Visual Localization for UAVs in Outdoor GPS-denied Environments

by

Bhavit Patel

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of University of Toronto Institute for Aerospace Studies
University of Toronto

Abstract

Visual Localization for UAVs in Outdoor GPS-denied Environments

Bhavit Patel
Master of Applied Science
Graduate Department of University of Toronto Institute for Aerospace Studies
University of Toronto
2019

Vision-based navigation techniques are commonly used for autonomous flight in outdoor Global Positioning System (GPS)-denied environments. We adapt Visual Teach and Repeat (VT&R), a vision-based autonomous route-following system, to use on multirotor Unmanned Aerial Vehicles (UAVs). Since multirotors are underactuated, there are no guarantees that the camera viewpoint will be the same at matching positions along the teach and repeat paths, which causes visual localization to be challenging. The first part of this thesis demonstrates that by using a 3-axis gimballed camera with an appropriate active pointing strategy, we can improve the visual localization performance and robustness within the VT&R framework. The second part of this thesis presents a method to estimate the global pose of a UAV by using an information-theoretic approach to register real images with rendered georeferenced images from 3D Google Earth. We show that this method is capable of accurately estimating the 6DoF pose with a position accuracy on par with GPS.

Contents

Notation

$\underrightarrow{\mathcal{F}}_a$	A reference frame for a three dimensional coordinate system
$SE(3)$	The Special Euclidean Group in three dimensions used to represent rigid body transformations and poses, a matrix Lie group
$\mathfrak{se}(3)$	The Lie algebra vectorspace associated with $SE(3)$
$SO(3)$	The Special Orthogonal Group in three dimensions used to represent rigid body rotations, a matrix Lie group
$\mathfrak{so}(3)$	The Lie algebra vectorspace associated with $SO(3)$
$\mathbf{T}_{b,a}$	A matrix in $SE(3)$ that transforms points expressed in $\underrightarrow{\mathcal{F}}_a$ to $\underrightarrow{\mathcal{F}}_b$
$\mathbf{C}_{b,a}$	A matrix in $SO(3)$ that rotates points expressed in $\underrightarrow{\mathcal{F}}_a$ to $\underrightarrow{\mathcal{F}}_b$
$\mathbf{r}_b^{a,b}$	A vector in $\mathbb{R}^{3\times 1}$ representing the translation of the origin of $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$ expressed in $\underrightarrow{\mathcal{F}}_b$
$\exp(\cdot^\wedge)$	A Lie algebra operator mapping from $\mathfrak{se}(3)$ to $SE(3)$ or $\mathfrak{so}(3)$ to $SO(3)$
$\ln(\cdot)^\vee$	A Lie algebra operator mapping from $SE(3)$ to $\mathfrak{se}(3)$ or $SO(3)$ to $\mathfrak{so}(3)$
$(\check{\cdot})$	A prior value
$(\hat{\cdot})$	A posterior value
$(\cdot)^*$	An optimized value
$\mathcal{N}(\mu, \sigma^2)$	A normal distribution with mean μ and variance σ^2

Chapter 1

Introduction

1.1 Background and Motivation

The demand for using Unmanned Aerial Vehicles (UAVs) in a variety of industrial and commercial applications has rapidly risen due to the their technological advancements over the past decade; emergency response, surveillance and reconnaissance, agriculture, remote sensing and monitoring, and delivery services are just a few of many expected applications. However, one crucial element that is limiting their widespread use is safety.

The majority of UAVs available today are capable of autonomous navigation using Global Positioning System (GPS) and inertial sensors. This reliance on GPS poses a problem for environments where poor satellite coverage, multipath propagation, and intentional jamming can hinder its use. As a result, government regulations generally restrict the use of UAVs to Visual Line of Sight (VLOS) operations to allow a human to manually pilot the vehicle in the event of GPS loss. To enable beyond VLOS operations and expand the scope of UAV applications, there is a need to develop safe and robust autonomous navigation solutions that can serve as standalone or backup solutions during GPS loss.

Vision-based autonomous navigation techniques are commonly used for UAVs in GPS-denied environments due to the light weight, low power consumption, and low cost of cameras. The majority of these techniques involve Visual Odometry (VO). VO alone is unreliable for accurate pose estimates since it ultimately drifts in the absence of corrections. Visual Simultaneous Localization and Mapping (SLAM) corrects these drifts through loop closure and has been sucessfully demonstrated in GPS-denied environments [Blösch et al., 2010, Weiss et al., 2013, Shen et al., 2015] but requires revisiting locations. On the other hand, VT&R [Furgale and Barfoot, 2010] can enable safe navigation without requiring globally accurate poses. VT&R is a route-following technique that enables long-range autonomous navigation without reliance on external positioning systems such as GPS. While its development has largely focused on ground vehicles, we adapt it for use on multirotor UAVs.

We use the VT&R framework to develop a redundant navigation system that enables the safe, autonomous return of multirotor UAVs in the event of primary navigation failure such as GPS loss. This system has numerous use cases including drone delivery and surveillance and reconnaissance where an attacker can be prevented from stealing an expensive package or recovering sensitive information by intentionally jamming GPS. During the outbound flight where the UAV is manually piloted or under autonomous GPS waypoint control, a visual map is generated using only a stereo camera and performing sparse feature-based VO. Following a GPS loss, the UAV is able to return to the takeoff location by autonomously navigating backwards along the outbound flight path using a vision-based flight controller.

Although part of the contribution of this thesis is engineering work and experimental verification of the VT&R system for emergency return, we opt to not include a separate chapter detailing the system. This thesis will instead focus on the contributions made in addressing two challenges: 1) handling large camera viewpoint changes between the teach and repeat flights due to the underactuated nature of multirotor UAVs, and 2) enabling flight along routes untraversed by the vehicle itself.

We address the first challenge by using a 3-axis gimbal to actively point the camera. Chapter 2 details various pointing strategies and compares their performance in multiple outdoor flight tests. We also demonstrate closed-loop vehicle control with active camera pointing to conclude our initial verification of using the VT&R system for emergency return of a multirotor UAV.

The second challenge arises from the desire to continue to the destination after GPS loss, especially if the destination is closer than the takeoff location, and perform emergency return along a route more efficient than the outbound. To address this challenge, we initially forego the use of a teach-and-repeat style technique, and instead solve the more general problem of global pose estimation. We develop a method to accurately estimate the global pose of a UAV by visually localizing using a set of georeferenced images. Chapter 3 presents the pose estimation method and successful results on real-world data.

1.2 Related Work

Vision-based autonomous navigation for UAVs in GPS-denied environments is a popular research area with many groups showing successful demonstrations under limited conditions. We briefly present a few examples with a focus on SLAM and teach-and-repeat style techniques. Related work pertaining to gimbals on UAVs (Chapter 2) and localization with georeferenced images (Chapter 3) are detailed in their respective chapters.

Visual SLAM techniques have been sucessfully demonstrated for GPS-denied environments in indoor [Blösch et al., 2010, Weiss et al., 2011] and outdoor settings [Achtelik et al., 2011, Weiss et al., 2013, Shen et al., 2013]. In many cases, a fixed downward facing monocular camera along with an Inertial Measurement Unit (IMU) is used [Blösch et al., 2010, Weiss et al., 2011, Achtelik et al., 2011, Weiss et al., 2013]. Shen et al. [2013] added a stereo camera as a secondary camera for metric scale.

