection of common LiDAR projection points. The points in the texel swath triangulation bundle adjustment represent matching LiDAR projection points not the matching features described above.

C. Creating the DSM Texture

The texture overlaid on the point cloud I is created from the digital images in the texel swaths. The pairwise homographies allow each swath image to be registered in a common image frame. The swath with the smallest cross-track distance is used as the common frame so no image resolution is lost. The convex hull of the swath images in the common frame determines the limits on the final texture size.

$$R_j = \begin{bmatrix} 1 - \frac{2(q_{yj}^2 + q_{zj}^2)}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} & \frac{2(q_{xj}q_{yj} - q_{wj}q_{zj})}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} & \frac{2(q_{xj}q_{zj} + q_{wj}q_{yj})}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} \\ \frac{2(q_{xj}q_{yj} + q_{wj}q_{zj})}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} & 1 - \frac{2(q_{xj}^2 + q_{zj}^2)}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} & \frac{2(q_{yj}q_{zj} - q_{wj}q_{xj})}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} \\ \frac{2(q_{xj}q_{zj} - q_{wj}q_{yj})}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} & \frac{2(q_{yj}q_{xj} + q_{wj}q_{zj})}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} & 1 - \frac{2(q_{yj}^2 + q_{xj}^2)}{q_{wj}^2 + q_{xj}^2 + q_{yj}^2 + q_{zj}^2} \end{bmatrix} \quad (1)$$

The projection ρ_i of each acquired LiDAR point \mathbf{b}_i into the common image frame is calculated using the homographies concatenated between the swath in which the LiDAR point was acquired (W_j for $\mathbf{b}_i \in I_j$) and the common image frame. This is necessary so the final texture is mapped to the point cloud. Note this projection mapping is not a function of the capture poses but determined entirely using the 2-D imagery and the calibrated projection points in each swath.

With the projections of each point into the final texture and the pairwise homographies, the color value for each pixel in the final texture can be determined. This value for a discrete pixel \mathbf{p} is chosen by selecting one of the image candidates to which the final texture is mapped, and interpolating the color value at the mapped location. Selecting the image from which the interpolation takes place is somewhat an arbitrary decision. One way to determine which image the value of pixel \mathbf{p} is drawn from is to find the closest projection ρ_i to the pixel \mathbf{p} then interpolate the values from the swath from which the corresponding point \mathbf{b}_i was acquired; that is, W_j if $\mathbf{b}_i \in I_j$. Swath W_j likely has a good view of the area at and surrounding the image point ρ_i , making it a good image from which to interpolate color information.

Interpolation is performed for each pixel \mathbf{p} in the final texture. Together with the projections ρ_i and point cloud I (consisting of points \mathbf{b}_i), the final texture forms a TDSM in a triangular irregular network format. Note that this is not a colored point cloud but rather a point cloud with a registered image. The registration from the point cloud to the final texture is a result of the texel camera calibration. At the conclusion of this step, the final texture is formed and projections of \mathbf{b}_i are identified.

D. Selection of Common Projection Points Using Local Homography Assumptions

This step finds matching LiDAR projections in several images. The search for these projections is seeded with the pairwise homography information generated in a previous step and relies on the texel camera calibration.

Texel swaths are correlated with one another by LiDAR points projected into several swath images. These projections are determined by matching image patches in neighboring swaths to the image patch surrounding the calibrated projection in swath W_j . The point $\mathbf{b}_i \in I_j$ projects into neighboring texel swaths W_k , $j \neq k$. Because the texel swaths W_j and W_k are acquired from similar perspectives, the imagery surrounding the projections of point \mathbf{b}_i into W_j and W_k are assumed to be related by a homography.

In other words, a small reference image patch surrounding the calibrated projection (x_{ij}, y_{ij}) is assumed to look similar to the small image patch surrounding the projection (x_{ik}, y_{ik}) . The purpose of this step is to find each (x_{ik}, y_{ik}) . Correlation matching of the reference patch to the W_k image is used to find these common projection points. In the experiments described in this paper, a square window around the calibrated projection point serves as the reference template for the normalized cross correlation. To reduce the computation effort, search areas for the patch matching can be restricted to a much smaller area

using the cascaded pairwise homographies. As in Section II-C, no 3-D information, only imagery, is used in this step.

This step finds the projections (x_{ij}, y_{ij}) using the normalized cross correlation. These projection points are used in the triangulation objective function and represent the relationship between each LiDAR point and imagery in neighboring swaths.

E. Texel Swath Triangulation

Once projections of each point \mathbf{b}_i are found in several images, this information is used to triangulate the swath poses. Each LiDAR measurement has a range value and a calibrated 2-D projection point in the texel swath in which it was acquired, as well as projections into several adjacent texel swaths. An example of a single measured 3-D point belonging to the set I_j and projected into texel swaths W_j and W_k , showing matching image patches, is illustrated in Fig. 3.

