GPS-DENIED NAVIGATION USING LOCATION ESTIMATION AND TEXEL IMAGE CORRECTION

by

Nikolas I. Jensen

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Scott E. Budge, Ph.D. Major Professor	Jacob Gunther, Ph.D. Committee Member
Don Cripps, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2024

ABSTRACT

GPS-denied navigation using location estimation and texel image correction

by

Nikolas I. Jensen, Master of Science

Utah State University, 2024

Major Professor: Scott E. Budge, Ph.D.
Department: Electrical and Computer Engineering

In recent years, the utilization of Unmanned Aerial Vehicles (UAVs) has witnessed a remarkable surge, finding their way into an extensive range of applications. Some of these applications include land surveying methodologies, remote sensing techniques, or even tasks involving aerial manipulation. Smaller UAVs are also suitable for many of these tasks, due to their low cost, size, accessibility for training, and maneuverability in complex locations. Thus, UAV-based surveying and surveillance can be cheaper and faster than classic methods. Due to these factors, localization techniques can become especially useful for UAVs as they can help locate features of surveyed areas or locate the UAV when needed.

This thesis presents a localization system for UAVs, specifically designed for small UAVs to be used in a local area, by using localization estimation algorithms. Localization techniques classically rely on a combination of Inertial Measurement Unit (IMU) sensors and Global Positioning System (GPS). However, GPS can be costly, and is prone to disconnection due to jamming and dependency on the sensor's environment. This thesis proposes methods of localization estimation without reliance on the GPS to introduce a overall low-cost system. To do so, methods of estimation using the systems IMU are implemented with the addition of a mounted texel camera. The texel camera consists of the fusion of camera and Light Detection and Ranging (LiDAR) sensors. This system utilizes Error-State

Kalman Filtering (ESKF) and the methods of camera to LiDAR sensor fusion to correct for error propagations in the aerial vehicle's estimated location. By comparison to an attached GPS sensor, this results in a robust localization system that estimates at a reliable accuracy close to sub-GPS position estimates.

(54 pages)

PUBLIC ABSTRACT

GPS-denied navigation using location estimation and texel image correction

Nikolas I. Jensen

In recent years, the use of small drones, also categorized as small Unmanned Aerial Vehicles (sUAV), has surged. They are used for tasks like surveying land, collecting data from a distance, and performing maneuvers for military operations. These drones are popular because they are affordable, small, easy to use, and can navigate well in complex areas. These factors make them a cheap and quick option for tasks like surveying and surveillance when compared to traditional methods.

This thesis introduces a system that uses algorithms to figure out where the drone is. Typically, this relies on sensors and GPS, but GPS can sometimes be unreliable for certain uses. To address this, the system uses complex algorithms by using motion sensors along with a camera and a specialized mapping sensor. By combining these technologies, the system can estimate the drone's location. Compared to relying solely on GPS, this system provides a reliable estimate close to GPS.

To my loving partner, Sage.

ACKNOWLEDGMENTS

I express my deepest appreciation to my beloved partner, Sage, for her patience towards me and my research. The countless hours and nights dedicated to this work would not have been achievable without her invaluable support. I would also like to direct my gratitude to Dr. Scott Budge, my major professor, for his instruction and help throughout this research. I am also grateful for family and friends for their support and care. I acknowledge this research was funded predominantly by L3 Harris. I shall also recognize AggieAir for their assistance in data capture and supply of aerial vehicles, as well as Dr. Droge for the use of his motion capture studio.

Nikolas I. Jensen

CONTENTS

Page

LIST OF FIGURES

Figure | Page

ACRONYMS

CAIL	Center for Advanced Imaging LiDAR
DEM	Digital Elevation Map
EO	Electro-Optical
ESEKF	Error-State Extended Kalman Filtering
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LiDAR	Light Detection and Ranging
LM	Levenberg–Marquardt
NCC	Normalized Cross Correlation
PnP	Perspective-n-Point
RANSAC	Random Sample Consensus
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
sUAV	Small Unmanned Aerial Vehicle
TDSM	Textured Digital Surface Model
TIN	Triangulated Interconnected Network
UAV	Unmanned Aerial Vehicle

CHAPTER 1

INTRODUCTION

In the past decade, there has been a significant rise in the popularity of Small Unmanned Aerial Vehicles (sUAV) across a wide range of applications [1]. Many of these applications require precise location information, which necessitates the implementation of localization techniques. By implementing localization techniques, the sUAV can accurately determine the positions of landmarks, objects, or potential hazards, allowing users to utilize this information according to their specific needs. Another possibility is localizing the sUAV itself, which can establish the sUAV's flight paths, measure distances covered, and even aid in calculating velocity [2].

Traditionally, localization techniques have relied on a combination of Inertial Measurement Unit (IMU) sensors and Global Positioning System (GPS) technology [3–5]. The IMU acts as a basis for movement as it contains information on the systems acceleration, orientation, angular velocity, and compass heading. These IMU measurements and the GPS data are then implemented into a designed Kalman filter for precise navigation and location estimation. However, GPS can be susceptible to disconnection and inaccuracies due to factors such as jamming and environmental conditions [6]. Therefore, this study aimed to explore alternative localization estimation methods that do not depend on GPS.

Specifically, the research utilized the acceleration and angular velocity measurements from an IMU sensor to integrate them into a dead-reckoning location estimator. Unfortunately, it is known that IMU sensors tend to experience drift over time, leading to increasing errors in position estimation and navigation [7].

To address these challenges, this research introduces two key approaches. Firstly, it leverages Error-State Extended Kalman Filtering (ESEKF) to track IMU sensor measurements. The designed ESEKF serves as a foundation for localizing the sUAV system during flight operations. Secondly, the ESEKF incorporates fused Light Detection and Ranging

(LiDAR) data and digital images to generate position corrections, enhancing the accuracy of the localization process. The LiDAR data comes from a remote sensing technology that uses laser pulses to measure distances to objects and create precise three-dimensional representations of the surrounding environment. Operating at a higher frequency in a rotational field of view, the LiDAR unit can rapidly deliver precise mapping of positional features, enhancing its capabilities for localization techniques.

1.1 Previous Work

1.1.1 GPS-denied navigation

Achieving accurate navigation in GPS-denied scenarios has been a hot research topic, as this is a very common problem that can assist in military and civilian domains. Use cases include land surveillance, weather monitoring, autonomous navigation, and even underwater/underground guidance [8–11]. Of these topics, it would be desired to create a system that is as low-cost and accurate as possible, ensuring accessibility and reliability for any use case.

In Brigadnov et al. study, the desire for navigation in a GNSS-denied enviornment was studied to ensure localization and mapping for underground mining operations. The implementation of this underground navigation allowed for accurate localization and mapping, improving the accuracy and efficiency of the mining process and surveying. Though, this work was accurate to within a local area using only wheel odometery.

As for Chisholm et al., the use of GPS-denied navigational reliance was needed for below-canopy forest surveying and management due to environmental factors such as GPS signal blocking tree canopies [12]. This work proved useful, but required multiple passes of a local area for reliable accuracy.

1.1.2 Visual-LiDAR SLAM

Common approaches to visual navigation in a 3D setting is the implementation of Simultaneous Localization and Mapping (SLAM). This process follows taking visual information of the system and applying it towards location estimation and mapping the environment during movement [13, 14]. Typically, SLAM systems do not rely on GPS, making the operation of SLAM a GPS-denied operation. Though, due to the flexibility of SLAM, there are many branches, including designs that are GPS reliant implementations [15]. Of these branches, it shall be noted the similarities of visual-LiDAR SLAM compared to the research presented.

This thesis aligns closely with research on a visual-LiDAR-based SLAM system, presenting a SLAM-like approach to navigation and mapping through its localization and mapping processes. An illustrative instance of visual-LiDAR SLAM for sUAV systems was explored by Qian et al., emphasizing local mapping and navigation with solely visual and LiDAR information affixed to an sUAV [16]. Points of divergence between Qian's study and the current thesis include the incorporation of Inertial Measurement Unit (IMU) data for position estimation using Kalman filtering, a common aspect in SLAM designs, and the registration of Textured Digital Surface Models (TDSM). While Qian's research primarily centered on local area mapping through feature extraction from LiDAR point clouds and visual features, it was limited to flight distances of less than 250 meters using two distinct datasets.

Hemann et al. followed a method of GPS-denied localization using a visual SLAM-like method by creating a Digital Elevation Map (DEM) from LiDAR data and then correlating the map to pre-processed satellite images [17], while simultaneously adding position estimation from a Kalman filter. Similar work was created by Miller et al., which uses image feature tracking methods to aid in their navigation methods [18]. This work was mainly tested on 2D scenarios in straight hallways and infers to be expanded to a 3D setting in the future.

This thesis introduces a SLAM-inspired methodology where the estimated pose also enhances the precision of the TDSM registrations in the overall system. Conventional SLAM systems typically emphasize constructing a dense, geometric map of the environment, often represented as a point cloud or mesh. In contrast, this research generates mapping from a top-down perspective, situated 60 meters above the ground and capturing only windows of data. Although generated TDSMs follow a mapping technique, they diverge from the regular map generation in a typical SLAM approach, therefore giving the characterization of the proposed research as a SLAM-like approach.

1.2 Contribution

This thesis describes implementing a navigational system that does not rely on GPS measurements, but instead estimates position using IMU data combined with LiDAR-Photogrammetry fusion. The system is also designed with the goal of being low-cost, giving accessibility for more use cases. Similar to visual-LiDAR based SLAM, the research conducted offers a way of navigating in a GPS-denied scenario while concurrently generating a TDSM for the user. This design is able to process the estimated location for over 3 kilometers, while simultaneously registering the TDSMs in a timely manner.

The specially designed system offers a precise local navigational solution, completely independent of GPS usage. This innovative approach ensures both cost-effectiveness and accurate device positioning. Such advancements have wide-ranging applications and can prove highly advantageous in various scenarios. For instance, a silviculturist working with limited resources in a remote and mountainous region may need to monitor a forest stand. In this case, a local position estimation becomes crucial to confirm that the forester is observing the correct stand, prompting the need for a GPS-denied navigational system. Similarly, the system also serves military purposes by enabling the localization of sUAVs or desired objects in GPS-jammed environments, while providing the user with valuable visual data of the area through TDSM generation.

In this thesis, Chapter 2 will introduce the texel camera and provide the context for TDSM registration. Chapter 4 will detail the process of GPS-denied navigation, utilizing the registered TDSM and measured IMU data. Chapter 5 will showcase the data collected and the results of estimated positions. Lastly, Chapter 6 will present the concluding remarks and possible future work.

CHAPTER 2
TEXEL CAMERA OVERVIEW & BACKGROUND

The research for this thesis was designed off of prior work with the Center for Advanced Imaging LADAR (CAIL) [19–22]. This previous work involves the design of multi-sensor fusion for registering TDSMs from a sUAV payload containing a "texel" camera. The texel camera contains a LiDAR unit, an electro-optical (EO) camera, and an Inertial Navigation System (INS). The texel camera implemented in this study can be seen in Figure 2.1 where this camera would then be mounted to the sUAV pointing downwards as seen in Figure 2.2. This sUAV then surveys a designated local area by following a flight path and using the

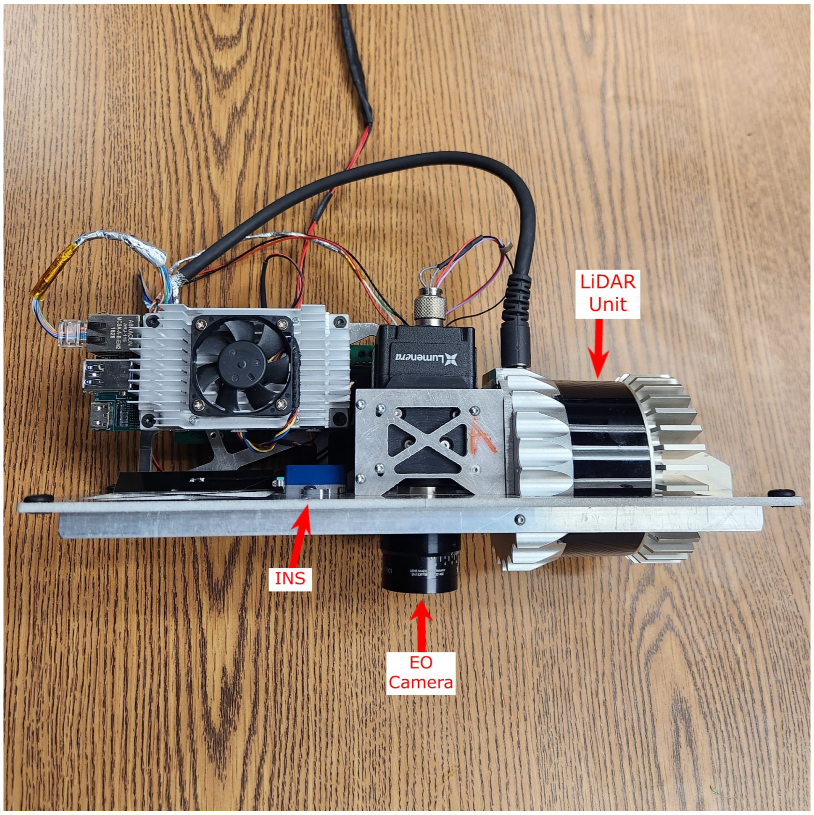


Fig. 2.1: Texel Camera with labeled mounted sensors.