Work in [Blösch et al., 2010, Weiss et al., 2011, Achtelik et al., 2011, Weiss et al., 2013] use Parallel Tracking and Mapping (PTAM) [Klein and Murray, 2007] implementation of visual SLAM, which splits the localization and mapping into two separate threads allowing high rate motion estimation with a lower rate mapping. Extensions to [Blösch et al., 2010] allow all processing to be performed onboard the UAV by reducing the number of features tracked and limiting the number of keyframes stored [Weiss et al., 2011]. The pose estimates were fused with an IMU in an Extended Kalman Filter (EKF) to self-calibrate the IMU and visual drifts [Weiss et al., 2013] resulting in a 1.47 m final position error from autonomous navigation along a 360 m outdoor flight with 70 m altitude change. While the keyframe limit enables real-time processing onboard, it also reduces the chance of loop closure in large environments due to the keyframe dropping. Additionally, revisiting a location may require remapping.

LIDAR SLAM has also been demonstrated on UAVs to explore and map indoor [Bachrach et al., 2009] and outdoor [Bachrach et al., 2011] environments. Huh et al. [2013] use a LIDAR, monocular camera, and IMU for indoor and outdoor autonomous flight. The LIDAR scans are used to estimate the depth of all features detected in a camera image for visual SLAM. However, in both cases the mapping and path planning was offloaded to a powerful ground station. Also, the high power consumption and weight of LIDAR sensors make them unsuitable for smaller multirotor UAVs.

The teach phase of VT&R differs from SLAM in two key ways. The first is that most SLAM techniques require the creation of monolithic, globally consistent maps whereas VT&R does not have this requirement. Instead, locally consistent overlapping maps are created during the teach run. During the repeat runs, the mobile robot can be localized against the locally consistent map at the nearest keyframe. Second, the human-operated teach phase precludes the need to develop a safe autonomous exploration algorithm.

A proof-of-concept demonstration of VT&R on multirotor UAVs was shown in [Pfrunder et al., 2014]. A monocular downward facing camera was utilized with an altitude sensor for scale to autonomously follow an 8 m straight and constant altitude route. Warren et. al adapted the localization engine of VT&R to successfully localize a fixed-wing UAV with a fixed downward facing camera over a taught 1200 m trajectory at an altitude of 80 m [Warren et al., 2018]. We extend this work to improve the visual localization and close the loop with a vision-based flight controller for our emergency return system.

Recently, there have also been demonstrations of similar teach-and-repeat style techniques for visual navigation [Toudeshki et al., 2018, Surber et al., 2017]. Sparse features are replaced with semantic objects in [Toudeshki et al., 2018] but this requires reliable detection of distinct objects in real time. Surber et al. [2017] perform map building offline thus their method cannot be used as an emergency return solution.

1.3 $SE(3)$ Math Overview

In this section, we define some of the notation used to represent poses and transforms, as well as provide a brief overview of important operators associated with the Special Euclidean Group in three dimensions ($SE(3)$) that will be used throughout this thesis. We refer the reader to [Barfoot, 2017] for a more detailed explanation.

$SE(3)$ represents rigid body transformations and poses. Similarly, the Special Orthogonal Group in three dimensions ($SO(3)$) represents rigid body rotations. In this work, we define

$$\mathbf{T}_{a,b} = \begin{bmatrix} \mathbf{C}_{a,b} & \mathbf{r}_a^{b,a} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in SE(3) \tag{1.1}$$

as the 4×4 transformation matrix that transforms points in coordinate frame $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$. This transformation matrix consists of a 3×3 rotation matrix, $\mathbf{C}_{a,b} \in SO(3)$, and a translation vector, $\mathbf{r}_a^{b,a} = [x_a^{b,a} \; y_a^{b,a} \; z_a^{b,a}]^\top$. The superscript of $\mathbf{r}_a^{b,a}$ indicates it is a vector from the origin of $\underrightarrow{\mathcal{F}}_a$ to $\underrightarrow{\mathcal{F}}_b$, while the subscript indicates the vector is expressed in $\underrightarrow{\mathcal{F}}_a$.

Both $SE(3)$ and $SO(3)$ are matrix Lie groups, and have an associated Lie algebra, $\mathfrak{se}(3)$ and $\mathfrak{so}(3)$, respectively. The Lie group and Lie algebra can be related through an exponential and logarithmic mapping. Here, we will only describe the relationship for $SE(3)$ and $\mathfrak{se}(3)$ as the relationship for $SO(3)$ and $\mathfrak{so}(3)$ is quite similar. The exponential map takes us from $\mathfrak{se}(3)$ to $SE(3)$:

$$\mathbf{T}_{a,b} = \exp(\boldsymbol{\xi}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\boldsymbol{\xi}^\wedge)^n, \tag{1.2}$$

where $\boldsymbol{\xi} = [\boldsymbol{\rho} \; \boldsymbol{\phi}]^\top \in \mathbb{R}^6$ is a pose vector with $\boldsymbol{\rho} \in \mathbb{R}^3$ as the translation and $\boldsymbol{\phi} \in \mathbb{R}^3$ as the rotation components, and

$$\boldsymbol{\xi}^{\wedge} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^{\wedge} = \begin{bmatrix} \boldsymbol{\phi}^{\times} & \boldsymbol{\rho} \\ \mathbf{0}^{\top} & 1 \end{bmatrix} \in \mathfrak{se}(3) \tag{1.3}$$

is a 4 × 4 matrix in the Lie algebra vectorspace. The $(\cdot)^{\times}$ operator creates a skew-symmetric matrix. The logarithmic map takes us from $SE(3)$ to $\mathfrak{se}(3)$ so we can obtain a pose vector from a transformation matrix:

$$\boldsymbol{\xi} = \ln(\mathbf{T}_{a,b})^{\vee}. \tag{1.4}$$

If we let $\mathbf{T}_{a,b}$ be an uncertain transform with uncertainty $\boldsymbol{\Sigma}_{a,b}$, then we can use an $SE(3)$ perturbation scheme to represent it as

$$\mathbf{T}_{a,b} = \exp(\boldsymbol{\epsilon}^{\wedge})\bar{\mathbf{T}}_{a,b}, \quad \boldsymbol{\epsilon} \in \mathbb{R}^6 \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{a,b}), \tag{1.5}$$

where $\bar{\mathbf{T}}_{a,b}$ is the nominal (i.e., mean) transformation, and $\boldsymbol{\epsilon}$ is a small Gaussian random pose perturbation. Finally, the adjoint of an $SE(3)$ transform,

$$\boldsymbol{\mathcal{T}}_{a,b} = \mathrm{Ad}(\mathbf{T}_{a,b}) = \mathrm{Ad}\left(\begin{bmatrix} \mathbf{C}_{a,b} & \mathbf{r}_a^{b,a} \\ \mathbf{0}^{\top} & 1 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{C}_{a,b} & (\mathbf{r}_a^{b,a})^{\times}\mathbf{C}_{a,b} \\ \mathbf{0}^{\top} & \mathbf{C}_{a,b} \end{bmatrix}, \tag{1.6}$$

is a 6 × 6 matrix that we will make use of when compounding uncertain transforms.

1.4 Contributions

The main contributions of this thesis are:

1. Aided in the development of a vision-based emergency return system for UAVs and experimentally verified the system in multiple field tests at: University of Toronto Institute for Aerospace Studies (UTIAS); Koffler Scientific Reserve; Suffield, Alberta; and downtown Montreal. This work resulted in a journal publication as a contributing author:
   Michael Warren, Melissa Greeff, Bhavit Patel, Jack Collier, Angela P. Schoellig, and Timothy D. Barfoot. There's no place like home: Visual teach and repeat for emergency return of multirotor UAVs during GPS failure. *IEEE Robotics and Automation Letters*, 4(1):161–168, 2019. doi: 10.1109/LRA.2018.2883408.

2. A modular gimbal controller that resides within the VT&R system and is capa-

ble of performing orientation matching or centroid pointing. Both strategies were demonstrated to improve the visual localization performance within the VT&R framework. This work resulted in a conference paper as first author and received a best paper award:

Bhavit Patel, Michael Warren, and Angela P. Schoellig. Point me in the right direction: Improving visual localization on UAVs with active gimballed camera pointing. In *Proc. of the Conference on Computer and Robot Vision (CRV)*, 2019.