The triangulation error is comprised of image projection error and measured range error, making use of the projection-range coordinate representation in (3) in the triangulation objective function. This projection-range coordinate system is chosen for the optimization as it separates image information $(\hat{x}_{ij}, \hat{y}_{ij})$ from range information $\hat{\lambda}_{ij}$ (see Fig. 3). Measured range error only exists for $\mathbf{b}_i \in I_j$. This leads to two types of triangulation error. First, for $\mathbf{b}_i \in I_j$, the projection error is defined in terms of distance from the calibrated projection in the image plane (x_{ij}, y_{ij}) to the projection of the point into the image plane $(\hat{x}_{ij}, \hat{y}_{ij})$. The range error is defined as the difference between the measured range value λ_{ij} and the range value $\hat{\lambda}_{ij}$. Second, for $\mathbf{b}_i \notin I_j$, the projection error is defined in terms of the distance from the assumed projection point (x_{ij}, y_{ij}) found by correlation matching (Section II-D) to the projection of the point into the image plane $(\hat{x}_{ij}, \hat{y}_{ij})$. Unlike the first case, there is no measured range of the 3-D point \mathbf{b}_i from texel swath W_j and does not contribute error to the objective function.

Formalizing this to include all M swaths and each 3-D point \mathbf{b}_i , the range and projection errors are combined into a single objective function

$$\mathcal{E}^2 = \sum_{j=1}^M \left(\sum_{\mathbf{b}_i \in I_j} \mathcal{E}_{ij}^T \Sigma_{xy\lambda}^{-1} \mathcal{E}_{ij} + \sum_{\mathbf{b}_i \notin I_j} \mathcal{E}_{ij}^T \Sigma_{xy}^{-1} \mathcal{E}_{ij} \right) \quad (4)$$

where

$$\mathcal{E}_{ij} = \mathbf{B}_{ij} - \hat{\mathbf{B}}_{ij} = \begin{bmatrix} x_{ij} - \hat{x}_{ij} \\ y_{ij} - \hat{y}_{ij} \\ \lambda_{ij} - \hat{\lambda}_{ij} \end{bmatrix}$$

$$\Sigma_{xy\lambda}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{\text{cal}}^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_{\text{cal}}^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_{\text{range}}^2} \end{bmatrix}, \quad \Sigma_{xy}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{\text{com}}^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_{\text{com}}^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_{\text{range}}^2} \end{bmatrix}$$

and, finally, σ_{cal}^2 , σ_{com}^2 , and σ_{range}^2 are the variances on the calibrated projected points, the common projection points, and the range measurements, respectively. If the measurements are

TABLE I
VALUES USED IN THE OPTIMIZATION

Value	Point Captured with Swath of Interest ($\mathbf{b}_i \in I_j$)	Point Not Captured with Swath of Interest ($\mathbf{b}_i \notin I_j$)
(x_{ij}, y_{ij})	Texel camera calibration	Patch matching described in Section II-D
λ_{ij}	Measured LiDAR range	Not used
$(\hat{x}_{ij}, \hat{y}_{ij})$	Computed from \mathbf{a}_j and \mathbf{b}_i , using (1)–(3)	Computed from \mathbf{a}_j and \mathbf{b}_i , using (1)–(3)
$\hat{\lambda}_{ij}$	Computed from \mathbf{a}_j and \mathbf{b}_i , using (1)–(3)	Not used

assumed to be from a Gaussian distribution, \mathcal{E}^2 is the log-likelihood of the estimated parameters.

This objective function combines the projection error from a calibrated projection, range error from the measured range, and projection error from a matching image patch in a neighboring image. The projection errors are implicitly a function of both the 3-D points and the imagery. Table I gives a summary of each value used in the optimization. The objective function is minimized by adjusting the pose values \mathbf{a}_j and points \mathbf{b}_i using established bundle adjustment optimization techniques. For clarity, we do not write the objective function explicitly in terms of \mathbf{a}_j and \mathbf{b}_i , but this can be done by substituting (1)–(3).

The specific optimization technique used for these experiments is described in Section III-C, which takes the advantage of the sparse relationships in the system. Although not required in the optimization, both equality and inequality constraints could be included in the optimization to represent other information such as ground control points.

The final texture with the calibrated projections (for example, in UV coordinates) of the optimized \mathbf{b}_i points form the TDSM, as shown in Fig. 2. The TDSM is a mesh with an overlaid texture.

III. SIMULATION RESULTS

A. Simulated Terrain, UAS Flight Path, and Texel Camera Data

Preliminary results for this algorithm were first discussed by Bybee and Budge [19] using the data from a handheld texel camera and a model scene. These results were visually promising, but ground truth information was not known exactly in the experimental setup and the step for fusing multiple digital images into the final texture had not been fully developed. A texel sensing system with realistic terrain and ground truth data was simulated using Open Source Robotics Foundation's Gazebo (described in [47]). This simulation models the flight of a small UAS with a texel camera. To the best of our knowledge, there is no existing data set of the fused LiDAR and aerial imagery using the texel camera calibration. Hence, Gazebo is used to provide a realistic data set.