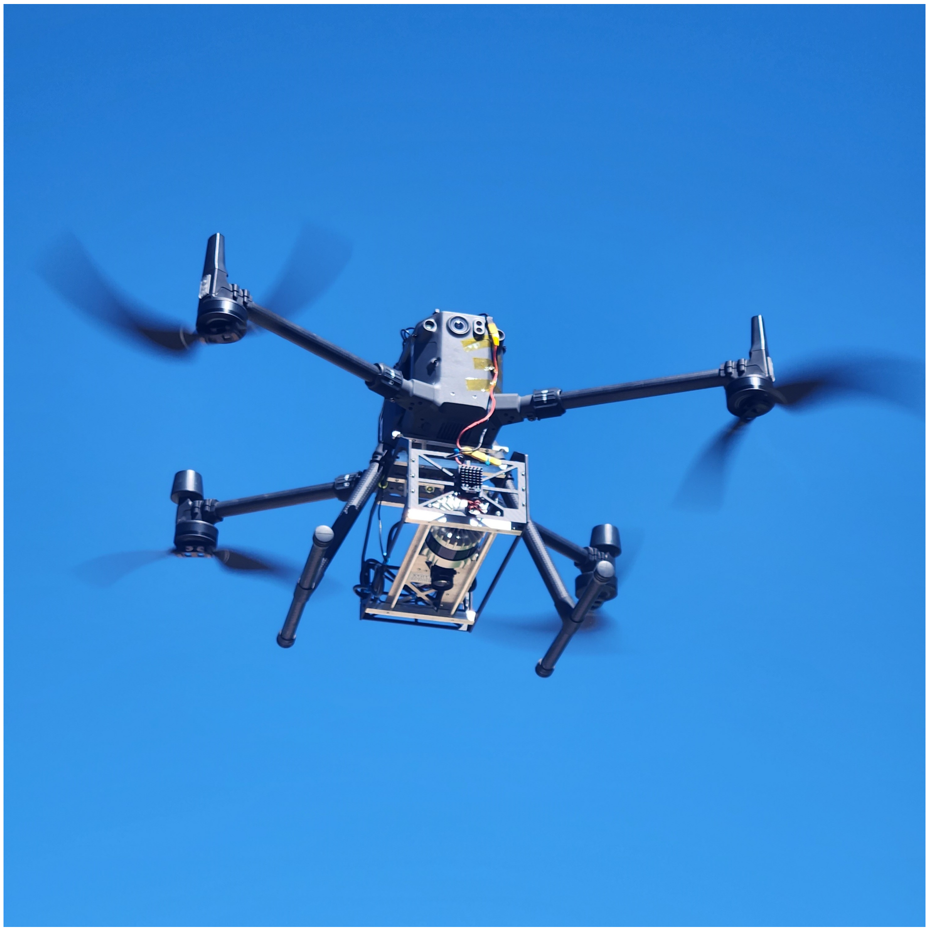
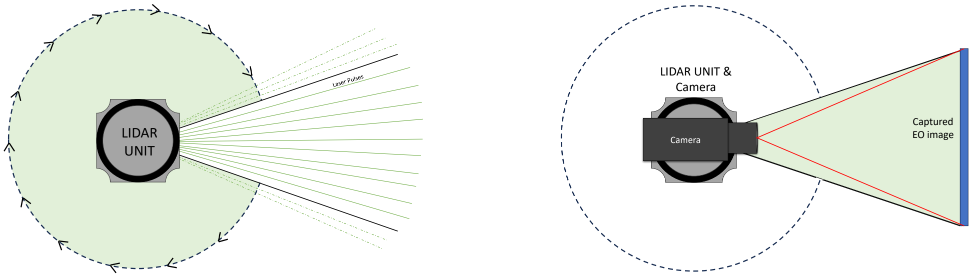


Fig. 2.2: sUAV with the mounted texel camera payload.

texel camera to generate data to be registered into a TDSM. The result of this process is a 3D surface that the user can then use for multiple applications such as forest management, emergency response, infrastructure, and defense [23–25].

2.1 Texel Camera Sensor Fusion & Textured Digital Surface Model Generation

While in aerial flight, the LiDAR unit undergoes rotation and collects data from a specific window of its rotation as seen in the example in Figure 2.3a resulting in a generated 3D point cloud. At the center of this designated LiDAR window, an image is acquired using the EO camera. This image capture with relation to the LiDAR rotation can be seen in Figure 2.3b. The acquisition of both imagery and point clouds creates what is called a texel image. This synchronization process ensures a match between a set of LiDAR points and the captured image. To visualize, the resulting window and image for a single rotation are seen Figure 2.4.



(a) Visual depiction of the implemented windowed LiDAR unit.

(b) Visual depiction of the synchronized image capture with the LiDAR window.

Fig. 2.3: LiDAR-to-Image window capture visualization

With the alignment calibrated, multiple texel images can be fused together by projecting the collected LiDAR point clouds to a similar plane, creating a visual flight-path of LiDAR data. To impose the imagery to the model, a 3D surface can be added to the point cloud by applying a Triangulated Interconnected Network (TIN).

TIN surfacing takes LiDAR data points and connects them to form a network of non-overlapping triangles. Linear interpolation within these triangles then enables the estimation of elevation values at any of these points. The triangulated network can be seen with the wireframe in Figure 2.5b. Lastly, the generated triangles are filled with the captured imagery. The TIN surface result is a flight-path of texel images that can be seen in Figure 2.5c below.

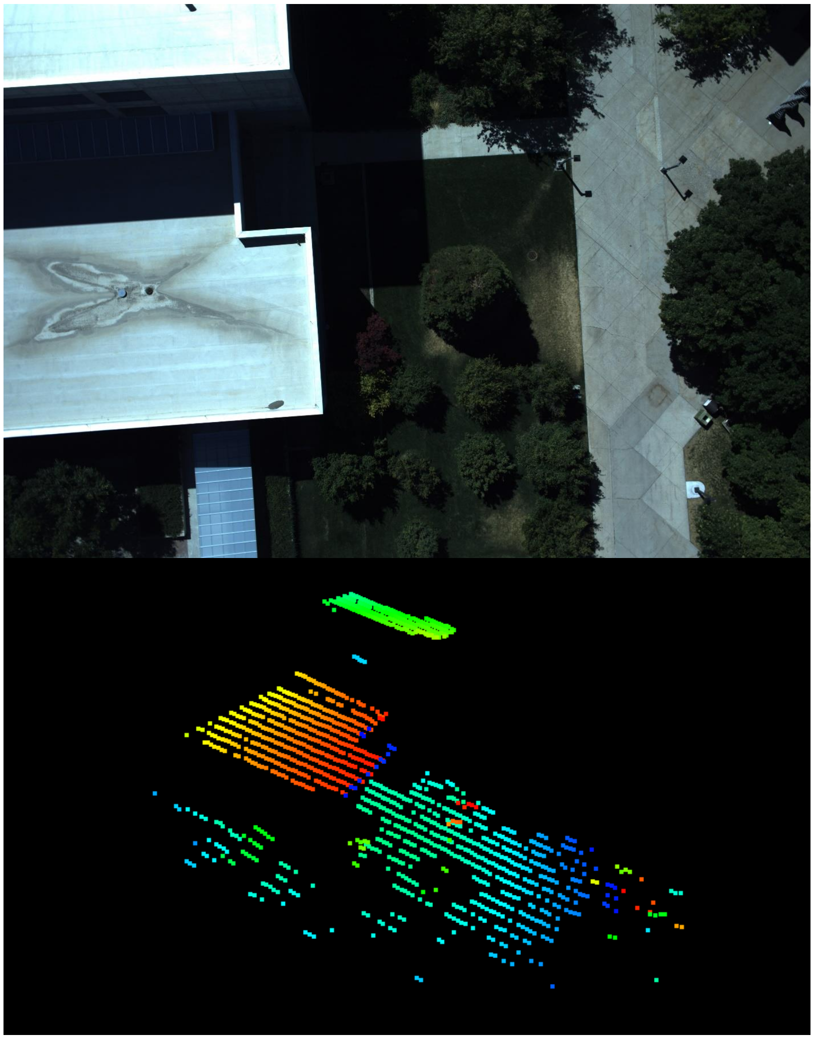
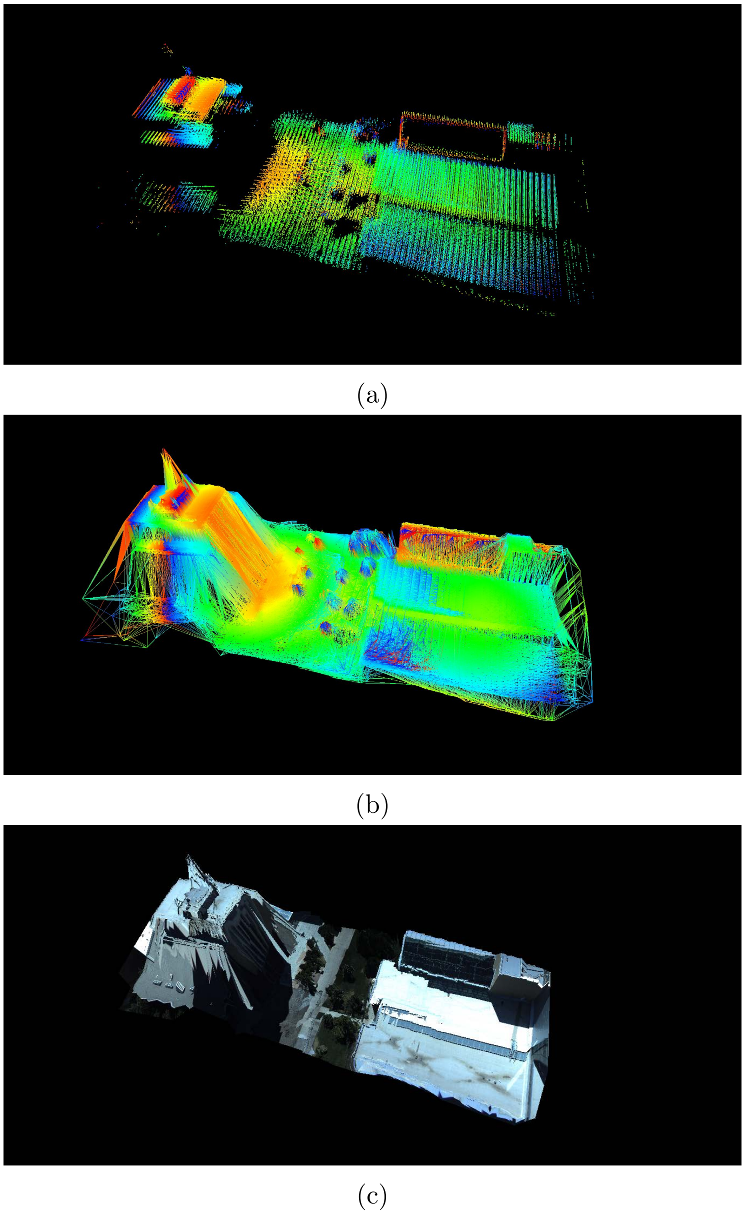


Fig. 2.4: EO image (top) captured at the window of rotation with the corresponding scan of LiDAR points (bottom).

While the 3D reconstruction from the texel images can be accurate for some areas, these 3D surfaces can drift over time causing errors. These errors can be seen again in Figure 2.5c with the taller building's shape and near the center of the 3D reconstruction with clippings of the building. To address this, the texel images are corrected using methods of image processing to find corrections and reduce point cloud registration errors through a form of texel image registration [26].



(a)

(b)

(c)

Fig. 2.5: Example collection of texel images with and without 3D surfacing (a) Corresponding LiDAR point cloud of fifty unregistered texel images. (b) Wireframe of the generated triangulated network for the texel images. (c) TIN 3D surfaced texel images using captured EO imagery.

2.2 Texel Image Registration

To register the texel images is to find a coordinate frame of 3D LiDAR points for all available texel images. Achieving this objective involves establishing correspondences between 3D LiDAR points, pinpointing their locations in the imagery, and determining their positions in adjacent texel images. Initially, the approach entails identifying a 2D position in the j^{th} captured image corresponding to each $\boldsymbol{b_i}$ LiDAR point. These 2D positions project the captured 3D LiDAR points onto the normalized image plane of the j^{th} image. The calibration process, as outlined in Khatiwada (2023) [26], encompasses three key steps: camera calibration, LiDAR-to-camera alignment, and estimation of mapping from the found LiDAR-to-camera. The outcome is the mapping of 2D positions of 3D LiDAR points onto the normalized image plane of the captured image.

Next, finding a common projection for the neighboring texel images is found. Given two texel images, j and k, a single 3D LiDAR point $\boldsymbol{b_i}$ can be seen for both images. If $\boldsymbol{b_i}$ was captured in the j^{th} texel image with a projected 2D position (u_{ij}, v_{ij}), then to find the projection of $\boldsymbol{b_i}$ onto the k^{th} image, a method of correlation is needed. This correlation for each point can be found through Normalized Cross Correlation (NCC).