Chapter 2 consists of this paper with some modifications and additional details.

3. A method to estimate the 6 Degree of Freedom (DoF) global pose of a UAV using a dense Mutual Information (MI) based image registration technique to metrically localize real images with rendered georeferenced images from Google Earth (GE). This work resulted in a journal paper submission as first author:

   Bhavit Patel, Timothy D. Barfoot, and Angela P. Schoellig. Visual localization with google earth images for robust global pose estimation of UAVs. *IEEE Robotics and Automation Letters*, 2020. Submitted.

   Chapter 3 consists of this paper with some modifications and additional details.

Chapter 2

Gimballed Camera Pointing

2.1 Motivation

One challenge associated with localization on multirotor UAVs for VT&R is that the camera viewpoint can be extremely different between teach and repeat flights. The underactuated nature of multirotor UAVs causes a camera mounted statically to the vehicle to undergo large viewpoint changes during accelerations. These viewpoint changes are an issue as many visual localization techniques rely on matching feature descriptors such as Speeded-Up Robust Features (SURF) [Bay et al., 2008], which are known to be highly sensitive to scene perspective changes. While a vehicle controller tries to keep the vehicle close to the originally flown path, there are no guarantees that the camera viewpoint will be the same at matching positions along the teach and repeat flights unless the UAV follows an identical acceleration profile.

To address this problem, we use a 3-axis gimbal to fully decouple the camera and vehicle orientations. Moreover, the gimbal allows independent camera viewpoint manipulation to improve visual localization robustness under conditions of high winds, large path-following errors, and faster flight speeds compared to the map-generation flight. The benefit is most apparent in scenarios where the scene is spatially close to the camera (such as when flying near the ground or in close proximity to buildings). In these situations, any small viewpoint errors result in a large reduction in image overlap, which makes it difficult to visually localize. Such close proximity flights are common in monitoring and inspection applications or when operating in urban environments. The use of a static camera in these scenarios is prone to localization failures from large perspective errors.

We use an active gimballed camera on a multirotor UAV in a similar manner as done in [Warren et al., 2018] for ground vehicles and in [Warren et al., 2019] for UAVs. We improve the response time of the gimbal controller by using angular rate commands to handle the UAV's fast dynamics. We also introduce a centroid pointing strategy as an alternative to orientation matching. Finally, we perform multiple outdoor flight experiments to i) highlight the robustness an active gimballed camera adds over a static camera, ii) show that an off-the-shelf passively stabilized gimbal can actually be detrimental for localization, and iii) demonstrate the ability of orientation matching and centroid pointing strategies to enable visual localization despite large path-following errors and velocity discrepancies. In this work, we define active strategies as those that require user control input and are further divided into those that use visual information to determine where to point the camera (e.g., orientation matching and centroid pointing) and those that simply stabilize the camera (e.g., active stabilization). Passive strategies, on the other hand, require no user control input.

2.2 Related Work

Early work using gimballed cameras on UAVs involved applications unrelated to vision-based navigation: they were used to increase the effectiveness of target tracking and surveillance [Quigley et al., 2005, Skoglar, 2002, Skoglar et al., 2012], and search and rescue [Goodrich et al., 2008]. Non-static cameras have been used for vision-based landing of UAVs: a pan-tilt monocular camera was utilized to increase the effective Field Of View (FoV) during landing [Sharp et al., 2001], and 3-axis gimballed monocular cameras were leveraged for autonomous landing on moving platforms [Borowczyk et al., 2017, Wang et al., 2017].

The majority of vision-based autonomous navigation solutions for UAVs use static cameras [Blösch et al., 2010, Weiss et al., 2011, Achtelik et al., 2011, Weiss et al., 2013, Shen et al., 2013, Pfrunder et al., 2014, Warren et al., 2017, Toudeshki et al., 2018, Surber et al., 2017]. Recent work demonstrates the integration of gimballed cameras with Visual-Inertial Odometry (VIO) [Choi et al., 2018] and visual SLAM [Playle, 2015]. Work in [Playle, 2015] performs a reactive viewpoint selection strategy by panning the camera to areas of high feature density with the goal of improving localization accuracy of monocular visual SLAM using a two-axis gimbal. In contrast, we perform a predictive strategy and control in more than one axis.

The most closely related work is our previous work demonstrating the use of VT&R as an emergency return system on multirotor UAVs [Warren et al., 2019] using the active gimballed camera implementation introduced in [Warren et al., 2018]. While work in [Warren et al., 2019] shows successful localization using an orientation matching strategy, in this work, in addition to improving the gimbal controller implementation, we perform new outdoor experiments to show the improvement and robustness that an active gimballed camera adds over a passive gimbal and static camera for visual localization.

2.3 VT&R Overview

In this section we provide a high-level overview of the VT&R system focussing on the localization as it pertains to the gimbal pointing. We refer the reader to Section 3.3 for more detailed explanation of the VO, and previous work for the localization [Paton et al., 2016] and multirotor UAV emergency return adaptation with closed-loop vehicle control [Warren et al., 2019].

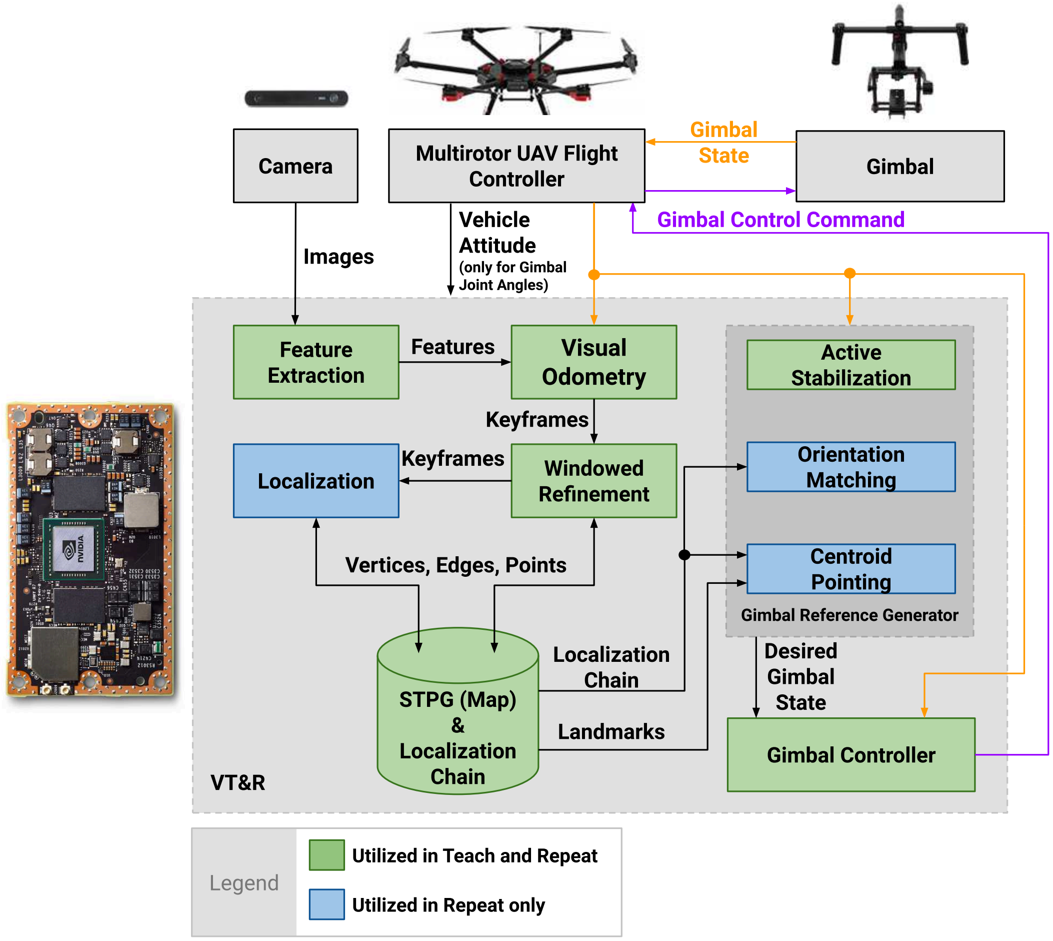


Figure 2.1: A simplified overview of the vision-based localization system with an active gimballed camera. During teach, the phase where the map is created, active stabilization can be performed while in repeat, the phase where the map is used for localization, any of the proposed gimbal pointing strategies can be selected.