The actual elevation data and aerial photographs were used to create a world in Gazebo for this experiment, courtesy of the U.S. Geological Survey.¹ The location is a section of Calls Fort Canyon near Honeyville, Utah, USA, approximately 1 km × 1 km, with a 3993 × 3992 pixel resolution aerial photographs and a registered 2049 × 2049 resolution height map in raster format created from a high-density LiDAR point cloud. In Gazebo, the controlling texel camera pose accurately

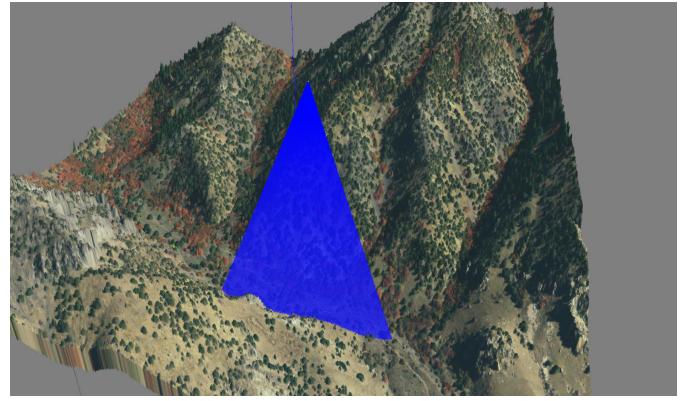


Fig. 4. Example data collection in Gazebo. Blue rays: LiDAR field of view on a simulated UAS. The texel camera (the LiDAR sensor and digital camera) is found at the origin of the rays. The terrain is shown, with the flight path toward the mountain.

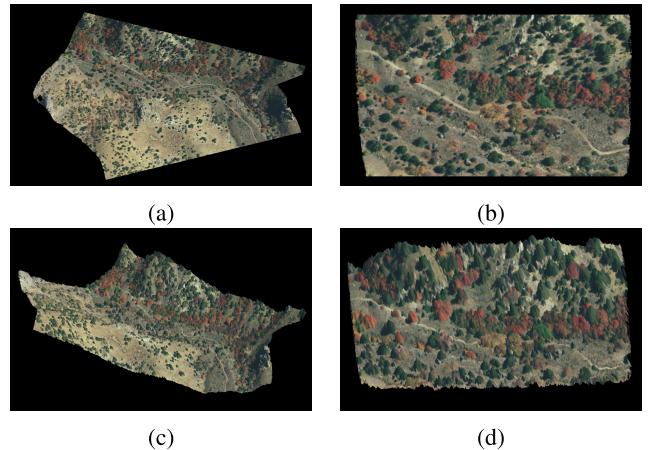


Fig. 5. Terrain below the UAS for different flight paths. (a) and (c) Result from a banked 25° right turn toward the mountains, with 1.5 m LiDAR GSD. (b) and (d) Result from a flight straight toward the mountains, with 0.5 m LiDAR GSD. (a) and (b) Views from directly above.

is available, allowing for realistic flight paths, with access to the ground-truth data. Fig. 4 shows a screenshot of a UAS texel camera simulation in Gazebo.

Eight UAS flight paths were simulated in Gazebo. This allowed the exploration of the performance of the algorithm with straight and turning scenarios, as well as approximate LiDAR ground sampling distances (GSDs) of 1.5, 1.0, 0.67, and 0.5 m in the cross-track direction. For straight flights, the texel camera maintained an approximately constant altitude above the ground level as the UAS flew toward the mountains. The turning path started up the canyon and performed a banked turn of approximately 25° to the right approaching the mountains. Examples of the terrain below the UAS are shown in Fig. 5, and the data set details are found in Table II.

¹<http://www.usgs.gov>

TABLE II
DATA SETS COLLECTED IN GAZEBO SIMULATION

Dataset	Nadir Ground Pixel Size (m)	LiDAR GSD (image pixels)	Sensor Field of View (°)	Total Number of LiDAR Points
1.5 m GSD with Turn	0.27	5.28	40	79,745
1.5 m GSD Straight	0.27	5.28	40	106,975
1.0 m GSD with Turn	0.25	3.82	35	107,736
1.0 m GSD Straight	0.25	3.82	35	107,736
0.67 m GSD with Turn	0.15	4.34	35	95,073
0.67 m GSD Straight	0.15	4.34	35	108,790
0.5 m GSD with Turn	0.10	4.79	30	86,229
0.5 m GSD Straight	0.10	4.79	30	95,667

Creating a texel camera in Gazebo is straightforward with its laser ray sensor and calibrated camera sensor models. The calibrated camera sensor in Gazebo is calibrated by virtue of its known parameters (pixel resolution and field of view) and an ideal pinhole model. No lens distortion is simulated. The laser ray sensor is similarly calibrated. Each digital image has a resolution of 2048×350 pixels. The LiDAR scanner intersects the middle of the image horizontally. In this Simulation, all points in a single laser scan are captured simultaneously, effectively simulating a time-of-flight array. The noiseless pose information is recorded alongside the LiDAR and imagery.

Noise is added to the Gazebo sensor measurements simulating the effects of a real-world sensor. Noise levels for the navigation system are designed to reflect accuracies of low-cost GPS and IMU measurements available on the market. Zero-mean Gaussian noise is added to UAS navigation measurements and LiDAR range measurements. The noise added to all attitude measurements consists of $\sigma_{\text{pitch}} = \sigma_{\text{roll}} = 0.1^\circ$ and $\sigma_{\text{yaw}} = 0.3^\circ$. The position error is of two types: error typical of GPS measurements, $\sigma_x = \sigma_y = \sigma_z = 2.5$ m (referred to as GPS noise), and lower error typical of differential GPS (DGPS) measurements, $\sigma_x = \sigma_y = \sigma_z = 0.1$ m (referred to as DGPS noise). The LiDAR range measurement noise levels have a $1-\sigma$ error of 0.05 meters for all experiments, representative of low-cost LiDAR units on the market currently. There are three data sets for each experiment: the ground truth (obtained directly from Gazebo), the noisy TDSM (the noise above added to the ground truth measurements), and the optimized TDSM (the noisy TDSM after our algorithm is applied).