For the purposes of this thesis, NCC for imagery is a statistical technique used to quantify similarities between two images. The correlation method is an extension of the standard cross-correlation with the added feature of normalization for more generalization. NCC provides a correlation coefficient that ranges from [-1 1], where 1 signifies strong similarity, -1 indicates a strong dissimilarity, and 0 represents no correlation. This normalization makes NCC robust to variations in intensity, and contrast in images. In the context of texel images, NCC is employed for feature matching, where it aids in identifying and aligning similar patterns or objects across multiple EO images. Unfortunately, NCC can be costly during processing, so an initial ”guess” of each point is needed to save time during NCC. To initially start this guess, a correspondence between the 2D position in image j and k is found by finding a homography between the two images.

The homography matrix establishes the correspondence between pixel coordinates in both images by solving the equation $\boldsymbol{P_k} = \boldsymbol{H_{jk}} \boldsymbol{P_j}$, where $\boldsymbol{P_j}$ represents the coordinates in the j^{th} image and $\boldsymbol{P_k}$ represents the coordinates in the k^{th} image. The matrix $\boldsymbol{H_{jk}}$ can be obtained by using the 4-point algorithm and Random Sample Consensus (RANSAC), and lastly finding corresponding image features between two images.

To find matching image feature points between two images, the homography process utilizes a method of Scale-Invariant Feature Transforming (SIFT). SIFT works by first detecting key points in an image that are distinctive and stable under different conditions. Then, it computes descriptors for these key points based on the gradient magnitude and orientation of image patches surrounding them. These descriptors capture the local visual characteristics of the image patches surrounding the key points [27]. The resulting homography is a found transformation of corresponding image points between image j and k. A visual of the results can be seen in Fig. 2.6 below, where transformation is found through the colored border and the lines represent the matched feature points.

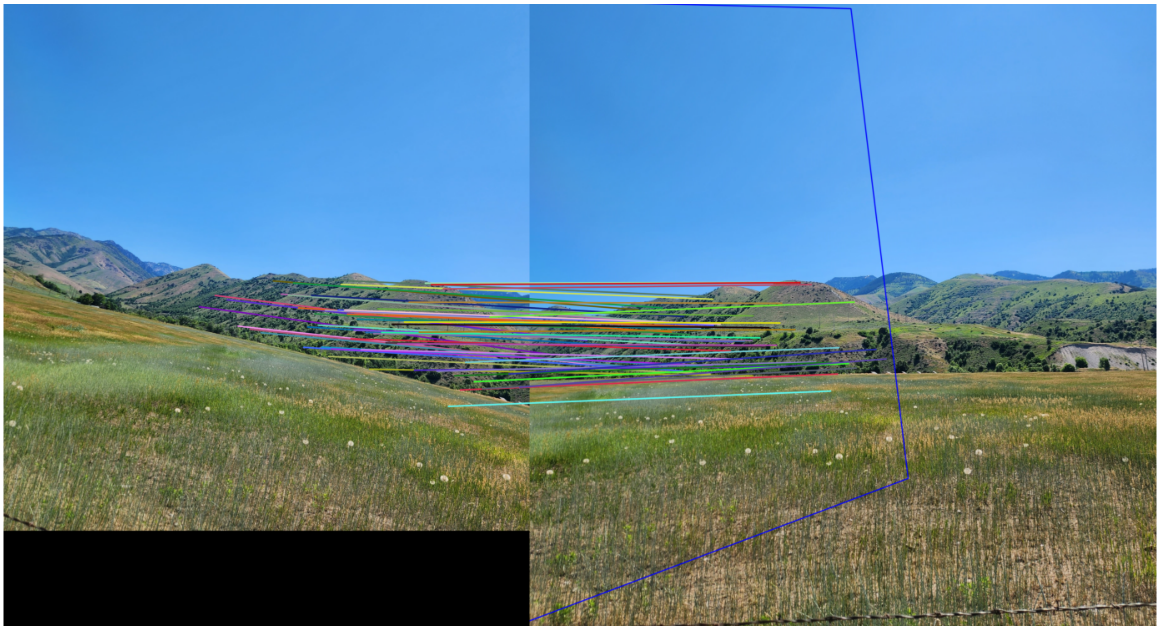


Fig. 2.6: Homography between two image captures in the same scene.

The resulting homography and putative feature points are then inserted into the NCC process. With a found homography $\boldsymbol{H_{jk}}$, the image k is registered, resampled, and warped to fit the j^{th} image space, enabling a smoother outcome with the NCC algorithm. Then a patch of pixels in the j^{th} image at the LiDAR projection of point $\boldsymbol{b_i}$ is set as a reference for the correlation of image k. Due to perspective adjustments between images, the pixel location in k with the highest NCC score is recognized as the same 2D position. The position is then set as the location of the projection of $\boldsymbol{b_i}$ onto texel image k. Then, for each of the $\boldsymbol{b_i}$, the projection is added to a projection matrix. The projection matrix is formulated to consists of an image projection for each 3D point onto all available texel images.

2.2.1 Bundle Adjustments

As the projections of 3D LiDAR points onto a common plane are non-linear approximations, errors and drift may manifest during the registration process. To correct these inaccuracies, bundle adjustments come into play. The bundle adjustment technique, pivotal in computer vision and photogrammetry, refines the estimation of camera poses and 3D structure from a set of images or a fusion of images and LiDAR data [28]. In the texel registration process, employing bundle adjustment becomes crucial to ensure the accuracy and refinement of the final TDSMs. Adopting a bundle adjustment approach optimizes the extrinsic parameters of the EO camera and the 3D LiDAR points for each texel image, striving for the most precise alignment between observed image features and the TDSM.

Sparse Levenberg-Marquardt

In the generation of TDSMs through texel images, a bundle adjustment technique is employed to minimize errors. This TDSM creation process involves the transformation of 3D LiDAR points from world space to normalized image space (a non-linear procedure). To enhance the precision of texel image registration and reduce errors, the LM algorithm is applied in the bundle adjustment method. This approach builds upon the prior TDSM registration work outlined by Khatiwada (2023) [26].

The LM algorithm is an optimization algorithm frequently used in nonlinear least squares parameter estimation. The algorithm is an iterative method that aims to find the best-fitting parameters of a model to a given set of data points, such as the pose and 3D LiDAR points. The LM algorithm combines the advantages of the Gauss-Newton method with gradient descent [29]. This allows for a fast and better convergence for early iterations. By adjusting the damping factor dynamically, it adapts the optimization process based on the local curvature of the cost function, leading to efficient parameter estimation for nonlinear models.

For the purpose of image reconstruction, the sparse LM algorithm is used due to its algorithm specifically designed to handle sparse Jacobian matrices. The LM algorithm is useful for this research because the bundle adjustment optimization problem involves a large number of parameters and a sparse Jacobian matrix. The sparsity of the Jacobian matrix can result from the limited overlap between consecutive scans as well as lack of image patch matches.

For the LM algorithm, the processes uses the set of parameter vectors for $\boldsymbol{S}$, (i.e. the camera and points), the measurement $\boldsymbol{X_{ij}}$, the covariance matrix Σ_X, and the function $f : \boldsymbol{S} \rightarrow \boldsymbol{\hat{X}_{ij}}$ which estimates the measurement matrix. The measurement matrix, denoted as $\boldsymbol{X}_{ij}$, comprises of the projection-range coordinates. These coordinates include $x_{n_{ij}}$ and $y_{n_{ij}}$, representing the 2D normalized image projections of $\boldsymbol{b_i}$ for the j^{th} texel image, and λ_{ij}, indicating the range of $\boldsymbol{b_i}$ for the same texel image. Likewise, the estimated measurement matrix, $\boldsymbol{\hat{X}_{ij}}$, is defined as a function of both the model parameters, $\mathbf{a}$, and the data points, $\mathbf{b}$. These model parameters for each j^{th} texel image are defined as,

$$\boldsymbol{a_j} = [q_{wj}, q_{xj}, q_{yj}, q_{zj}, p_{xj}, p_{yj}, p_{zj}]^T, \tag{2.1}$$

while each i^{th} data point is defined as,

$$\boldsymbol{b_i} = [x_{xi}, x_{yi}, x_{zi}]^T \tag{2.2}$$

which also divides up the parameter vector $\mathbf{S}$ into $\boldsymbol{S} = [a_0, a_1, ..., a_j, b_0, b_1, ..., b_i]^T$. Lastly, we can define the two derivative matrices, $\boldsymbol{A_{ij}} = \frac{\partial \boldsymbol{\hat{X}_{ij}}}{\partial \boldsymbol{a_j}}$ and $\boldsymbol{B_{ij}} = \frac{\partial \boldsymbol{\hat{X}_{ij}}}{\partial \boldsymbol{b_i}}$. The goal is to minimize the error $\boldsymbol{\varepsilon = X - \hat{X}}$ for the measurement position of each LiDAR point and the pose of each texel image. Following a highly simplified version of the partitioned LM algorithm in Hartley and Zisserman [30], the process follows the steps of Algorithm 1.

Essentially, the LM algorithm minimizes the error by computing the Jacobian matrix of the estimated measurement matrix. This Jacobian matrix represents the partial derivatives

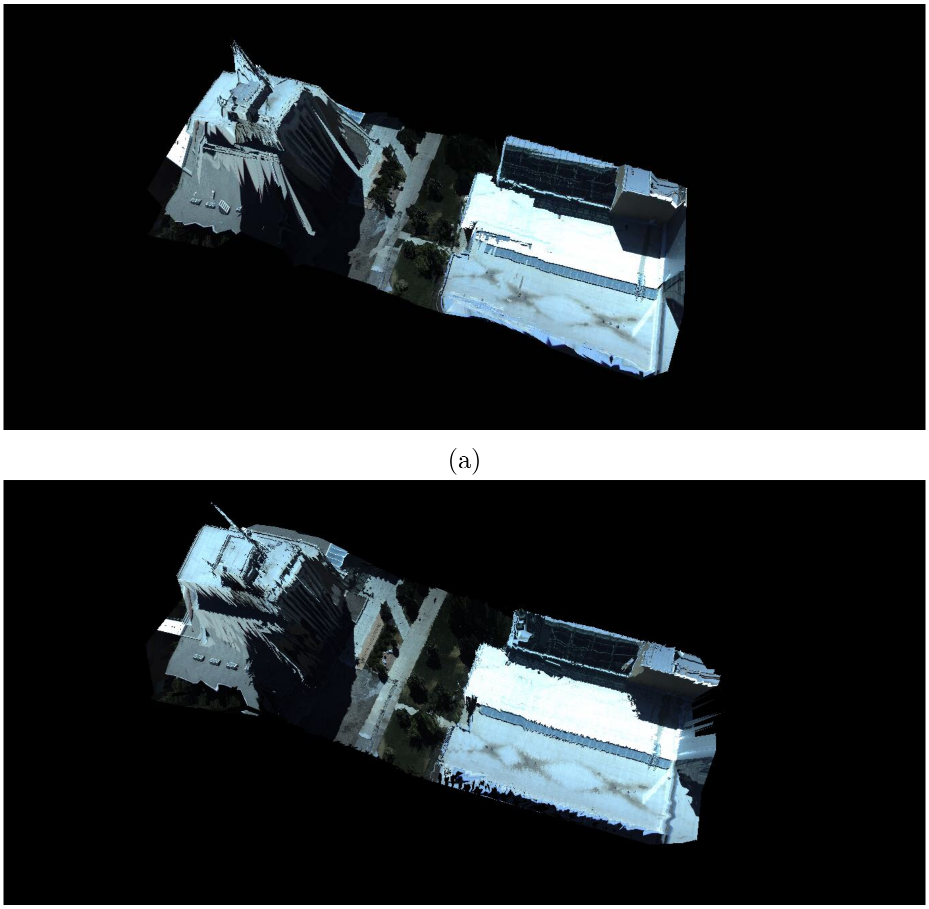
Algorithm 1 Sparse Levenberg-Marquardt

```
1: Initialize the constant $\lambda = 0.001$.
2: Compute the Jacobian matrix, $\boldsymbol{J} =\frac{\delta\hat{\boldsymbol{X}}}{\delta\boldsymbol{S}} = [\frac{\delta\hat{\boldsymbol{X}}}{\delta\boldsymbol{a}},\frac{\delta\hat{\boldsymbol{X}}}{\delta\boldsymbol{b}}]$.
3: Compute expressions $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, $\mathbf{Y}$, $\boldsymbol{\varepsilon_A}$and $\boldsymbol{\varepsilon_B}$

       $\boldsymbol{U_j = A^T\Sigma_X^{-1}A},\ \boldsymbol{V_i = B^T\Sigma_X^{-1}B},$
       $\boldsymbol{W_{ij} = A^T\Sigma_X^{-1}B},\ \boldsymbol{Y_{ij} = W^TV^{*-1}}$
       $\boldsymbol{\varepsilon_A = A^T\Sigma_X^{-1}\varepsilon},\ \boldsymbol{\varepsilon_B = B^T\Sigma_X^{-1}\varepsilon}$

4: Compute $\delta_a$ and $\delta_b$
   → Solve for $\delta_a$ from the equation:

       $(U^* - \sum_i Y_iW_i^T)\delta_a \;=\; \epsilon_a - \sum_i Y_i\epsilon_b$

   → Compute $\delta_b$ using the found $\delta_a$:

       $\delta_b = V_i^{*-1}(\epsilon_b - W_i^T\delta_a)$

5: Formulate the set of equations $\boldsymbol{J\delta = \varepsilon}$ to compute error.

       $\begin{bmatrix}\boldsymbol{U}^* & \boldsymbol{W}\\ \boldsymbol{W} & \boldsymbol{V}^*\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta_a}\\ \boldsymbol{\delta_b}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\varepsilon_a}\\ \boldsymbol{\varepsilon_b}\end{bmatrix}$

6: If the latest error is less than the old error, save the new values of the parameters for
   the next iteration, starting at step 2 and dividing $\lambda$ by 10.
7: If this error difference is the best after an predetermined number of iterations and/or
   threshold, terminate. → The final parameters are saved to minimize the resulting
   squared Mahalnobis distance $||\boldsymbol{\varepsilon}||^2_{\boldsymbol{\Sigma_X}} = \sum_i\sum_j \boldsymbol{\varepsilon_{ij}^T\Sigma_{X_{ij}}^{-1}\varepsilon_{ij}}$.
```

of the predicted measurements with respect to its model parameters, **a** and **b**. Expressions can then be computed to represent a set of normal equations to find a set of vectors, $\boldsymbol{\delta_a}$ and $\boldsymbol{\delta_b}$, to incrementally shift the original estimate. If these incremental vectors improve the overall error, the new values of the parameters are accepted and process starts again.