During a human-piloted or autonomous GPS waypoint outbound flight, termed the teach phase, a visual map is generated using a stereo camera and performing sparse feature-based VO. Following a GPS loss, the UAV returns home by autonomously navigating backwards along the outbound flight path using a vision-based flight controller and a gimbal controller to promote localization; this is the repeat phase although it is different than a typical repeat phase since we are repeating the path in the opposite direction.

Figure 2.1 shows an overview of the VT&R software system without the vehicle controller. We include a new gimbal controller implementation that allows active control in both the teach and repeat phases with faster response times. The new implementation also provides the ability to select different pointing methods to use in each phase.

During an outbound teach flight, sparse feature-based gimballed VO is performed to estimate the pose of the vehicle and scene structure using only the stereo images and gimbal joint angles captured at 10 Hz. The visual observations are inserted into a relative map of pose and scene structure in the form of a Spatio-Temporal Pose Graph (STPG) (see Figure 2.2). Each vertex α stores the 3D positions of landmarks with associated covariances observed by the camera, $\{\mathbf{p}_j^\alpha, \boldsymbol{\sigma}_j^\alpha\}$, and the non-static vehicle-to-sensor transform, $\mathbf{T}_{\alpha_s,\alpha}$ (i.e., the pose of the vehicle in the camera frame at vertex α). The vehicle-to-sensor transform is obtained by applying forward kinematics with the roll, pitch, and yaw gimbal angular positions. Edges link temporally and spatially adjacent vertices metrically with a 6DoF $SE(3)$ transformation, $\mathbf{T}_{\alpha,\alpha-1}$, with uncertainty $\boldsymbol{\Sigma}_{\alpha,\alpha-1}$. The set of linked temporal edges represent the locally consistent path. During teach, this path is marked as privileged.

During an inbound repeat flight, the same visual odometry and map building as teach is performed, however, the experience is saved as non-privileged. In parallel, the system visually localizes to the map of the privileged experience, which provides the error to the privileged path. These localization updates are used for gimbal control in the orientation matching and centroid pointing strategies. Although not demonstrated here, the updates can also be sent to our vision-based path-follower to autonomously retraverse the path.

To facilitate tracking of important vertices and associated transforms in the STPG, a localization chain is used with a ‘tree’ naming convention: leaf ($\mathfrak{l}$), twig ($\mathfrak{w}$), branch ($\mathfrak{b}$), trunk ($\mathfrak{t}$). The leaf (latest live frame) connects to the twig vertex (last successfully localized vertex on the current path) by a temporal transform. The branch is the privileged vertex that was most recently localized against; connected to the twig by a spatial transform. The trunk vertex is the spatially nearest privileged vertex to the leaf frame. Note that the leaf does not necessarily need to be a vertex. Only VO keyframes are saved as vertices in the STPG. With every successful VO update, the estimated transform from the trunk to the leaf, $\check{\mathbf{T}}_{\mathfrak{l},\mathfrak{t}} = \check{\mathbf{T}}_{h,c} = \mathbf{T}_{h,g}\mathbf{T}_{g,d}\mathbf{T}_{d,c}$ in Figure 2.2, is updated in the localization chain. This includes updating the trunk vertex to the privileged vertex that is spatially closest to the new leaf if necessary.

When VO inserts a new vertex into the STPG, visual localization attempts to estimate a spatial transform from the new vertex to its trunk. For example, in Figure 2.2 the new vertex will be H with C as its trunk. The first step is to extract a local window of privileged vertices around the trunk of the new vertex. All 3D landmarks in the local window are transformed into trunk vertex using the privileged temporal transforms in a step termed *landmark migration*. Features are matched from the non-privileged new vertex to the features associated with all migrated landmarks using their SURF descriptors. These raw matches are sent to a Maximum Likelihood Estimation SAmple Consensus (MLESAC) robust estimator to generate a set of localization inlier matches and estimate the spatial transform. Finally, the spatial transform is optimized using the Simultaneous Trajectory Estimation And Mapping (STEAM) engine [Anderson and Barfoot, 2015]. The localization chain is updated with the new spatial transform: $\mathbf{T}_{\mathfrak{w},\mathfrak{b}} \leftarrow \mathbf{T}_{h,c}$.

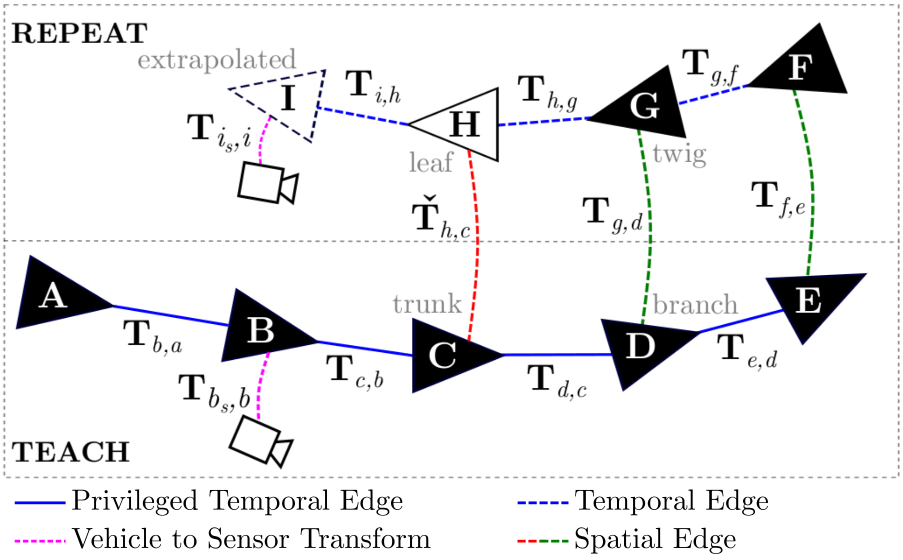


Figure 2.2: Depiction of an STPG with a single privileged experience. Active vision pointing strategies use the transforms from the live (repeat) path to the privileged (teach) path for gimbal control (e.g., $\mathbf{T}_{g,d}$, the 6DoF transformation from vertex D to G).

2.4 Gimbal Control

All active gimbal strategies use a cascaded position-velocity control loop. The outer position loop applies proportional gains to the angular position errors to generate angular rate commands, which are sent to the gimbal's internal controller. Let $\mathbf{\Phi} = [\phi \ \theta \ \psi]^\top$ be the roll, pitch, and yaw joint angles of the gimbal, respectively. The angular velocity commands are computed as

$$\mathbf{u} = \mathbf{K} \left(\mathbf{\Phi}_d - \mathbf{\Phi} \right), \tag{2.1}$$

where $\mathbf{K}$ is a 3×3 diagonal matrix with proportional gains k_ϕ, k_θ, and k_ψ, and $\mathbf{\Phi}_d$ are the desired joint angles. The gimbal controller is run at 10 Hz to match the update rate of the gimbal state and VO. To be clear, the angles provided by the gimbal are with respect to a gravity-aligned frame attached to the vehicle (i.e., the roll and pitch angles are global while the yaw angle is relative to the vehicle heading). Therefore, we use the IMU to obtain the vehicle attitude, which we can then use to recover the gimbal joint angles.