B. TDSM Accuracy Metrics

The quality and accuracy measure for a TDSM depends on the use-case and is often hard to quantify between disciplines. Some might be interested in volume measurements, others in distances, some in slope, and others simply for esthetics. Some accuracy measures for DSMs are described in the literature (see [48]–[50]). The objective function presented in this paper represents a quantifiable value for the quality of the TDSM in terms of relative position and the size of geographical features,

important in volume, distance, or slope measurements. However, to cross validate the objective function, we propose an accuracy metric for TDSMs: computing the statistics of distance errors across the scene. This accuracy metric incorporates both the imagery and point cloud information because of the texel camera calibration. Distance errors directly relate to spatial accuracy, making this metric relevant to real-world applications.

The quality of the optimized TDSM is measured using the statistics of the error in distances between randomly selected points in the ground truth data and the TDSMs before and after optimization. In each data set, $N_{\text{points}} = 2000$ 3-D points are randomly selected. Each of these points is identified in the ground truth TDSM, the noisy TDSM, and the optimized TDSM. There are $D = \binom{N_{\text{points}}}{2} = 1.999 \times 10^6$ distance measurements in this selection of points. The error for each of the D measurements is given by the difference between the Euclidean distance between each two points in the optimized or noisy data and the Euclidean distance between the same two points in the ground truth data. The mean error and the error standard deviation are calculated for each of the data set-noise combinations.

Distance error is used instead of the direct point-to-point error because the objective function used in this work does not attempt to register to any reference points in space. A shift in the entire TDSM relative to ground truth would result in large error even though the distance between points within the TDSM may have low error.

C. Optimization Technique

The sparse Levenberg–Marquardt algorithm (LMA) described in [41] is employed for the optimization as many relationships between \mathbf{a}_j and \mathbf{b}_i do not exist and, thus, do not contribute any error to the objective function. The LMA is initialized using the measured range, calibrated projections, and measured poses, and inverting (1)–(3) to find the initial \mathbf{b}_i . To further improve computational load, the relationships between \mathbf{a}_j and \mathbf{b}_i for swaths acquired more than 30 swaths before or after the current swath are not considered. The LMA makes use of closed-form Jacobian

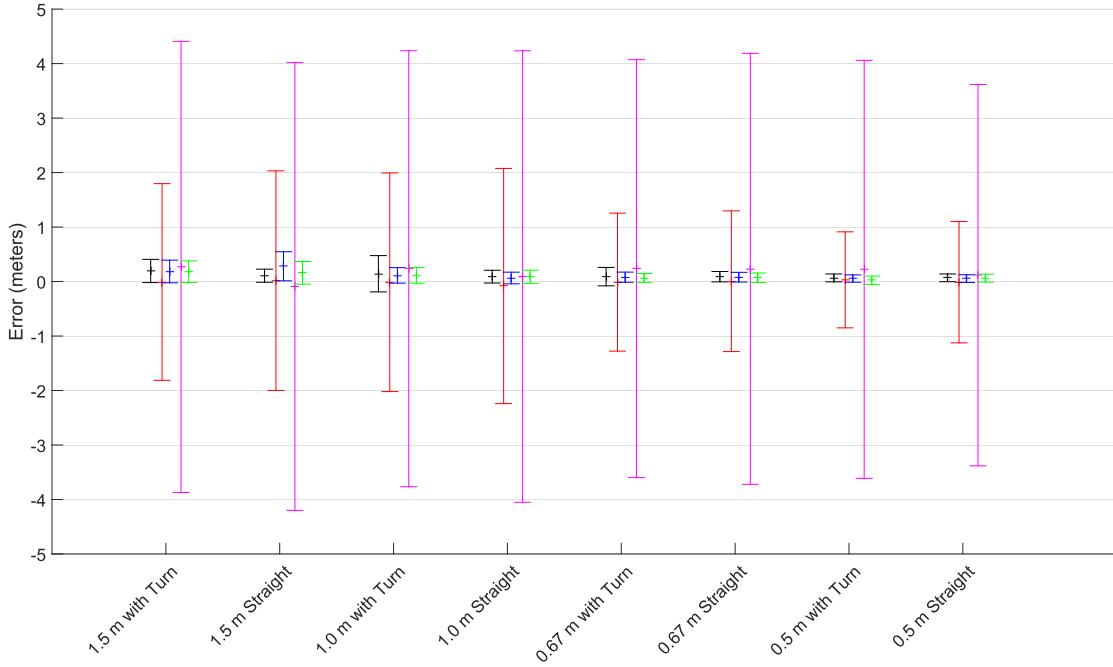


Fig. 6. Noisy and optimized $1-\sigma$ error distributions. The error bars represent the $\pm 1-\sigma$ bounds around the mean distance error. Each of the eight data sets has five error bars from left to right: “optimized” ground truth (black), DGPS noisy (red), DGPS optimized (blue), GPS noisy (magenta), and GPS optimized (green). Best viewed in color.