Final TDSM Results

Once multiple images are matched to their corresponding point cloud data and transformed to a single plane, the set of data is then textured by applying a TIN surface to the 3D point clouds and then overlaying matched imagery. The surfacing creates a single 3D reconstruction for those scans. The result of the design is a TDSM for the captured image's corresponding terrain. Following these steps, the final TDSM is registered as seen in Figure 2.7b, which can be compared to the original 50 texel image model in Figure 2.7a. Most notably, it can be seen that the shape of the taller building is more precise, the sidewalks are straighter, and there is minimal clipping of data. Consequently, due to the resolution of the LiDAR unit and the top-down nature of the image captures, vertical walls or edges of the building may not be completely straight, but are improved through TDSM registration.



(a)

(b)

Fig. 2.7: Example TDSM of an area of Utah State University's campus created from texel images captured from the texel camera mounted on a sUAV during a flight. (a) TDSM from 50 texel images. (b) Registered TDSM compiled into a single scan.

2.3 Attitude Conventions

Attitude plays a critical role during navigation, whether it's in the context of aviation, marine navigation, or any other form of transportation. Attitude refers to the aircraft's orientation relative to the Earth's horizon and includes roll, pitch, and yaw. Maintaining the proper attitude enables precise control, allowing efficient and effective registration with the texel camera.

2.3.1 Quaternions

The mathematical concept of quaternions extends the idea of complex numbers for four dimensions and were first derived in 1819 by Carl Friedrich Gauss and later introduced by Sir William Rowan Hamilton in 1843 [31,32]. Quaternions represent rotations in 3D space, and are defined in the form: $q = w + ix + jy + kz$. The values *i, j, k* represent imaginary units, and satisfy: $i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$, while *w, x, y, z* represent real numbers. For the purpose of this research, quaternions give a representation of the systems attitude.

A benefit to quaternions is that they represent an orthonormal rotation when the quaternion is normalized. By normalizing the quaternion, this ensures that the axis of rotation (the imaginary part: *i, j, k*) is orthogonal to the angle of rotation (the real part: w), while also representing a rotation with a magnitude of 1. This normalization ensures orthonormality of the quaternion.

For the purpose of GPS-denied navigational methods, the quaternion must be normalized accordingly. To normalize a quaternion, each component of the quaternion must be divided by the quaternions magnitude, or length. This is done by calculating the Euclidean norm, or L2 norm, of the quaternion vector as defined below,

$$\boldsymbol{q_k} = \frac{\boldsymbol{q}}{||q||} = \frac{1}{\sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}} \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T \tag{2.3}$$

where q_k represents the normalized quaternion at the measured discrete index k. The normalization of the quaternion ensures the vector is unit length. A unit quaternion shall guarantee a true representation of a 360 degree rotation, without any additional scaling.

Quaternions also offer numerous other operations, such as addition, inversion, multiplication, and conjugation. Of these operations, the quaternion multiplication shall be defined as:

$$\boldsymbol{q_1} \otimes \boldsymbol{q_2} = \begin{bmatrix} (q_{w1}q_{w2} - q_{x1}q_{x2} - q_{y1}q_{y2} - q_{z1}q_{z2}) \\ (q_{w1}q_{x2} + q_{x1}q_{w2} + q_{y1}q_{z2} - q_{z1}q_{y2}) \\ (q_{w1}q_{y2} - q_{x1}q_{z2} + q_{y1}q_{w2} + q_{z1}q_{x2}) \\ (q_{w1}q_{z2} + q_{x1}q_{y2} - q_{y1}q_{x2} + q_{z1}q_{w2}) \end{bmatrix} \tag{2.4}$$

2.3.2 Gimbal Lock

Another advantage that make quaternions desirable is how the system avoids gimbal lock, which occurs in other rotation representations such as Euler angles [33]. A gimbal typically represents perpendicular rings or frames that pivot and rotate around the axes of the system. For a given system, classically it will have six degrees of freedom. The degrees of freedom represent the 3D translation and 3D rotation of the system.

When two of the axes align in a system, the system loses a degree of freedom, causing the device to be “locked“. This locking can lead to unexpected behavior and difficulties in rotation representation, hence the desire to represent rotation of a system in quaternions, which avoid this problem. To visualize, Figure 2.8 depicts a system with gimbal lock if the attitude representation were to be in roll (ϕ), pitch (θ), and yaw (ψ) for navigational purposes. The figure presents a aerial system where the yaw axis rotates 90 degrees to align with the pitch axis. This results in gimbal lock, causing a loss of one degree of freedom and potentially leading to difficulties in controlling the orientation of the system. Generally, for a aerial system, gimbal lock must be avoided as precise orientation control is required.

By representing orientation of the sUAV using quaternions, smooth calculations can be made when predicting the sUAV’s position during a flight. Quaternion error representation can also be useful, as quaternion properties can handle orientation deviations and error correction with ease. For the purpose of this chapter and research, the quaternions were used for the ESEKF estimator introduced in this chapter.

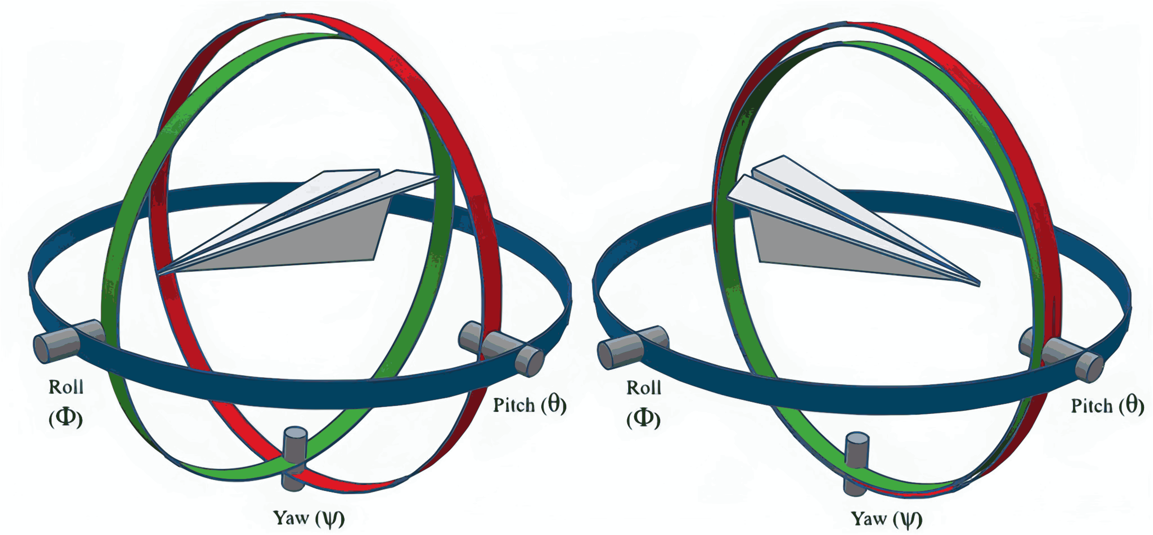


Fig. 2.8: Visual depiction of gimbal lock in an aerial system. Left represents the system with all degrees of freedom. Right represents the system losing one degree of freedom and in gimbal lock.

2.3.3 Matrix Representation of Camera Orientation and Location

The orientation and location of each scan's camera pose can be represented by seven parameters: the normalized quaternion rotation vector, and the 3D location vector relative to the sUAV's starting position. With these seven parameters, the orientation and location of the system can be represented by a vector $\boldsymbol{S} = [q_w,\ q_x,\ q_y,\ q_z,\ p_x,\ p_y,\ p_z]^T$. This vector can also be represented as,

$$\boldsymbol{S} = \left[\begin{array}{c|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right] = \left[\begin{array}{ccc|c} 1 - \frac{2(q_y^2+q_z^2)}{q_w^2+q_x^2+q_y^2+q_z^2} & \frac{2(q_x q_y - q_w q_z)}{q_w^2+q_x^2+q_y^2+q_z^2} & \frac{2(q_x q_z + q_w q_y)}{q_w^2+q_x^2+q_y^2+q_z^2} & p_x \\ \frac{2(q_x q_y + q_w q_z)}{q_w^2+q_x^2+q_y^2+q_z^2} & 1 - \frac{2(q_x^2-q_z^2)}{q_w^2+q_x^2+q_y^2+q_z^2} & \frac{2(q_y q_z - q_w q_x)}{q_w^2+q_x^2+q_y^2+q_z^2} & p_y \\ \frac{2(q_x q_z - q_w q_y)}{q_w^2+q_x^2+q_y^2+q_z^2} & \frac{2(q_y q_z + q_w q_x)}{q_w^2+q_x^2+q_y^2+q_z^2} & 1 - \frac{2(q_x^2-q_y^2)}{q_w^2+q_x^2+q_y^2+q_z^2} & p_z \end{array}\right]$$

which is a 3x4 matrix containing the rotation of the system, $\mathbf{R}$, and the location, $\mathbf{t}$, at the current texel image scan. The defined rotation $\mathbf{R}$ and location $\mathbf{t}$ are relative to the world coordinate system's origin matrix. This origin matrix is defined as,

$$\boldsymbol{S_o} = \left[\begin{array}{c:c} \boldsymbol{R_o} & \boldsymbol{t_o} \end{array}\right] = \left[\begin{array}{ccc:c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

The defined matrices are then used to map 3D homogeneous coordinate points from one coordinate system into another coordinate system. These matrices are also used as reference for origin points and previous system characteristics during GPS-denied navigation.

2.4 Error State Extended Kalman Filter

To achieve sub-accurate navigation in a GPS-denied scenario, nearly all articles reviewed use a variation of the Kalman filter. For the purpose of this thesis, the ESEKF (as introduced in Chapter 1) is used because the orientation error-state is minimal, it is void of gimbal lock issues, and the error-state is always small [34–36]. Renowned for its application in navigation, the ESEKF enhances filter efficiency by concentrating on error states while addressing non-linearities and uncertainties in system dynamics. Unlike directly estimating the entire state vector, the ESEKF focuses on estimating errors within the state vector.

For GPS-Denied navigation, the ESEKF specifically estimates pose errors derived from IMU measurements. Additionally, external measurement updates can refine these error state estimates, guiding the pose estimation toward the assumed correct direction. In Nazaruddin et al. the ESEKF is used with the measurement update from a memory network to return the next displacement of their autonomous vehicle, achieving a reduced error in a GPS-denied environment [36]. In Lu et al. similar work is done that is presented in this proposal, using position estimation through the ESEKF, as well as the addition of a camera for visual odometry corrections [9].

2.4.1 System and State Kinematics

The vehicle state ($\boldsymbol{x_k}$) vector at each time step contains the position, velocity, and orientation in quaternions; while the control input vector ($\boldsymbol{u_k}$) contains the IMU's measured acceleration, angular velocity and rotation in quaternions as seen below:

$$\boldsymbol{x_k} = \begin{bmatrix} p_k \\ v_k \\ q_k \end{bmatrix}, \qquad \boldsymbol{u_k} = \begin{bmatrix} \bar{a}_k \\ \bar{\theta}_k \\ \bar{q}_k \end{bmatrix}.$$

where $\bar{u}$ represents the incoming measured data from the INS. These states can be broken into their respective vectors at the current index of k, where the position is defined as $\boldsymbol{p_k} = [p_x\ p_y\ p_z]^T$, velocity as $\boldsymbol{v_k} = [v_x\ v_y\ v_z]^T$, and attitude in quaternions as $\boldsymbol{q_k} = [q_w\ q_x\ q_y\ q_z]^T$.