The selected gimbal only allows control of the pitch and yaw axes with rate commands. The roll axis is left to the gimbal to passively stabilize, which promotes consistent tracking of features. Figure 2.3 shows a visual comparison of the pointing strategies we compare in this thesis.

2.4.1 Passive Stabilization

The gimbal used in this work stabilizes all three axes without any user control input (i.e., passively). The roll and pitch axes are globally stabilized in a gravity-aligned inertial frame while the yaw follows the vehicle heading. This off-the-shelf solution, however, is slow to respond to changes in the vehicle heading to promote smooth image motion for filmmaking.

2.4.2 Active Stabilization

To address the yaw following issue, the gimbal can be actively controlled to more closely follow the vehicle heading while stabilizing the pitch. With this strategy, the camera can also be pointed at a non-zero fixed yaw angle relative to the vehicle heading or maintain a global yaw angle. During active stabilization, no information from the vision system is used for gimbal control. It is typically used during our teach phase, but we also test its use in the repeat phase for a full comparison.

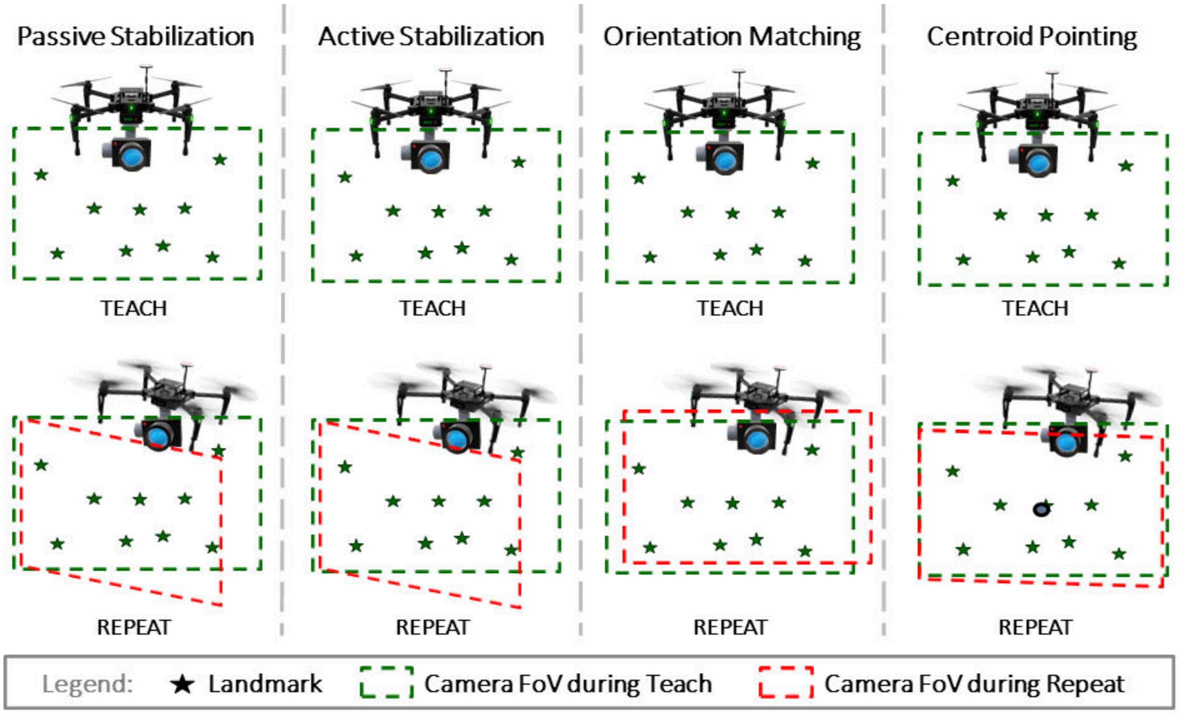


Figure 2.3: The top row shows an example vehicle and camera orientation that occurred during the teach flight with the camera FoV and landmarks observed also shown. The bottom row shows how each of the pointing strategies would respond during the repeat flight when the vehicle orientation is different at a nearby position. The passive and active stabilization strategies simply stabilize the roll and pitch axes in a gravity-aligned frame and align the yaw with the vehicle heading. The orientation matching strategy matches the camera orientation that occured during the teach flight but does not account for path offsets. Our centroid pointing strategy points the camera at the geometric centroid of the previously observed landmarks. In this example, passive and active stabilization result in a landmark falling out of the camera FoV.

2.4.3 Orientation Matching

The goal of orientation matching is to minimize the camera's viewpoint orientation error during repeats. The gimbal yaw and pitch axes are actively controlled to match the camera's recorded orientation at the spatially nearest privileged vertex using the current camera pose estimated by the visual system.

At the beginning of each control step, the localization chain is queried to obtain the latest trunk to leaf transform, $\check{\mathbf{T}}_{\mathfrak{l},\mathfrak{t}}$. To compensate for the gimbal actuation delay, we extrapolate the vehicle pose 200 ms ahead on a trajectory generated by the STEAM engine. We denote the extrapolated pose as $\mathfrak{l}'$ with its associated trunk as $\mathfrak{t}'$ (vertex I and B, respectively, in Figure 2.2). The pose of the camera at $\mathfrak{t}'$ with respect to the pose of the camera at $\mathfrak{l}'$ is given by:

$$\mathbf{T}_{\mathfrak{l}'_s,\mathfrak{t}'_s} = \mathbf{T}_{i_s,b_s} = \mathbf{T}_{i_s,s}\mathbf{T}_{i,h}\check{\mathbf{T}}_{h,c}\mathbf{T}_{c,b}\mathbf{T}_{b_s,b}^{-1}, \tag{2.2}$$

where $\mathbf{T}_{ih}$ is obtained from extrapolation, and $\check{\mathbf{T}}_{h,c} \leftarrow \mathbf{T}_{h,g}\mathbf{T}_{g,d}\mathbf{T}_{d,c}$. Currently a motion model is not used to predict the vehicle-to-sensor transform at I. Instead we set it to the live transform (i.e., $\mathbf{T}_{i_s,i} \leftarrow \mathbf{T}_{h_s,h}$). The camera's viewpoint orientation error is extracted from $\mathbf{T}_{i_s,b_s}$ to compute the desired gimbal angular position, $\mathbf{\Phi}_d$. Figure 2.4 shows an additional visual representation of the relative pose (2.2) that is used for both orientation matching and centroid pointing.