derivatives of \mathcal{E}^2 calculated with respect to \mathbf{a}_j and \mathbf{b}_i , shown in (5) and (6), respectively. A_{ij} is a 3×7 matrix and B_{ij} is a 3×3 matrix, and specific expressions for each entry can be found using the chain rule on (1)–(3). For the terms in the objective function not requiring the computation of λ_{ij} , the corresponding row in the Jacobians are zero

$$A_{ij} = \frac{\partial}{\partial \mathbf{a}_j} \begin{bmatrix} x_{ij} - \hat{x}_{ij} \\ y_{ij} - \hat{y}_{ij} \\ \lambda_{ij} - \hat{\lambda}_{ij} \end{bmatrix} \quad (5)$$

$$B_{ij} = \frac{\partial}{\partial \mathbf{b}_i} \begin{bmatrix} x_{ij} - \hat{x}_{ij} \\ y_{ij} - \hat{y}_{ij} \\ \lambda_{ij} - \hat{\lambda}_{ij} \end{bmatrix}. \quad (6)$$

D. Registration Results and Discussion

Our method does not assume overlap between LiDAR measurements, and there are no other known point cloud registration methods with which to compare the results of this algorithm fairly. Similarly, the use of fused LiDAR and imagery does not allow for any one-to-one comparison with reconstruction techniques using photogrammetry.

The mean error with $1-\sigma$ error bars for each combination of noise levels and data set are shown in Fig. 6, with numerical results listed in Table III.

The error bars represent a $1-\sigma$ error about the mean. The “optimized” ground truth (black in Fig. 6) is a measure of the error inherent in the method, which is introduced in the resulting TDSM. It is the result of running the algorithm on the ground truth data and is largely due to the integer pixel-level matching in the selection of common projection points and the nondeterministic effects of RANSAC matching in generating pairwise homographies. In fact, we expect much

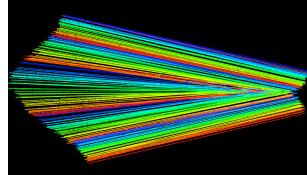
of the variation between the “optimized” ground truth and the other experiments to be due to these causes. The mean bias is about one nadir ground pixel size (see Table II) in the positive direction, indicating on average, distances across the optimized TDSM are slightly farther apart than ground truth. This might be explained by the fact that projection errors of one pixel at the edges of the images represent larger errors at greater distances from the sensor than one-pixel projection errors at nadir.

This method significantly reduces the distance error variance with each noise level. Two important observations arise. First, the optimized $1-\sigma$ error of both DGPS and GPS noise levels is approximately the same as the inherent algorithm error, suggesting the noise level is irrelevant over a large range of position error (2.5 m versus 0.1 m). Second, the $1-\sigma$ error is generally less than the size of an image pixel on the ground (Table II), implying if the resolution of the digital image sensor in the texel camera is increased (without changing the LiDAR sensor configuration), the distance error in the TDSM will reduce accordingly. This idea is further supported by the observation that the error reduces as the size of a pixel reduces: The error in the 1.5-m LiDAR GSD case is of approximately the same fraction of a pixel as the error in the 0.5-m LiDAR GSD case.

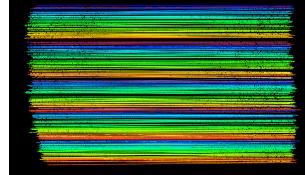
Further insight is gained by viewing the TDSMs directly. Fig. 7 shows the power of the optimization in Correcting the errors caused by pose sensor noise. Fig. 7 was created by cyclically coloring each swath LiDAR data according to acquisition time. Note the spacing errors, nonparallel swaths, and out-of-order placement of each line of LiDAR measurements before optimization for both the turn and straight paths, and the correct reordering of the swaths after optimization.

TABLE III
ERROR MEASURES FOR NOISY AND OPTIMIZED TDSMs

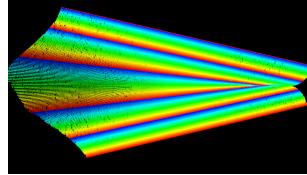
Noise Level	Noisy Mean (m)	Noisy 1- σ (m)	Optimized Mean (m)	Optimized 1- σ (m)	Optimized 1- σ (image pixels)
1.5 m GSD with Turn					
Ground Truth			0.19	0.21	0.79
DGPS	-0.01	1.80	0.18	0.21	0.77
GPS	0.27	4.14	0.18	0.20	0.74
1.5 m GSD Straight					
Ground Truth			0.11	0.12	0.44
DGPS	0.02	2.02	0.28	0.27	0.99
GPS	-0.09	4.11	0.16	0.21	0.77
1.0 m GSD with Turn					
Ground Truth			0.14	0.33	1.33
DGPS	-0.01	2.01	0.11	0.14	0.57
GPS	0.23	4.00	0.11	0.14	0.57
1.0 m GSD Straight					
Ground Truth			0.09	0.12	0.47
DGPS	-0.08	2.16	0.07	0.11	0.43
GPS	0.09	4.14	0.09	0.12	0.48
0.67 m GSD with Turn					
Ground Truth			0.09	0.17	1.13
DGPS	-0.01	1.27	0.08	0.09	0.62
GPS	0.24	3.83	0.07	0.08	0.54
0.67 m GSD Straight					
Ground Truth			0.09	0.10	0.63
DGPS	0.01	1.29	0.08	0.09	0.59
Large	0.23	3.96	0.07	0.09	0.59
0.5 m GSD with Turn					
Ground Truth			0.07	0.07	0.72
DGPS	0.03	0.88	0.06	0.07	0.67
GPS	0.22	3.84	0.03	0.08	0.76
0.5 m GSD Straight					
Ground Truth			0.07	0.07	0.71
DGPS	-0.01	1.11	0.05	0.07	0.70
GPS	0.12	3.50	0.06	0.07	0.73