2.4.2 Prediction Step and Uncertainty Propagation

Once a control input vector is measured from the IMU, the vehicle states need to be predicted to give a best estimate for the sUAV's positiong. To do so, the ESEKF follows a set of state prediction equations to update the estimated state of the system. The prediction step is derived off of the previous state estimates and the system dynamics of the sUAV. During the prediction state of the ESEKF, the system follows,

$$\boldsymbol{\hat{p}_k^-} = \boldsymbol{\hat{p}_{k-1}^+} + \Delta t \boldsymbol{\hat{v}_{k-1}} + \frac{\Delta t^2}{2}(\boldsymbol{R_{k-1}^+ \bar{a}_k} + \boldsymbol{g}), \tag{2.5}$$

$$\boldsymbol{\hat{v}_k^-} = \boldsymbol{\hat{v}_{k-1}^+} + \Delta t(\boldsymbol{R_{k-1}^+ \bar{a}_k} + \boldsymbol{g}), \tag{2.6}$$

and

$$\boldsymbol{\hat{q}_k^-} = \boldsymbol{\hat{q}_{k-1}^+} \otimes \boldsymbol{\bar{q}_k} \tag{2.7}$$

where the vector g is the gravity vector, g = $[0\ 0\ -9.81]^T$ [34,37,38]. These equations are used to process the next state of the vehicle by integrating the system's IMU measurements. The notation $^-$ and $^+$ shall denote the *a priori* and *posteriori* vector state estimation at their respective time index. The equations (2.5)–(2.7) represent the discrete motion model and dynamics of the system. $\boldsymbol{R_k^+}$ then represents the rotational matrix of the given normalized quaternion $\boldsymbol{q_k^+}$ and is calculated using the representation defined:

$$\boldsymbol{R_k^+} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_xq_y - q_wq_z) & 2(q_xq_z + q_wq_y) \\ 2(q_xq_y + q_wq_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_yq_z - q_wq_x) \\ 2(q_xq_z - q_wq_y) & 2(q_yq_z + q_wq_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{2.8}$$

Lastly, we note the operator $\otimes$ as the quaternion product mentioned in 2.4. These functions represent the discretized representations according to the time intervals Δt.

Next is the predicted covariance estimate of the system, or the uncertainty matrix. Using linear error state kinematic equations for position, velocity, and rotation, the addition of the error state is included in the prediction of the covariance update as follows:

$$\boldsymbol{\hat{P}_k^-} = \boldsymbol{F P_{k-1}^+ F}^T + \boldsymbol{LQL}^T, \tag{2.9}$$

where $\boldsymbol{F}$ represents the transition matrix obtained as a Jacobian form of the error state; $\boldsymbol{L}$ represents the perturbation Jacobian matrix; $\boldsymbol{Q}$ represents the measurements covariance matrix; and $\boldsymbol{P_{k-1}}$ represents the *posteriori* state covariance matrix of the previous index. These equations can then function as the position estimator without texel image corrections.

2.4.3 Measurement Update

During an aerial data capture, the texel camera takes scans of data that is then processed into TDSMs. At the same time, the IMU is running and recording measurements of velocity and rotation. During the position estimation, if the recorded time matches the time of a scan captured from the texel camera, a measurement update occurs in the ES-EKF. This results in the measurement vector, $\boldsymbol{y_k}$, which represents the incoming data that updates the desired estimate and is defined as,

$$\boldsymbol{y_k} = [p_{xj},\ p_{yj},\ p_{zj},\ 0,\ 0,\ 0,\ \theta_{xj},\ \theta_{yj},\ \theta_{zj}]^T.$$

where θ represents the Euler angle of rotation found by conversion of the LM corrected quaternion and p represents the 3D position of the j^{th} texel image after extracting the resulting $\boldsymbol{a_j}$ from the LM algorithm 1. Depending on what measured data is updated in an ESEKF, any of the states in $\boldsymbol{y_k}$ can be updated. The estimated state vector $\boldsymbol{\hat{x}_k^-}$ is then converted to follow the dimensions and states of $\boldsymbol{y_k}$ by conversion of quaternion to Euler angle as defined by

$$\boldsymbol{\hat{x}_k^-} = [\hat{p}_k^-,\ \hat{v}_k^-,\ \theta\{\hat{q}_k^-\}]^T.$$

Given the estimated position from the ESEKF and the corrected position from the texel registration (discussed in Section 4.1), the estimated position is corrected using the measurement update process below:

$$\boldsymbol{K_k} = \boldsymbol{P_k^-}\boldsymbol{H_k}^T(\boldsymbol{H_k}\boldsymbol{P_k^-}\boldsymbol{H_k}^T + \boldsymbol{R})^{-1} \tag{2.10}$$

where $\boldsymbol{R}$ is the measurement noise covariance matrix, $\boldsymbol{K_k}$ is the gain matrix, and $\boldsymbol{H}$ is the observation matrix that indicates what measurements are to be updated from the texel registration process. For the purpose of our texel camera, the observation matrix is defined as $\boldsymbol{H} = Diag([1,1,1,0,0,0,1,1,1])$, where the one's indicate the measured states of the system that are being updated during this step. The updated states are then measured to be position and rotation during texel registration.

The matrix $\boldsymbol{R}$ is defined as a diagonal matrix with variances for the position, velocity, and orientation. The updated error state vector is then found and the estimated position, velocity, and rotation are corrected using

$$\boldsymbol{\delta\hat{x}_k^+} = \boldsymbol{K_k}(\boldsymbol{y_k} - \boldsymbol{\hat{x}_k^-}) = [\boldsymbol{\delta\hat{p}_k^+}\ \ \boldsymbol{\delta\hat{v}_k^+}\ \ \boldsymbol{\delta\hat{\theta}_k^+}]^T, \tag{2.11}$$

$$\boldsymbol{\hat{p}_k^+} = \boldsymbol{\hat{p}_k^-} + \boldsymbol{\delta\hat{p}_k^+}, \tag{2.12}$$

$$\boldsymbol{\hat{v}_k^+} = \boldsymbol{\hat{v}_k^-} + \boldsymbol{\delta\hat{v}_k^+}, \tag{2.13}$$

and

$$\boldsymbol{\hat{q}_k^+} = \boldsymbol{\hat{q}_k^-} \otimes \boldsymbol{q}\{\boldsymbol{\delta\hat{\theta}_k}\}, \tag{2.14}$$

Lastly, the covariance matrix is updated for the next iteration with

$$\boldsymbol{\hat{P}_k^+} = (\boldsymbol{I} - \boldsymbol{K_k H_k})\boldsymbol{P_{k-1}^-}(\boldsymbol{I} - \boldsymbol{K_k H_k})^T + \boldsymbol{K_k R K_k}^T. \tag{2.15}$$

This then adds the additional gain updates that were found during the measurement update process to the covariance matrix.

CHAPTER 3
INS-TO-CAMERA PRINCIPAL POINT ALIGNMENT

3.1 Introduction

The camera principal point is a specific point within the camera's image plane, where the optical axis intersects and for the purpose of this chapter, can be seen as the central position of the camera. This principal point is conventionally denoted as $(c_x,\ c_y)$ and is expressed in pixel coordinates on the camera's image plane. The c_x coordinate corresponds to the horizontal position, aligning with the image's width, while the c_y coordinate denotes the vertical position, corresponding to the image's height.

The principle point along with the camera's focal lengths in pixels, (f_x, f_y), can be combined to create the camera calibration matrix, which can be represented as,

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.1}$$

It shall be noted that the texel camera used in this research has undergone prior calibration, resulting in the availability of the camera calibration matrix. To enhance the accuracy of registration and position estimation, one effective approach involves aligning the INS sensor with the EO camera's principal point.

While the INS sensor is affixed to the same mounting bracket as the EO camera and LiDAR unit, an offset is present between the two sensors. The purpose of this alignment is to accurately represent the potential displacement in rotation and translation between the INS and EO camera. Aligning the INS with the principal point of the texel camera is a critical step in enhancing the accuracy and reliability of aerial position estimation and texel image registration.

As presented in this thesis, there is a heavy reliance on navigational data from the INS and camera. The significance of this alignment becomes particularly evident when considering the accuracy of the orientation of the texel camera. Even the slightest deviation in the orientation of the texel camera, especially when positioned at a significant altitude, such as 60 meters in the air, can possibly lead to a large impact on the accuracy of texel image registration and the reliability of the estimated position. While the orientation of the EO camera and the INS may seem parallel in theory, sensor mounting errors might be present. To combat this, a calibration in the INS and EO camera's principal point was performed.

3.2 INS Sensor to Camera in Context of Motion Capture

Recognizing the need for INS-to-camera alignment, an approach was conducted using motion capture technology. This technology allows for the identification of real-world features, their 3D positions, and orientation to a fixed origin. By leveraging this 3D positional data along with the information gathered from the INS and the images acquired by the camera, we can effectively reduce the misalignment between the INS poase and the camera's pose.

3.2.1 Motion Capture Background & Methodology

Motion capture (MoCap), is a technology used to record the movements of objects or living beings, typically humans, and translate those movements into digital data. MoCap technology is widely used in various fields, including entertainment, medicine, computer vision, and robotics, for a range of purposes [39–41]. Commonly, for optical based motion capture, there are two types of systems: marker and marker-less motion capture [42]. The approach used in this chapter shall use a marker based motion capture system.

To do so, a motion capture studio, hosted by Dr. Greg Droge at Utah State University, was used in achieving alignment precision. This studio, characterized by an array of strategically positioned cameras enveloping a workspace, locates and tracks placed markers featuring reflective material. Because this MoCap studio operates by tracking placed markers, the 3D positions of the INS and ten placed markers can be located throughout multiple image captures. Furthermore, it captures the orientations of the INS relative to a predefined origin. Figure 3.1 presents the setup where the mounted EO camera/INS is located using the three attached markers, while the table holds the ten placed markers to be located and captured through imagery.

To maximize the efficacy of alignment efforts, the ten markers were consistently featured within the camera's field of view. Multiple images of the markers were then captured and the pixel positions of the markers in each image were recorded. Presented in Figure 3.2, the resulting images show varying orientations and positions, keeping each of the ten highlighted markers in the image. The variation in imagery ensures greater accuracy in INS-to-camera alignment.

Utilizing the 3D positions of the INS and markers, as well as marker locations in the image, and orientation of the INS, an approach can be found to describe the rotation and translation offset of the INS to the camera principal point.

3.2.2 Perspective-n-Point Alignment Estimation

Perspective-n-Point (PnP) is a computer vision and robotics problem used to estimate the pose of a camera in 3D space relative to a set of known 3D points and their corresponding 2D image projections [43, 44]. PnP algorithms are typically used when you have a set of 2D-3D correspondences between points in an image and their 3D coordinates in world space. This approach can then be represented using the collected data from the MoCap studio.

By taking the 3D MoCap world-space positions of the markers and representing them as the matrix $\boldsymbol{X_w}$, and the normalized image projected positions as the vector $[u, v, 1]^T$, these can be added to the PnP equation below:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{K \Pi {}^{c}T_{w}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.2}$$

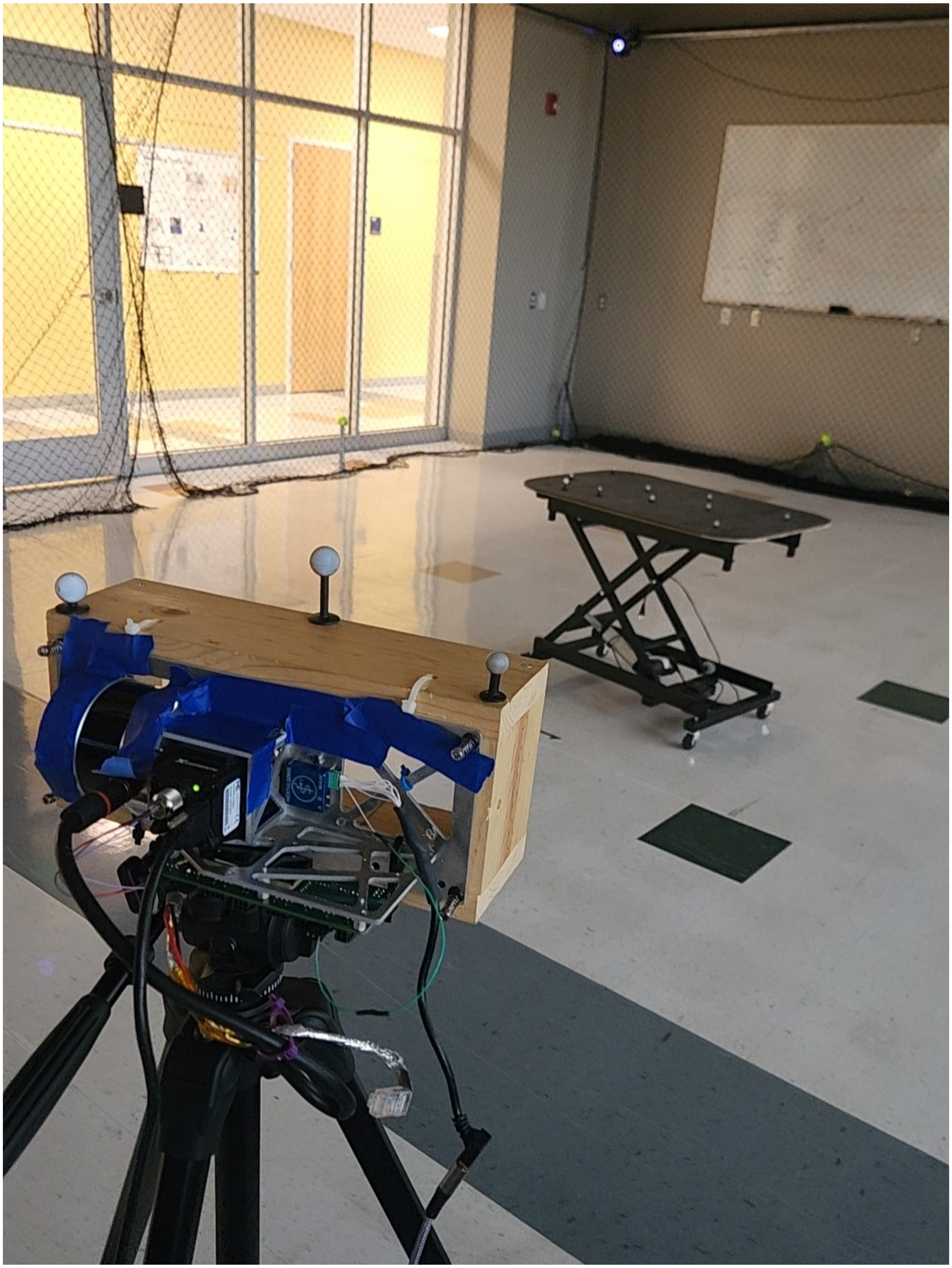


Fig. 3.1: Motion capture setup for alignment of the INS to camera principal point.