2.4.4 Centroid Pointing

Pointing the camera at the centroid of previously observed 3D landmarks accounts for vehicle path-following errors during repeats. The first two steps in the centroid pointing procedure are submap extraction and landmark migration (similar to visual localization). The STEAM trajectory is queried to obtain the extrapolated vehicle pose with respect to its spatially nearest privileged vertex, $\mathbf{T}_{\mathfrak{l}',\mathfrak{t}'} = \mathbf{T}_{i,b}$. The uncertainty in this transform in the direction of the privileged path is used to extract a window of vertices around vertex $\mathfrak{t}'$ (i.e., a submap denoted as S). The privileged temporal transform between the extrapolated trunk and the next vertex in the privileged path, $\mathbf{T}_{\mathfrak{t}',\mathfrak{n}}$, and the extrapolated trunk to extrapolated leaf, $\mathbf{T}_{\mathfrak{l}',\mathfrak{t}'}$, give the direction along the path expressed in the extrapolated leaf vehicle frame:

$$\hat{\mathbf{u}}_{\mathfrak{l}'}^{\mathfrak{n},\mathfrak{t}'} = \mathbf{C}_{\mathfrak{l}',\mathfrak{t}'}\frac{\mathbf{r}_{\mathfrak{t}'}^{\mathfrak{n},\mathfrak{t}'}}{\left\|\mathbf{r}_{\mathfrak{t}'}^{\mathfrak{n},\mathfrak{t}'}\right\|_2}, \tag{2.3}$$

where $\mathbf{r}_{\mathfrak{t}'}^{\mathfrak{n},\mathfrak{t}'}$ is the position of vertex $\mathfrak{n}$ in $\mathfrak{t}'$, and $\mathbf{C}_{\mathfrak{l}',\mathfrak{t}'}$ is the 3×3 rotation matrix from the extrapolated trunk to extrapolated leaf. Let $\mathbf{\Sigma}_r$ be the 3×3 translational component of the pose covariance matrix $\mathbf{\Sigma}_{\mathfrak{l}',\mathfrak{t}'}$. The uncertainty along the path is given by

$$\boldsymbol{\sigma}_{\hat{\mathbf{u}}} = \sqrt{\hat{\mathbf{u}}_{\mathfrak{l}'}^{\mathfrak{n},\mathfrak{t}'\ \top}\mathbf{\Sigma}_r\hat{\mathbf{u}}_{\mathfrak{l}'}^{\mathfrak{n},\mathfrak{t}'}}. \tag{2.4}$$

This uncertainty is used as a distance criterion for selection of a window of vertices. The maximum window size is restricted to limit the spread of the 3D landmarks used to compute the centroid.

All landmarks in this window are transformed into the sensor frame at the extrapolated trunk vertex, $\mathfrak{t}'_{\mathfrak{s}}$, using the privileged temporal transforms. The centroid of these landmarks is further transformed into the sensor frame at the extrapolated leaf, $\mathfrak{l}'_{\mathfrak{s}}$. Let $\tilde{\mathbf{p}}_j^{\alpha_s}$ be the jth landmark in the sensor frame of vertex $\alpha \in S$ in homogeneous coordinates. Using the extrapolated leaf and trunk vertex in Figure 2.2, the centroid in the sensor frame at the extrapolated leaf in homogeneous coordinates, denoted $\tilde{\mathbf{c}}$ is given by

$$\tilde{\mathbf{c}} = \mathbf{T}_{i_s,b_s} \frac{\sum\limits_{\alpha\in S} \sum_{j=1}^{n_\alpha} \mathbf{T}_{b_s,b}\mathbf{T}_{b\alpha}\mathbf{T}^{-1}_{\alpha_s,\alpha}\mathbf{p}_j^{\alpha_s}}{\sum\limits_{\alpha\in S} n_\alpha}, \tag{2.5}$$

where n_α is the number of landmarks at vertex α, and $\mathbf{T}_{i_s,b_s}$ is computed by (2.2). A spherical wrist model for the gimbal is used to compute the desired gimbal angles $\mathbf{\Phi}_d$ to align the camera's optical axis with the centroid.

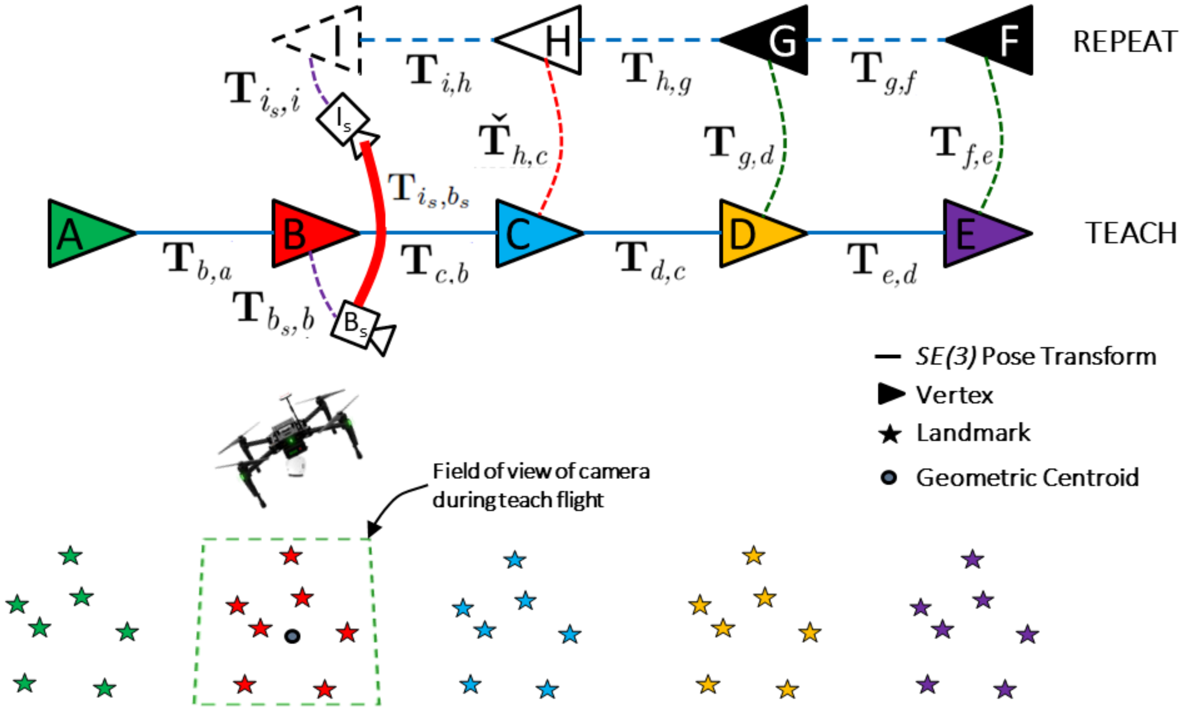


Figure 2.4: In this example, orientation matching uses the relative pose between the predicted and map camera, $\mathbf{T}_{i_s,b_s}$, to match the global camera orientation that occured during the outbound flight at vertex B. Centroid pointing considers the landmarks that were observed at vertex B and possibly a window of vertices surrounding B.

2.5 Experimental Results

We perform multiple outdoor flight tests at the University of Toronto Institute for Aerospace Studies to compare: i) a static and gimballed camera on dynamic and non-dynamic paths, ii) all gimbal pointing strategies in the presence of speed discrepancies, and iii) orientation matching and centroid pointing in the presence of cross-track errors. An example flight path is shown in Figure 2.5. Unless otherwise noted, the camera is pitched down 30 degrees relative to a gravity-aligned inertial frame (or vehicle body frame for the static camera). To perform a proper comparison, we do not use a vision-based path-follower. Instead, we send a GPS waypoint mission to follow the path in the reverse direction. This allows us to directly evaluate the localization performance without adding any coupling effects from control-in-the-loop. Furthermore, it enables experimentation on complicated, dynamic paths to explore failure cases safely.