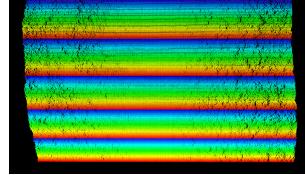
(a)



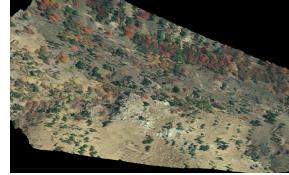
(b)



(c)



(d)



(a)



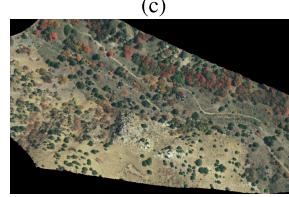
(b)



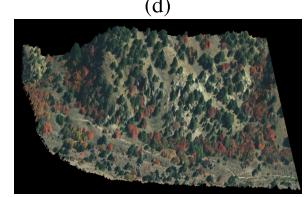
(c)



(d)



(e)



(f)

Fig. 7. LiDAR swaths before and after optimization. Swaths are viewed from above and are cyclically colored according to acquisition time. (a) and (c) Result from a banked 25° right turn toward the mountains, with 1.0-m LiDAR GSD. (b) and (d) Result from a flight straight toward the mountains, with 0.67-m LiDAR GSD. (a) and (b) Contain the GPS level of position noise. Best viewed in color.

The registration results for some of the final TDSMs are illustrated in Fig. 8. The GPS-level noise in the pose measurements results in a significant distortion of the TDSM as shown in Fig. 8(a) and (b), where the trees are significantly blurred and the path is gone. The optimized TDSMs in Fig. 8(c) and (d) are visually indistinguishable from the ground truth TDSMs in Fig. 8(e) and (f), respectively.

Fig. 8. Registration Results. (a), (c), and (e) Result from a banked 25° right turn toward the mountains, with 1.0-m LiDAR GSD. (b), (d), and (f) Result from a flight straight toward the mountains, with 0.67-m LiDAR GSD. (a) and (b) Contain GPS-level noise in the pose measurements. (c) and (d) Results of the registration algorithm. (e) and (f) Ground truth.

IV. CONCLUSION

The primary contribution of this paper is the triangulation technique using the fused LiDAR and digital imagery and

the associated objective function presented in Section II. This method uses both LiDAR measurements and image projections together in a single optimization algorithm aligning capture poses to ensure the LiDAR measurements are properly referenced. The quantitative and visual analysis of the method shows the impressive accuracy for the texel camera data acquired using low-cost navigation equipment typical of a small UAS. Of particular interest is this method's capability in registering texel swath captures forming a TDSM accurate to a digital image pixel level, implying higher resolution image cameras will allow more accurate LiDAR points in the reconstruction. Since high-resolution digital cameras are much less expensive than high-accuracy navigation systems, the goal of a low-cost system with high accuracy is met.

A second advantage of this method is the texture (imagery) overlaid on the 3-D point cloud is, by virtue of its sensor-level calibrated fusion, automatically registered at the subpixel level to the point cloud. Therefore, the scientific analysis of the TDSM is accurate, both in topology and in imagery, to the pixel level. This allows the analysis of topology such as watershed changes, riverbank erosion, and forest fire impacts at resolutions near the accuracy of the LiDAR measurements.

Further work may include optimizing the 3-D location of points not having distance measurements using information from nearby projections. Superresolution techniques in creating the texture can be applied. The investigation into other image and point matching techniques can be done to ensure more accurate common projection points. Effort should be made to identify and ignore outliers and to mitigate the effects of points far from nadir. Incorporating hard constraints into the optimization, such as surveyed ground control points, will also prove useful in the real-world scenarios.

ACKNOWLEDGMENT

The use of the Armadillo C++ Matrix Library is greatly appreciated as it enabled the ease of development code to implement the algorithm ([51]). The Open Source Robotics Foundation's Gazebo Simulation software was invaluable in simulating the texel camera in a realistic scenario.