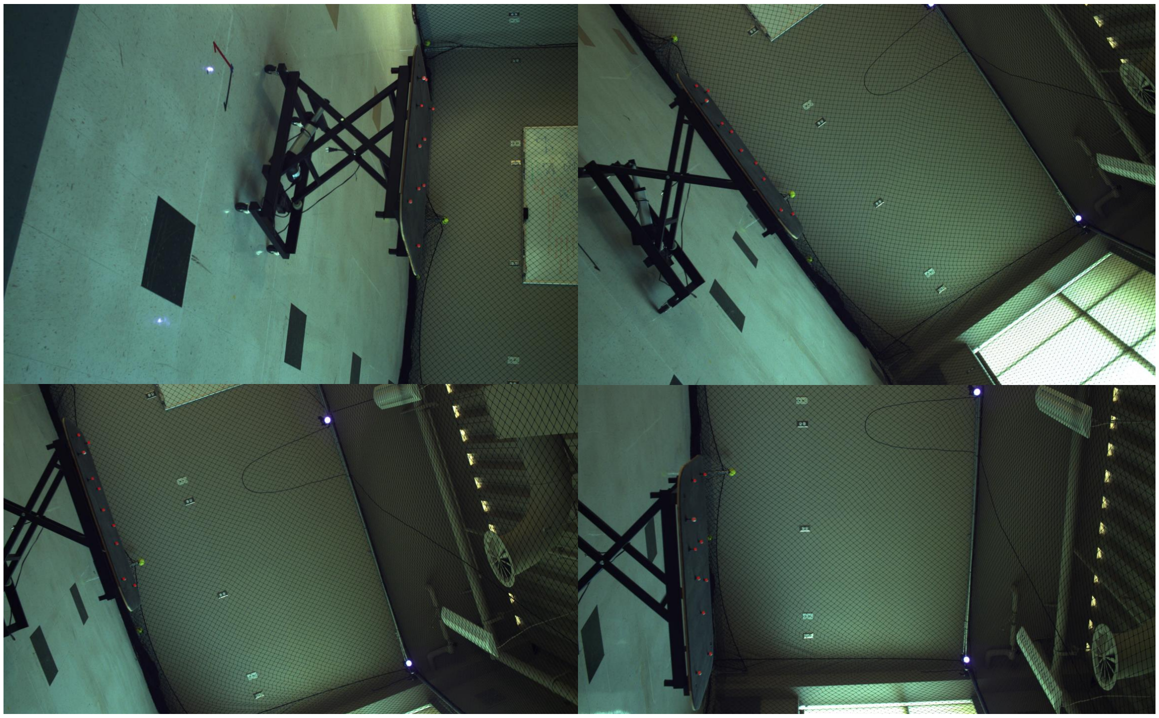


Fig. 3.2: Various MoCap image captures with highlighted (red) pixel location for each marker in their images.

where $\mathbf{K}$ is the calibrated camera intrinsic matrix, $\mathbf{\Pi^c}$ is the perspective projection model, and $\boldsymbol{T_w}$ is the pose matrix containing the desired rotation and translation to minimize 3D-2D re-projection error. Equation 3.2 then can be expanded out to be defined as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.3}$$

Numerous code libraries provide ready-made functions for tackling the PnP problem, often featuring diverse variations to cater to users' specific requirements. Among these libraries is OpenCV's solvePnPRefineLM function, which accepts the input of initial estimates for rotation and translation from INS to the principal point. This process involves the utilization of the Levenberg-Marquart (LM) algorithm, as detailed in Section 2.2.1, in an iterative manner to minimize errors within the resulting pose offset matrix, all initiated from the initial guess.

For the process of this thesis, the initial guess of rotation and translation was found using caliper measurements of an assumed identity matrix rotation and a translation of $[0.041, 0, 0.01]$ meters in the $[x, y, z]$ direction.

3.2.3 Further Motion Capture Minimization

While the PnP solution obtained by minimizing the PnP problem's solution closely approximated the initial guess derived from caliper tool-based measurements, the level of accuracy remains uncertain. As an additional step in addressing the offset between the INS and camera, an alignment technique was adapted from work by Dr. Scott Budge [45]. Originally designed for aligning LiDAR with the EO camera's principal point using positional data from LiDAR point clouds and matching them with imagery, this section employs a similar approach. This alignment utilizes MoCap data, INS orientation, and a minimization process to mitigate errors.

Through the measurement of positional data from the MoCap system, the following data will be available: $\boldsymbol{P^{W}_{f_i}}$, the world space feature position for the i^{th} marker, $\boldsymbol{P^{W}_{r_j}}$, the world space position of the INS at the j^{th} captured image, and $\boldsymbol{\hat{R}^{W}_{b_j}}$, the rotation from the body to the world-frame of the INS at the j^{th} image. This data will be set as known parameters to build a system for minimization of error. Then, for each $\boldsymbol{P^{W}_{r_j}}$, an estimated vector $\boldsymbol{\hat{l}_{ij}}$ can be extended from the principal point, through the projected image space, to then intersect with other rays in images at $\boldsymbol{P^{W}_{f_i}}$.

$\boldsymbol{\hat{l}_{ij}}$ can be estimated by finding the distance in meters, l_d, between the solved PnP position of the principal point and each marker position. This distance vector can then be applied to the projected image positions to create the desired vector:

$$\boldsymbol{l_{ij}} = \frac{l_{d_{ij}}}{\sqrt{u_j + v_j + 1}} * \begin{bmatrix} u_j \\ v_j \\ 1 \end{bmatrix}$$

Lastly, a subtraction of the translation offset from the INS-to-camera and a rotation from the INS body to the camera is used to correct for the INS-to-camera offsets.

By then combining all of these parameters together, an equation can be produced to estimate the position of each marker in world space as seen in (3.4),

$$\boldsymbol{\hat{P}_{f_i}^{W}} = \boldsymbol{P_{r_j}^{W}} + \boldsymbol{\hat{R}_{b_j}^{W}} \boldsymbol{R_b^C} (\boldsymbol{l_{ij}} - \boldsymbol{d_{cam}}). \tag{3.4}$$

Because the process requires a minimization of error to find the closest solution from the marker position estimate, an enumeration of errors is required to find the two INS-to-camera parameters: $\boldsymbol{R_b^C}$, the rotation from INS body to camera, and $\boldsymbol{d_{cam}}$, the translation from the INS to the principal point in meters. The minimization of error is then set up to be:

$$\min_{\boldsymbol{R_b^C}, \boldsymbol{d_{cam}}} \sum_i \sum_j \Sigma_{ij}^T \Sigma_{ij} \tag{3.5}$$

where

$$\begin{aligned} \Sigma_{ij} &= \boldsymbol{P_{f_i}^{W}} - \boldsymbol{\hat{P}_{f_i}^{W}} \\ &= \boldsymbol{P_{f_i}^{W}} - (\boldsymbol{P_{r_j}^{W}} + \boldsymbol{\hat{R}_{b_j}^{W}} \boldsymbol{R_b^C} (\boldsymbol{l_{ij}} - \boldsymbol{d_{cam}})) \end{aligned} \tag{3.6}$$

Resulting Alignment

Using MATLAB's Nelder-Mead simplex search method from Lagarias et al. [46], a minimization is conducted by expanding and contracting small iterations of the desired estimated variables, then comparing the objective function's results for a minimized result. For the purpose of INS-to-camera offset minimization, the variables used will be $\boldsymbol{R_b^C}$ and $\boldsymbol{d_{cam}}$, and the objective function will be (3.4). By iteratively stepping through the objective function, the PnP variable solution was corrected to have a resulting alignment of,

$$\boldsymbol{d_{cam}} = \begin{bmatrix} 0.041324652681350 & 0.000423532551884 & 0.007562668310419 \end{bmatrix}$$

$$\boldsymbol{R_b^C} = \begin{bmatrix} -0.072210607891832 & 0.987409656189228 & 0.140740182507212 \\ 0.994297071943732 & 0.082369485001079 & -0.067739210689071 \\ -0.078479047089102 & 0.135046061789777 & -0.987726480541581 \end{bmatrix}$$

This offset can then be applied to the collected INS data, and LiDAR data prior to producing the texel images at each swath of the LiDAR unit's rotation. The result is an improved accuracy in the texel image registration into TDSMs and GPS-denied navigation.

CHAPTER 4

GPS-DENIED NAVIGATION USING TEXEL IMAGE REGISTRATION

In this thesis, GPS-Denied navigation is presented through the integration of the ESEKF with texel image registration, as discussed in Chapter 2. This involves incorporating the ESEKF's estimated pose into the common plane projection and bundle adjustment stages of texel image registration. Consequentially, the filtered INS data is fused with the texel image registration process, enhancing the overall navigation system and removing GPS dependency.

4.1 Position Correction through Texel Registration

During aerial flight, the texel camera simultaneously captures data for both texel images and the ESEKF navigation system. In the texel image registration, an accurate camera pose is crucial for precise 3D-to-2D projection of LiDAR point clouds throughout a flightline dataset. To minimize errors in this projection, a bundle adjustment algorithm is applied as mentioned in Section 2.2.1.

In traditional systems, the camera pose is typically determined using a GPS receiver linked to the INS. However, when faced with a GPS-Denied environment, the pose becomes an estimation task for the introduced ESEKF. These updates, sourced from the bundle adjustment stage of texel image registration, contribute to the overall accuracy of the navigation system.

Initially, the IMU sources data to the ESEKF's state vector where the estimated pose is found for each sample. Once the timestamp of the IMU matches the timestamp of the captured texel image, the estimated pose is saved for that single texel image. This pose is then placed into the bundle adjustment (i.e. LM algorithm) stage during registration.

As discussed, the LM algorithm is employed to minimize errors in the 3D-to-2D projection and camera pose. This minimization process involves iteratively adjusting the initial pose estimate to identify the optimal fit solution. Once the minimized error falls below a predefined threshold, the resulting measurement matrix is saved, and the refined pose is extracted. This corrected pose, obtained through the LM algorithm, is subsequently integrated into the measurement update step of the ESEKF. This integration serves to adjust the estimated position, bringing it closer to the true location. The estimated pose is then continually found as the data is streamed to the texel camera system.

With all parameters and steps combined, the resulting GPS-denied system can be completed by following the block diagram as seen in Figure 4.1 below.