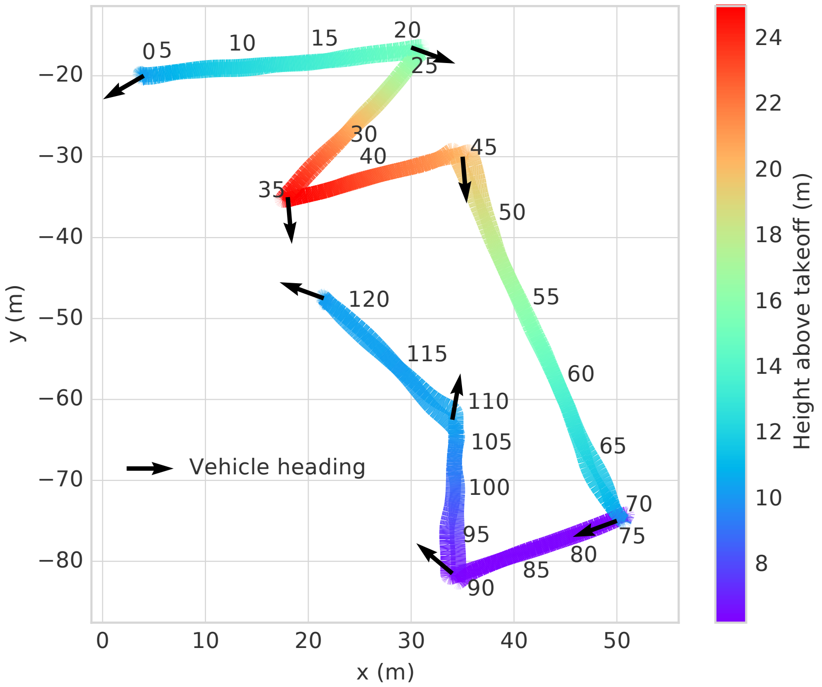


Figure 2.5: The dynamic path used for our gimbal pointing strategy comparisons which shows the height above takeoff, vehicle heading, and location of privileged vertices. Only the heading at each GPS waypoint and every fifth privileged vertex are shown for clarity. The vehicle heading rotates in the shortest direction between waypoints. Note that at the fifth waypoint (privileged vertices 70 to 75 in this example) the altitude changes on-the-spot from 10 m to 6 m (and vice-versa during repeat).

Figure 2.6 shows the hardware setup for the static and gimballed camera systems. We use the DJI Matrice 600 Pro (M600) multirotor UAV with a 3-axis DJI Ronin-MX gimbal. All processing for the VT&R system is performed on-board by an NVIDIA Tegra TX2. A StereoLabs ZED camera is connected to the onboard computer to provide 672 × 376 greyscale stereo images. A 900 MHz XBee low-bandwidth, long-range radio communication link is used to send high-level mission commands to the onboard computer. These high-level mission commands include manually triggering state transitions and sending GPS waypoint missions to the flight controller. The gimbal connects to the flight controller to accept control commands and feedback angular positions. The M600's flight controller communicates with the onboard computer via Robot Operating System (ROS).

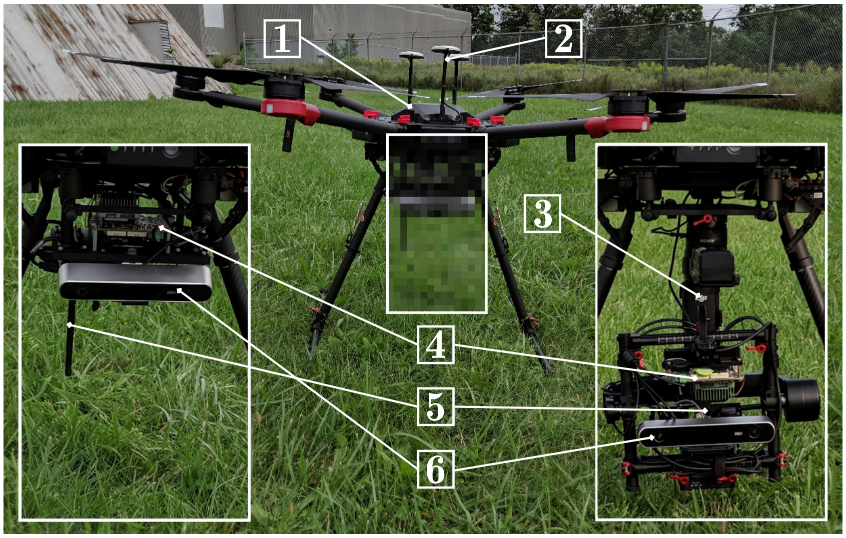


Figure 2.6: The hardware setup with a static camera (left) and gimballed camera (right) on a multirotor UAV: (1) DJI Matrice 600 Pro, (2) DJI A3 GPS module, (3) DJI Ronin-MX 3-axis gimbal, (4) NVIDIA Tegra TX2, (5) XBee Pro 900 MHz XSC S3B RF module, (6) StereoLabs ZED camera.

2.5.1 Gimballed Camera Robustness

The performance of a static and gimballed camera on a simple 315 m path at 15 m altitude taught at 3 m/s and returned at 9 m/s is shown in Figure 2.7. One interesting outcome is a higher maximum number of inliers for the static camera system. This can be attributed to inaccuracies and latency in the gimbal angular positions indicating more careful calibration is required. However, the inconsistency of a static camera due to large perspective shifts is clearly shown by the variance in the inliers.

On more dynamic paths, the large perspective shifts undergone by a static camera result in localization failures. Figure 2.8 shows a zigzag pattern flight path highlighted

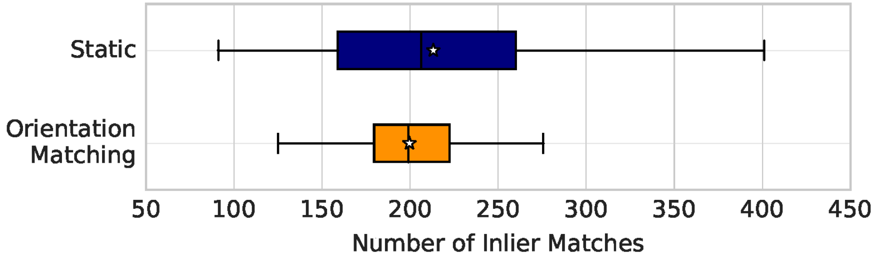


Figure 2.7: For a simple path with few dynamic motions, a static camera localizes even when returning at a faster speed (from 3 m/s outbound to 9 m/s target inbound speed). However, a gimbal reduces the variance in localization inliers by maintaining similar perspective.