REFERENCES

- [1] E. P. Baltsavias, "A comparison between photogrammetry and laser scanning," *ISPRS J. Photogramm. Remote Sens.*, vol. 54, nos. 2–3, pp. 83–94, Jul. 1999.
- [2] G. H. Schut, "An analysis of methods and results in analytical aerial triangulation," *Photogrammetria*, vol. 14, pp. 16–33, Jan. 1958.
- [3] P. Agouris and T. Schenk, "Automated aerotriangulation using multiple image multipoint matching," *Photogramm. Eng. Remote Sens.*, vol. 62, no. 6, pp. 703–710, Jun. 1996.
- [4] O. Küng, C. Strecha, A. Beyeler, J.-C. Zufferey, D. Floreano, P. Fua, and F. Gervais, "The accuracy of automatic photogrammetric techniques on ultra-light UAV imagery," in *Proc. UAV-G Unmanned Aerial Vehicle Geomatics*, 2011.
- [5] J. Vallet, F. Panissod, C. Strecha, and M. Tracol, "Photogrammetric performance of an ultra light weight swinglet UAV," *ISPRS-Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3822, p. C253, Sep. 2011.
- [6] A. Ahmad and H. Karimian, "Performance of aerial triangulation based on airborne GPS data," in *Proc. 11th South East Asian Surv. Congr. 13th Int. Surveyors' Congr. Innov. Towards Sustainability (11th SEASC)*, Jun. 2011.
- [7] F. He and A. Habib, "Automated relative orientation of UAV-based imagery in the presence of prior information for the flight trajectory," *Photogramm. Eng. Remote Sens.*, vol. 82, no. 11, pp. 879–891, Nov. 2016.
- [8] T. Schenk, "Towards automatic aerial triangulation," *ISPRS J. Photogramm. Remote Sens.*, vol. 52, no. 3, pp. 110–121, Jun. 1997.
- [9] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 92, pp. 79–97, Jun. 2014.
- [10] M. Uysal, A. S. Toprak, and N. Polat, "DEM generation with UAV photogrammetry and accuracy analysis in Sahitler hill," *Measurement*, vol. 73, pp. 539–543, Sep. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0263224115003243>
- [11] B. Ruzgienė, T. Berteška, S. Gečyte, E. Jakubauskienė, and V. C. Aksmitauskas, "The surface modelling based on UAV photogrammetry and qualitative estimation," *Measurement*, vol. 73, pp. 619–627, Sep. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0263224115002316>
- [12] A. Wehr and U. Lohr, "Airborne laser scanning—An introduction and overview," *ISPRS J. Photogramm. Remote Sens.*, vol. 54, nos. 2–3, pp. 68–82, Jul. 1999.
- [13] D. Roca, J. Martínez-Sánchez, S. Lagüela, and P. Arias, "Novel aerial 3D mapping system based on UAV platforms and 2D laser scanners," *J. Sensors*, vol. 2016, Dec. 2016, Art. no. 4158370.
- [14] G. Tao and Y. Yasuoka, "Combining high resolution satellite imagery and airborne laser scanning data for generating bareland DEM in urban areas," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 30, Feb. 2002.
- [15] M. Lim, D. N. Petley, N. J. Rosser, R. J. Allison, A. J. Long, and D. Pybus, "Combined digital photogrammetry and time-of-flight laser scanning for monitoring cliff evolution," *Photogramm. Rec.*, vol. 20, no. 110, pp. 109–129, Jun. 2005.
- [16] D. Leckie, F. Gougeon, D. Hill, R. Quinn, L. Armstrong, and R. Shreean, "Combined high-density lidar and multispectral imagery for individual tree crown analysis," *Can. J. Remote Sens.*, vol. 29, no. 5, pp. 633–649, Oct. 2003.
- [17] E. K. Forkuo and B. King, "Automatic fusion of photogrammetric imagery and laser scanner point clouds," *Int. Arch. Photogramm. Remote Sens.*, vol. 35, pp. 921–926, Jul. 2004.
- [18] L. C. Chen, T.-A. Teo, Y.-C. Shao, Y.-C. Lai, and J.-Y. Rau, "Fusion of LiDAR data and optical imagery for building modeling," *Int. Arch. Photogramm. Remote Sens.*, vol. 35, no. B4, pp. 732–737, Jul. 2004.
- [19] T. C. Bybee and S. E. Budge, "Textured digital elevation model formation from low-cost UAV LADAR/digital image data," *Proc. SPIE*, vol. 9465, May 2015, Art. no. 94650H. doi: [10.1117/12.2177474](https://doi.org/10.1117/12.2177474).
- [20] B. Wu and S. Tang, "Review of geometric fusion of remote sensing imagery and laser scanning data," *Int. J. Image Data Fusion*, vol. 6, no. 2, pp. 97–114, Apr. 2015.
- [21] P. Rönnholm, E. Honkavaara, P. Litkey, H. Hyppä, and J. Hyppä, "Integration of laser scanning and photogrammetry," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 39, pp. 355–362, Sep. 2007.
- [22] H. Badino, D. Huber, and T. Kanade, "Integrating LiDAR into stereo for fast and improved disparity computation," in *Proc. IEEE Int. Conf. 3D Imag., Modeling, Process., Vis. Transmiss. (3DIMPVT)*, May 2011, pp. 405–412.
- [23] A. S. Gneeniss, "Integration of LiDAR and photogrammetric data for enhanced aerial triangulation and camera calibration," Ph.D. dissertation, Newcastle Univ., Newcastle upon Tyne, U.K., 2014.
- [24] F. C. Nex and F. Rinaudo, "LiDAR or photogrammetry? Integration is the answer," *Italian J. Remote Sens.*, vol. 43, no. 2, pp. 107–121, Jan. 2011.
- [25] E. Mitishita, P. Debiasi, F. Hainosz, and J. Centeno, "Calibration of low cost digital camera using data from simultaneous LiDAR and photogrammetric surveys," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 1, pp. 133–138, Jul. 2012.
- [26] K. Di, Q. Zhao, W. Wan, Y. Wang, and Y. Gao, "RGB-D SLAM based on extended bundle adjustment with 2D and 3D information," *Sensors*, vol. 16, no. 8, p. 1285, Aug. 2016.