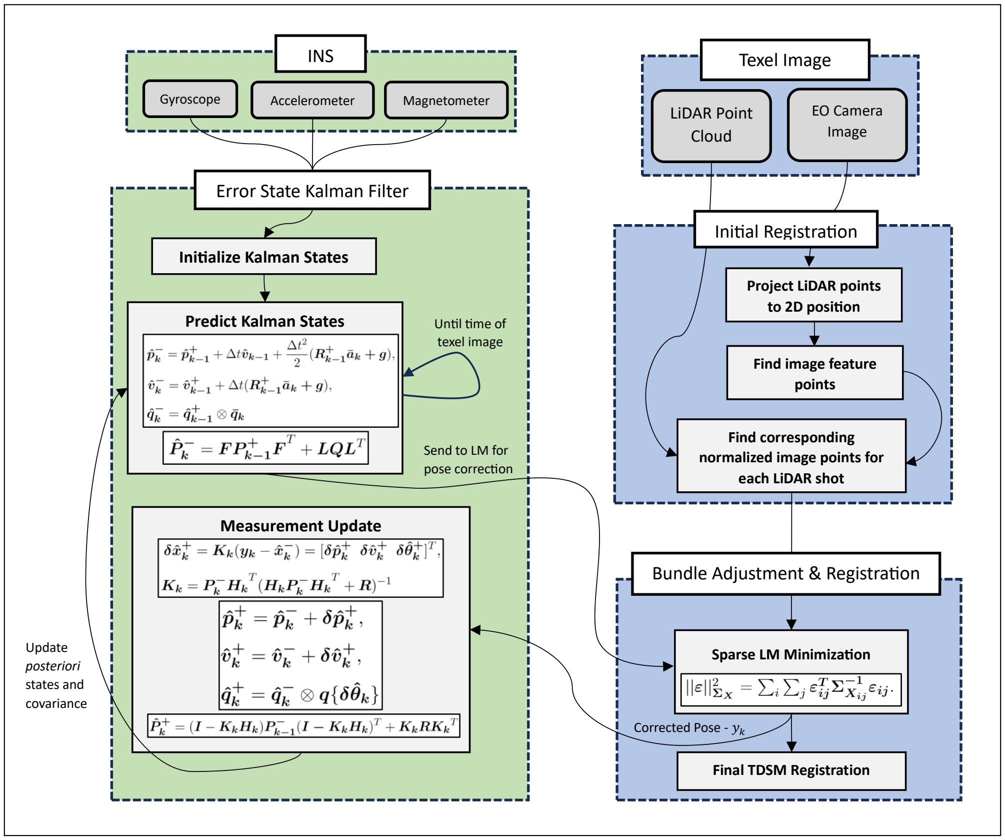

INS
Gyroscope
Accelerometer
Magnetometer
Error State Kalman Filter
Initialize Kalman States
Predict Kalman States
$\hat{p}_k^- = \hat{p}_{k-1}^+ + \Delta t \hat{v}_{k-1} + \frac{\Delta t^2}{2}(R_{k-1}^+ \bar{a}_k + g),$
$\hat{v}_k^- = \hat{v}_{k-1}^+ + \Delta t(R_{k-1}^+ \bar{a}_k + g),$
$\hat{q}_k^- = \hat{q}_{k-1}^+ \otimes \bar{q}_k$
$\hat{P}_k^- = F P_{k-1}^+ F^T + L Q L^T$
Until time of texel image
Send to LM for pose correction
Measurement Update
$\delta\hat{x}_k^+ = K_k(y_k - \hat{x}_k^-) = [\delta\hat{p}_k^+ \;\; \delta\hat{v}_k^+ \;\; \delta\hat{\theta}_k^+]^T,$
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$
$\hat{p}_k^+ = \hat{p}_k^- + \delta\hat{p}_k^+,$
$\hat{v}_k^+ = \hat{v}_k^- + \delta\hat{v}_k^+,$
$\hat{q}_k^+ = \hat{q}_k^- \otimes q\{\delta\hat{\theta}_k\}$
$\hat{P}_k^+ = (I - K_k H_k) P_{k-1}^- (I - K_k H_k)^T + K_k R K_k^T$
Update *posteriori* states and covariance
Texel Image
LiDAR Point Cloud
EO Camera Image
Initial Registration
Project LiDAR points to 2D position
Find image feature points
Find corresponding normalized image points for each LiDAR shot
Bundle Adjustment & Registration
Sparse LM Minimization
$||\varepsilon||_{\Sigma_X}^2 = \sum_i \sum_j \varepsilon_{ij}^T \Sigma_{X_{ij}}^{-1} \varepsilon_{ij}.$
Corrected Pose - y_k
Final TDSM Registration

Fig. 4.1: GPS-Denied block diagram algorithm based on the INS and texel imagery

CHAPTER 5

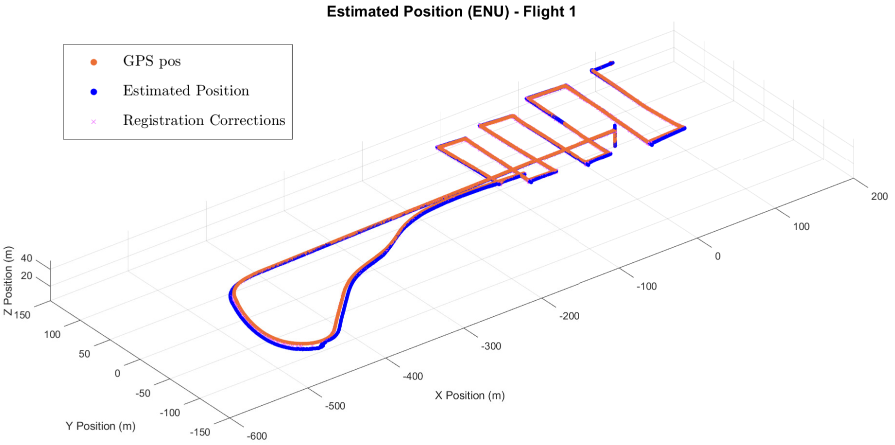
RESULTS

To acquire data for GPS-denied navigation, two flights were documented, during which the texel camera was affixed to a sUAV, as introduced earlier in this thesis. Both flights followed similar paths, yet yielded distinct datasets. The sUAV traversed the Utah State University campus, initially maintaining a straight trajectory at speeds of 15 m/s. Upon reaching the Old Main building, the sUAV retraced its path to the starting point, creating multiple flightline pathways.

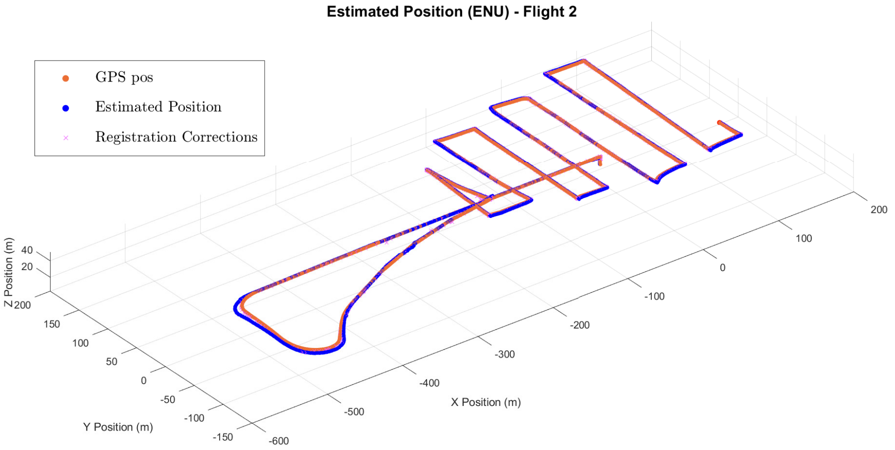
The first flight covered a recorded and estimated distance of 3.79783 kilometers, while the second path spanned 4.76594 kilometers. The results indicate a close alignment between the estimated position and the tracked sUAS path, with an average error of 4.21956 meters for the first flight and 4.14791 meters for the second collection. This error averages out to 4.18373 meters, which is within the range of U.S civilian based GPS accuracy of 3 to 5 meters as stated in the U.S. Department of Defenses' GPS positioning service performance standard [47].

The resulting estimated pathways can be seen in the plotted diagrams of Figure 5.1a and 5.1b. To better view the estimation performance and registration corrections, Figure 5.2 represents the first flight of data capture. Within this plot, GPS position is depicted by the orange data points, while the blue points represent the estimated position. The blue points signify the estimated position incorporating texel registration corrections, while the magenta "x" marks represent the corresponding pose corrections. The processed data is also represented by position in meters from the starting point. From our flight path, the starting point at (0,0,0) was the time of capture once the sUAV reached 15 meters in altitude.

Applying overlaid GPS location of the flights recorded, we can visually see in Figure 5.3 the performance of the estimator over urban landmarks such as buildings, parking lots,



(a)



(b)

Fig. 5.1: Estimated position vs GPS position for the recorded flight #1 (a) flight #2 (b).

courtyards, and walkways. This figure contains the color-mapped results representing the error of the estimator compared to the GPS, where the black line represents GPS position.

It shall be noted that certain sections of these flights perform worse than others. These sections seem to involve areas with less features for the registration process. If the sUAV flies over areas with a lack of features such as grassy areas or snowy landscapes, this can throw off the registration results due to the bundle adjustment's reliance of image features. Another possibility in error is the speed of the flight at these positions. Until the flightlines that are halfway through each flight, the system was travelling close to 30 mph, while 15

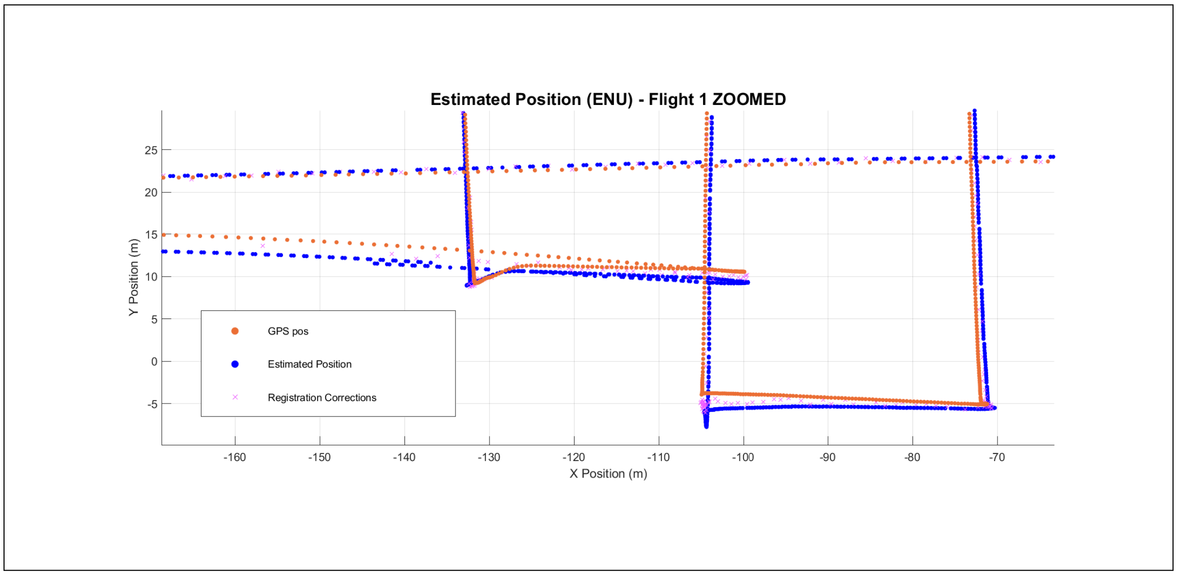
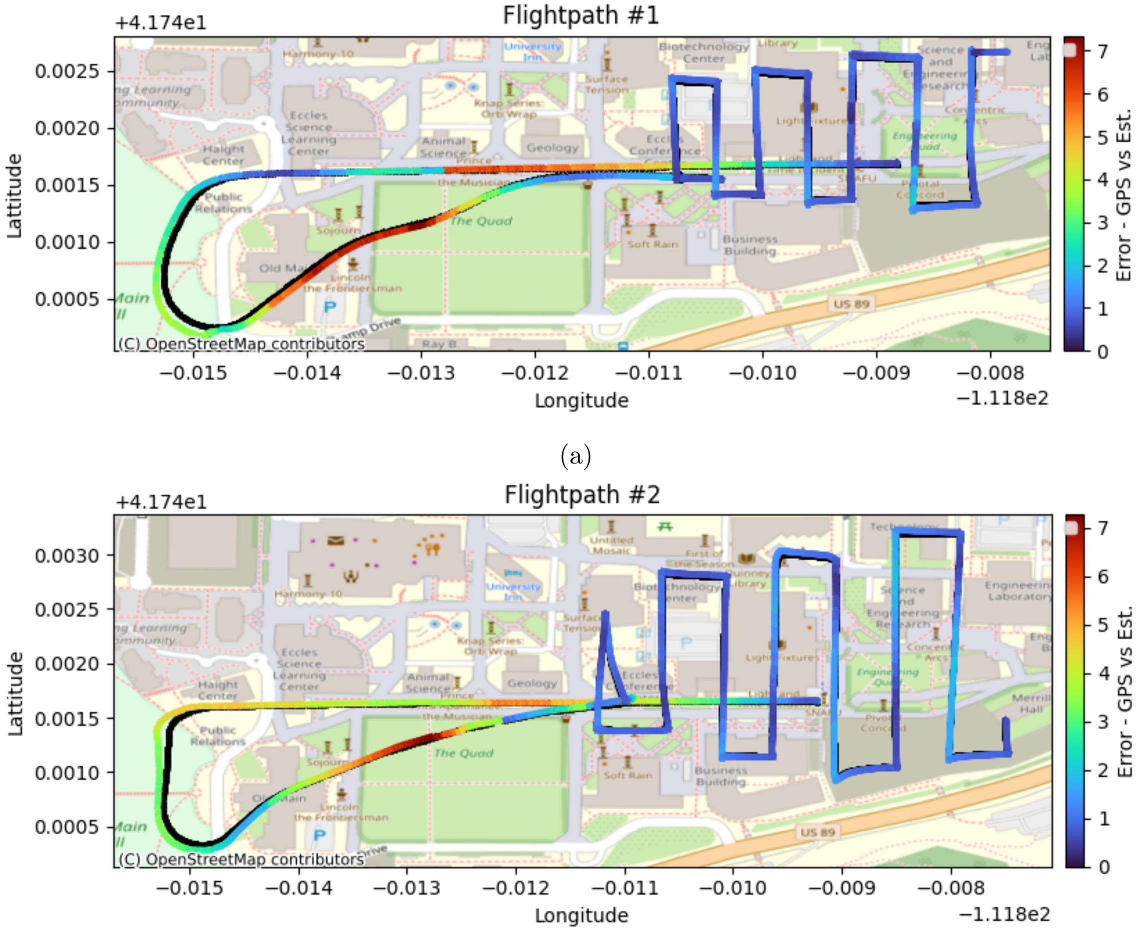


Fig. 5.2: Zoomed in results of the first flight of data capture to highlight the GPS position (orange), estimated position (blue), and registration corrections (pink).

mph at the time of the flightlines. The speed of the system does effect how far apart images are. The space of imagery could determine how well the registration process performs and relate to why the estimation performed worse at the first straight line from the start of the two flights. As for the section near The Quad, the system seemed to be trying to catch up to a correct altitude. This problem in altitude could relate to the system previously flying over an overall sloped area near Old Main or from error in imagery. More data over a variable slope of terrain could show a more viable conclusion to this error during the flight.