- [27] E. G. Parmehr, C. S. Fraser, C. Zhang, and J. Leach, "Automatic registration of optical imagery with 3D LiDAR data using statistical similarity," *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 28–40, Feb. 2014.
- [28] M. Brell, C. Rogass, K. Segl, B. Bookhagen, and L. Guanter, "Improving sensor fusion: A parametric method for the geometric coalignment of airborne hyperspectral and lidar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3460–3474, Jun. 2016.
- [29] M. Brell, K. Segl, L. Guanter, and B. Bookhagen, "Hyperspectral and lidar intensity data fusion: A framework for the rigorous correction of illumination, anisotropic effects, and cross calibration," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2799–2810, May 2017.
- [30] C. Paris and L. Bruzzone, "A three-dimensional model-based approach to the estimation of the tree top height by fusing low-density LiDAR data and very high resolution optical images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 467–480, Jan. 2015.
- [31] E. G. Parmehr, C. S. Fraser, and C. Zhang, "Automatic parameter selection for intensity-based registration of imagery to LiDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7032–7043, Dec. 2016.
- [32] J. Lee, X. Cai, C.-B. Schönlieb, and D. A. Coomes, "Nonparametric image registration of airborne LiDAR, hyperspectral and photographic imagery of wooded landscapes," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6073–6084, Nov. 2015.
- [33] B. Jutzi, M. Weinmann, and J. Meadow, "Weighted data fusion for UAV-borne 3D mapping with camera and line laser scanner," *Int. J. Image Data Fusion*, vol. 5, no. 3, pp. 226–243, Jul. 2014.
- [34] J.-S. Yoon and J. Shan, "Combined adjustment of MOC stereo imagery and MOLA altimetry data," *Photogramm. Eng. Remote Sens.*, vol. 71, no. 10, pp. 1179–1186, Oct. 2005.
- [35] B. Wu, J. Guo, Y. Zhang, B. A. King, Z. Li, and Y. Chen, "Integration of Chang'E-1 imagery and laser altimeter data for precision lunar topographic modeling," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4889–4903, Dec. 2011.
- [36] R. T. Pack, P. Irsaelsen, and K. Sealy, "A co-boresighted synchronized lidar/EO imager for creating 3D images of dynamic scenes," *Proc. SPIE*, vol. 5791, pp. 42–50, May 2005.
- [37] S. E. Budge and N. S. Badamikar, "Calibration method for texel images created from fused flash lidar and digital camera images," *Opt. Eng.*, vol. 52, no. 10, Oct. 2013, Art. no. 103101, doi: 10.1117/1.OE.52.10.103101.
- [38] P. J. Besl and D. N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [39] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IROS*, vol. 3, Oct. 2003, pp. 2743–2748.
- [40] A. Habib, K. I. Bang, A. P. Kersting, and J. Chow, "Alternative methodologies for LiDAR system calibration," *Remote Sens.*, vol. 2, no. 3, pp. 874–907, Mar. 2010.
- [41] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [42] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, Sheffield, U.K.: University of Sheffield Printing Office, Aug. 1988, pp. 147–151.
- [43] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 404–417.
- [44] L. M. Fonseca and B. S. Manjunath, "Registration techniques for multisensor remotely sensed imagery," *Photogramm. Eng. Remote Sens.*, vol. 62, no. 9, pp. 1049–1056, Sep. 1996.
- [45] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [46] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 4, no. 4, pp. 629–642, 1987.
- [47] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, Sep./Oct. 2004, pp. 2149–2154.
- [48] F. Agüera-Vega, F. Carvajal-Ramírez, and P. Martínez-Carricundo, "Accuracy of digital surface models and orthophotos derived from unmanned aerial vehicle photogrammetry," *J. Surveying Eng.*, vol. 143, no. 2, Aug. 2017, Art. no. 04016025. [Online]. Available: <http://ascelibrary.org/doi/abs/10.1061/%28ASCE%29SU.1943-5428.0000206>
- [49] M. Jaud, S. Passot, R. Le Bivic, C. Delacourt, P. Grandjean, and N. Le Dantec, "Assessing the accuracy of high resolution digital surface models computed by PhotoScan and MicMac in sub-optimal survey conditions," *Remote Sens.*, vol. 8, no. 6, p. 465, Jun. 2016.
- [50] S. Gindraux, R. Boesch, and D. Farinotti, "Accuracy assessment of digital surface models from unmanned aerial vehicles' imagery on glaciers," *Remote Sens.*, vol. 9, no. 2, p. 186, Feb. 2017.
- [51] C. Sanderson and R. Curtin, "Armadillo: A template-based C++ library for linear algebra," *J. Open Source Softw.*, vol. 1, no. 2, pp. 26–32, Jun. 2016.



Taylor C. Bybee received the B.S. and M.S. degrees in electrical engineering from Utah State University, Logan, UT, USA, in 2015, where he researched the registration of fused LiDAR and optical imagery to form textured digital surface models.

He is an Engineering Technical Lead in robotic perception with Autonomous Solutions, Inc., Petersboro, UT, USA. His research interests include robotics sensing, motion planning and control, signal and image processing, and mapping.



Scott E. Budge (SM'00) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Brigham Young University, Provo, UT, USA, in 1984, 1985, and 1990, respectively.

In 1989, he joined the Faculty with Utah State University, Logan, UT, USA, where he has been involved in research into image data compression algorithms for transmission systems and space-based observation platforms. He was also involved in the development of high-performance image processing algorithms designed for implementation in hardware and VLSI systems. His research interests include signal and image processing and sensor systems, especially the methods for exploitation of full-waveform LADAR and EO images for 3-D image applications and pattern recognition.