Flightpath #1

(a)

Flightpath #2

(b)

Fig. 5.3: Estimated position vs GPS position overlaid onto a map containing location features for flight #1 (a) and flight #2 (b) (Sourced by OpenStreetMap under the Open Database Liscense.)

CHAPTER 6
CONCLUSION & FUTURE WORK

6.1 Conclusion

The effectiveness of GPS-Denied navigation, aided by LiDAR and image registration, was demonstrated to achieve accuracy comparable to GPS. Employing an Extended Kalman Filter (ESEKF) for the measured IMU velocity and rotation, along with texel registration to align LiDAR and imagery with pose, yielded results indicating an average accuracy of approximately 4.31720 meters from the measured GPS location for an average flight distance of around 4.281885 kilometers. These findings demonstrate a reliable accuracy close to that of GPS within a reasonable distance for low-cost sUAV flights.

6.2 Future Work

Future work of this research is to gather dependable data from numerous texel registration cases. The dataset used in this study was acquired from Utah State Universities campus, which exhibited a relatively constrained range of landmarks. Although the collected data included buildings, vegetation, and flat terrains, it does not represent the performance of forested areas, regions with bodies of water, or areas characterized by significant changes in elevation, such as mountains, hills, or canyons.

The presence of these landmarks has the potential to impact texel registration performance and influence the effectiveness of bundle adjustments. Surveyed areas with limited features in the imagery may present challenges for texel registration, as it could struggle to identify image features. Additionally, issues can arise when dealing with tree canopies, as the LiDAR data points may be affected by returns that pass through the canopy, thereby impacting the registration process as well. By incorporating more features in the texel imagery, it is expected to enhance texel registrations and consequently improve position estimation.

Another direction this research could take is improving processing speeds while without data loss. During registration, optimizing the bundle adjustments using the LM algorithm is used for large LiDAR point clouds. While the LM algorithm is one of the most effective optimization processes for this research, the computations are costly with a complexity of $\mathcal{O}(n^3)$ [28]. When dealing with denser datasets, the LM algorithm's complexity can lead to increased costs and longer processing times. To address this issue, future efforts could focus on incorporating alternative optimization methods like Gauss-Newton, Conjugate Gradient, or Quasi-Newton. Although each method has its limitations, they offer faster processing or training time compared to the LM algorithm [48].

Lastly, it's worth noting that the IMU data collected for this research was sampled at a low rate. The process of GPS-Denied navigation using LM corrections and the ESEKF are heavily reliant on the IMU data sampled. If this data is noisy, inaccurate, or sampled too low, this process will return poor results. Implementing a separate processing thread for IMU data collection while ensuring calibrated sensors could significantly enhance the quality of this data, as higher sample rates would contribute to improved accuracy in the ESEKF.

REFERENCES

[1] J. Wang, S. Wang, D. Zou, H. Chen, R. Zhong, H. Li, W. Zhou, and K. Yan, "Social network and bibliometric analysis of unmanned aerial vehicle remote sensing applications from 2010 to 2021," *Remote Sensing*, vol. 13, no. 15, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/15/2912

[2] D. Bevly, J. Gerdes, C. Wilson, and G. Zhang, "The use of gps based velocity measurements for improved vehicle state estimation," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, vol. 4, 2000, pp. 2538–2542 vol.4.

[3] S.-M. Oh, "Multisensor fusion for autonomous uav navigation based on the unscented kalman filter with sequential measurement updates," in *2010 IEEE Conference on Multisensor Fusion and Integration*, 2010, pp. 217–222.

[4] W. You, F. Li, L. Liao, and M. Huang, "Data fusion of uwb and imu based on unscented kalman filter for indoor localization of quadrotor uav," *IEEE Access*, vol. 8, pp. 64 971–64 981, 2020.

[5] G. Ariante, U. Papa, S. Ponte, and G. Del Core, "Uas for positioning and field mapping using lidar and imu sensors data: Kalman filtering and integration," in *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2019, pp. 522–527.

[6] R. Ferreira, J. Gaspar, N. Souto, and P. Sebastião, "Effective GPS jamming techniques for UAVs using low-cost SDR platforms," in *2018 Global Wireless Summit (GWS)*, 2018, pp. 27–32.

[7] M. Narasimhappa, A. D. Mahindrakar, V. C. Guizilini, M. H. Terra, and S. L. Sabat, "MEMS-based IMU drift minimization: Sage husa adaptive robust kalman filtering," *IEEE Sensors Journal*, vol. 20, no. 1, pp. 250–260, 2020.

[8] G. M. Reis, M. Fitzpatrick, J. Anderson, J. Kelly, L. Bobadilla, and R. N. Smith, "Increasing persistent navigation capabilities for underwater vehicles with augmented terrain-based navigation," *OCEANS 2017 - Aberdeen*, pp. 1–8, 2017.

[9] H. Lu, H. Shen, B. Tian, X. Zhang, Z. Yang, and Q. Zong, "Flight in GPS-denied environment: Autonomous navigation system for micro-aerial vehicle," *Aerospace Science and Technology*, vol. 124, p. 107521, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S127096382200195X

[10] J. Melo and A. Matos, "Survey on advances on terrain based navigation for autonomous underwater vehicles," *Ocean Engineering*, vol. 139, pp. 250–264, 07 2017.

[11] I. Brigadnov, A. Lutonin, and K. Bogdanova, "Error state extended kalman filter localization for underground mining environments," *Symmetry*, vol. 15, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2073-8994/15/2/344

[12] R. A. Chisholm, J. Cui, S. K. Y. Lum, and B. M. Chen, "Uav lidar for below-canopy forest surveys," *Journal of Unmanned Vehicle Systems*, vol. 01, no. 01, pp. 61–68, 2013. [Online]. Available: https://doi.org/10.1139/juvs-2013-0017

[13] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proceedings of Robotics: Science and Systems (RSS '14)*, July 2014.

[14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[15] D. Kiss-Illés, C. Barrado, and E. Salamí, "Gps-slam: An augmentation of the orb-slam algorithm," *Sensors*, vol. 19, no. 22, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/4973

[16] J. Qian, K. Chen, Q. Chen, Y. Yang, J. Zhang, and S. Chen, "Robust visual-lidar simultaneous localization and mapping system for uav," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[17] G. Hemann, S. Singh, and M. Kaess, "Long-range GPS-denied aerial inertial navigation with LIDAR localization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1659–1666.

[18] M. M. Miller, A. Soloviev, M. Uijt de Haag, M. Veth, J. Raquet, T. J. Klausutis, and J. E. Touma, "Navigation in gps denied environments: Feature-aided inertial systems," Air force research lab eglin afb fl munitions directorate, Tech. Rep., 2010.

[19] B. M. Boldt, S. E. Budge, R. T. Pack, and P. D. Israelsen, "A handheld texel camera for acquiring near-instantaneous 3d images," in *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, 2007, pp. 953–957.

[20] T. Bybee and S. Budge, "Method for 3-D scene reconstruction using fused lidar and imagery from a texel camera," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–11, 07 2019.

[21] B. Khatiwada and S. E. Budge, "Three-dimensional image reconstruction using bundle adjustment applied to multiple texel images," in *Laser Radar Technology and Applications XXI*, M. D. Turner and G. W. Kamerman, Eds., vol. 9832, International Society for Optics and Photonics. SPIE, 2016, p. 98320S. [Online]. Available: https://doi.org/10.1117/12.2223259

[22] T. Welch, C. Coopmans, and S. Budge, "Development of a small unmanned aerial system-mounted texel camera," 05 2019, p. 11.

[23] Y. Zhang, Y. Onda, H. Kato, B. Feng, and T. Gomi, "Understory biomass measurement in a dense plantation forest based on drone-sfm data by a manual low-flying drone under the canopy," *Journal of Environmental Management*, vol. 312, p. 114862, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301479722004352

[24] T. Suzuki, D. Miyoshi, J. ichi Meguro, Y. Amano, T. Hashizume, K. Sato, and J. ich Takiguchi, "Real-time hazard map generation using small unmanned aerial vehicle," in *2008 SICE Annual Conference*, 2008, pp. 443–446.

[25] W. Adi, A. Aghastya, N. Imron, N. Apriliani, I. Anwer, and P. Puangprakhon, "Comparison of uav drone and online terrain model for railway route planning," *Journal of Railway Transportation and Technology*, vol. 1, pp. 20–27, 05 2022.

[26] B. Khatiwada, "Super-resolution textured digital surface map (dsm) formation by selecting the texture from multiple perspective texel images taken by a low-cost small unmanned aerial vehicle (uav)," Ph.D. dissertation, Utah State University, 2023, all Graduate Theses and Dissertations, Spring 1920 to Summer 2023. [Online]. Available: https://doi.org/10.26076/341c-5ad6

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis," *ICCV '99 Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pp. 198–372, 01 2000.

[29] H. P. Gavin, "The levenberg-marquardt method for nonlinear least squares curve-fitting problems c ©," 2013.

[30] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. USA: Cambridge University Press, 2003.

[31] G. Krishnaswami and S. Sachdev, "Algebra and geometry of hamilton's quaternions," 06 2016.

[32] J. Pujol, "Hamilton, Rodrigues, Gauss, Quaternions, and Rotations: a Historical Reassessment," *Communications in Mathematical Analysis*, vol. 13, no. 2, pp. 1 – 14, 2012.

[33] E. Hemingway and O. O'Reilly, "Perspectives on euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments," *Multibody System Dynamics*, vol. 44, 09 2018.

[34] J. Solà, "Quaternion kinematics for the error-state kalman filter," *CoRR*, vol. abs/1711.02508, 2017. [Online]. Available: http://arxiv.org/abs/1711.02508

[35] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended kalman filter vs. error state kalman filter for aircraft attitude estimation," 08 2011.

[36] Y. Y. Nazaruddin, F. A. Maani, E. R. Muten, P. W. L. Sanjaya, G. Tjahjono, and J. A. Oktavianus, "An approach for the localization method of autonomous vehicles in the event of missing gnss information," in *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2021, pp. 881–886.

[37] M. Avzayesh, M. Abdel-Hafez, M. AlShabi, and S. Gadsden, “The smooth variable structure filter: A comprehensive review,” *Digital Signal Processing*, vol. 110, p. 102912, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200420302578

[38] A. Guo, Z. Zhou, X. Zhu, and F. Bai, “Low-cost sensors state estimation algorithm for a small hand-launched solar-powered UAV,” *Sensors*, vol. 19, no. 21, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/21/4627

[39] K. Yamane and J. Hodgins, “Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2510–2517.

[40] D. Noonan, P. Mountney, D. Elson, A. Darzi, and G.-Z. Yang, “A stereoscopic fibroscope for camera motion and 3d depth recovery during minimally invasive surgery,” 05 2009, pp. 4463–4468.

[41] S. Sharma, S. Verma, M. Kumar, and L. Sharma, “Use of motion capture in 3d animation: Motion capture systems, challenges, and recent trends,” in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMIT-Con)*, 2019, pp. 289–294.

[42] E. Ceseracciu, Z. Sawacha, and C. Cobelli, “Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: Proof of concept,” *PloS one*, vol. 9, p. e87640, 03 2014.

[43] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnp: An accurate o(n) solution to the pnp problem,” *International Journal of Computer Vision*, vol. 81, 02 2009.

[44] S. Li, C. Xu, and M. Xie, “A robust o(n) solution to the perspective-n-point problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.

[45] S. E. Budge and N. S. Badamikar, “Calibration method for texel images created from fused flash lidar and digital camera images,” *Optical Engineering*, vol. 52, no. 10, p. 103101, 2013. [Online]. Available: https://doi.org/10.1117/1.OE.52.10.103101

[46] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence properties of the nelder-mead simplex method in low dimensions,” *SIAM J. Optim.*, vol. 9, pp. 112–147, 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:9245771

[47] U.S. Dept. of Defense, “Global positioning system standard positioning service performance standard,” U.S. Dept. of Defense, Washington, DC, Tech. Rep., April 2020. [Online]. Available: https://www.gps.gov/technical/ps/2020-SPS-performance-standard.pdf

[48] I. Mukherjee and S. Routroy, “Comparing the performance of neural networks developed by using levenberg–marquardt and quasi-newton with the gradient descent algorithm for modelling a multiple response grinding process,” *Expert Systems with Applications*, vol. 39, no. 3, pp. 2397–2407, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417411012188