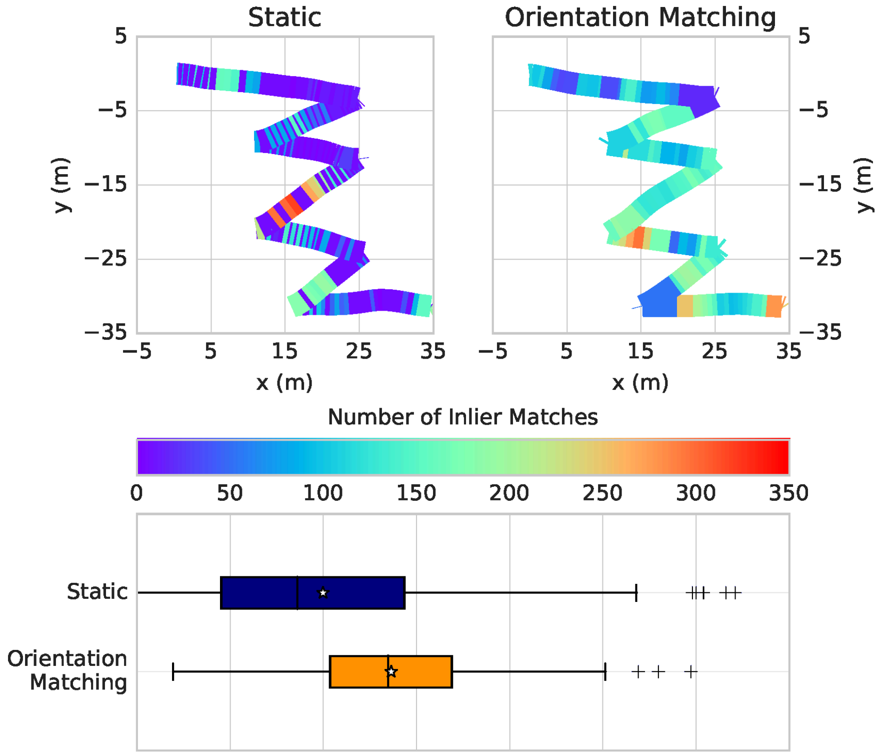
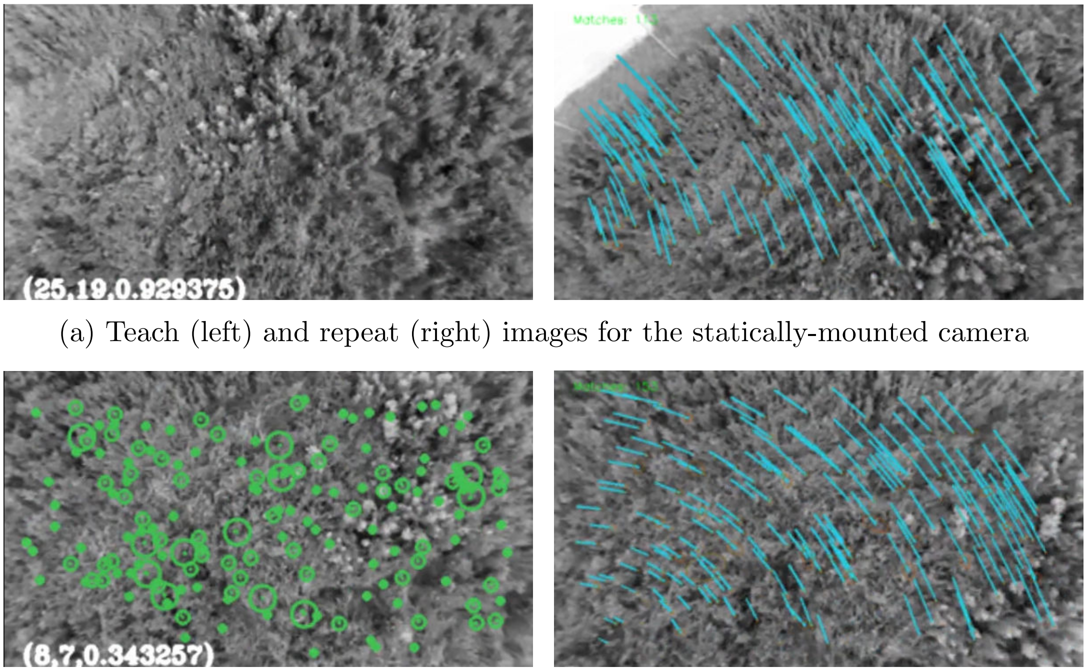


Figure 2.8: For highly dynamic paths, the static camera system has trouble localizing due to large viewpoint changes. The active gimballed camera system minimizes perspective error during the dynamic motions to improve localization performance with a 37% increase in the mean number of inliers.

with the average number of localization inliers at each position over two runs. The path is 115 m in length with 130 degree rotations in the vehicle heading between waypoints. It was flown at a height of 7 m above ground with the camera pitched down 80 degrees to promote the adverse effects of viewpoint orientation error. Even for 3 m/s flights, a static camera frequently fails to localize since small perspective errors result in a large reduction in image overlap on this path. The gimbal enables successful localizations by attenuating viewpoint orientation errors. Figure 2.9 shows an example of the static camera failure at one of the corners. The gimbal with active camera pointing increases the mean number of inliers by 37% over a static camera.



(a) Teach (left) and repeat (right) images for the statically-mounted camera

(b) Teach (left) and repeat (right) images for the gimballed camera

Figure 2.9: At one of the corners in the zig-zag pattern path, the vehicle undergoes a large roll and pitch to quickly change directions. Since the statically-mounted camera is coupled to the vehicle's orientation, the viewpoint changes drastically between the outbound and return flights as shown by the fence and road becoming visible in the top left of the return image. Our gimballed camera with an orientation matching strategy reduces the perspective error, which allows it to localize at the same corner. The return images show the SURF feature tracks while the outbound images show the localization inliers.

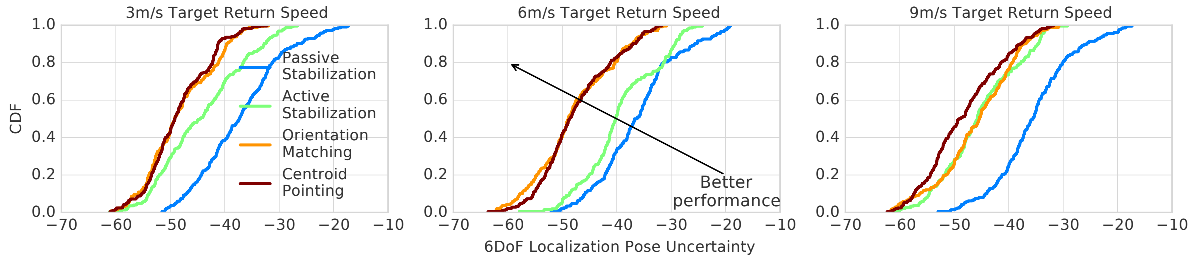


Figure 2.10: The CDF of the localization pose uncertainties show that active pointing strategies result in more confident localization estimates with centroid pointing showing slightly more confidence than orientation matching. The localization uncertainty is computed as the log determinant of the 6DoF spatial pose uncertainty matrix for each localization update (i.e., $\mathbf{\Sigma}_{\mathrm{l,t}}$).

2.5.2 Handling Velocity Discrepancies

In this experiment, we evaluate the localization performance of passive and active gimballed strategies with increasing return velocities from 3 m/s to 9 m/s with all teach flights conducted at 3 m/s. Figure 2.5 shows the altitude-varying 170 m flight path used for these tests. The CDF of the localization uncertainties is shown in Figure 2.10 while Figure 2.11 summarizes the localization inliers for each strategy over two runs. Active pointing strategies are able to handle increasing speed discrepancies as they show only a small drop in inliers with no failures. Pointing strategies with vision-in-the-loop (i.e., orientation matching and centroid pointing) result in the highest number of inliers and the greatest localization confidence as expected. Off-the-shelf passive stabilization actually causes localization failures when there are speed discrepancies between flights, which demonstrates the necessity of active pointing to add visual localization robustness on UAVs.

Camera perspective errors that result from different vehicle orientations at matching positions along the teach and repeat paths can be reduced using active pointing strategies. Figure 2.12 shows the vehicle and camera orientation errors grouped as a pair for each pointing strategy and across different return velocities. Each pair of orientation errors is obtained using the vehicle and camera spatial localization transforms (e.g., $\mathbf{T}_{g,d}$ and $\mathbf{T}_{g_s,d_s}$ in Figure 2.2). As the velocity discrepancy between flights increases, the vehicle orientation error also increases as expected. Passive stabilization actually increases the camera viewpoint orientation error due to lag in following the vehicle heading. Since the vehicle heading rotates in opposite directions in the teach and reverse repeat flights, the lag results in an increase in the camera orientation error on the yaw axis. This effect is more pronounced with larger velocity discrepancies resulting in localization failures as shown in Figure 2.13. Active stabilization removes the yaw lag but does not account for vehicle yaw error between teach and repeat runs as it only follows the current vehicle heading. However, the act of stabilizing the roll and pitch to reduces the camera error. Both active strategies with vision-in-the-loop provide the greatest reduction in perspective error. Centroid pointing does not directly attempt to minimize the perspective error but provides an improvement by pointing at previously observed landmarks. Orientation matching clearly performs its duty by minimizing the perspective error the most. It provides a 65%, 58%, and 54% reduction in the root-mean-square orientation error from the vehicle to camera frame for 3 m/s, 6 m/s, and 9 m/s return flights, respectively.

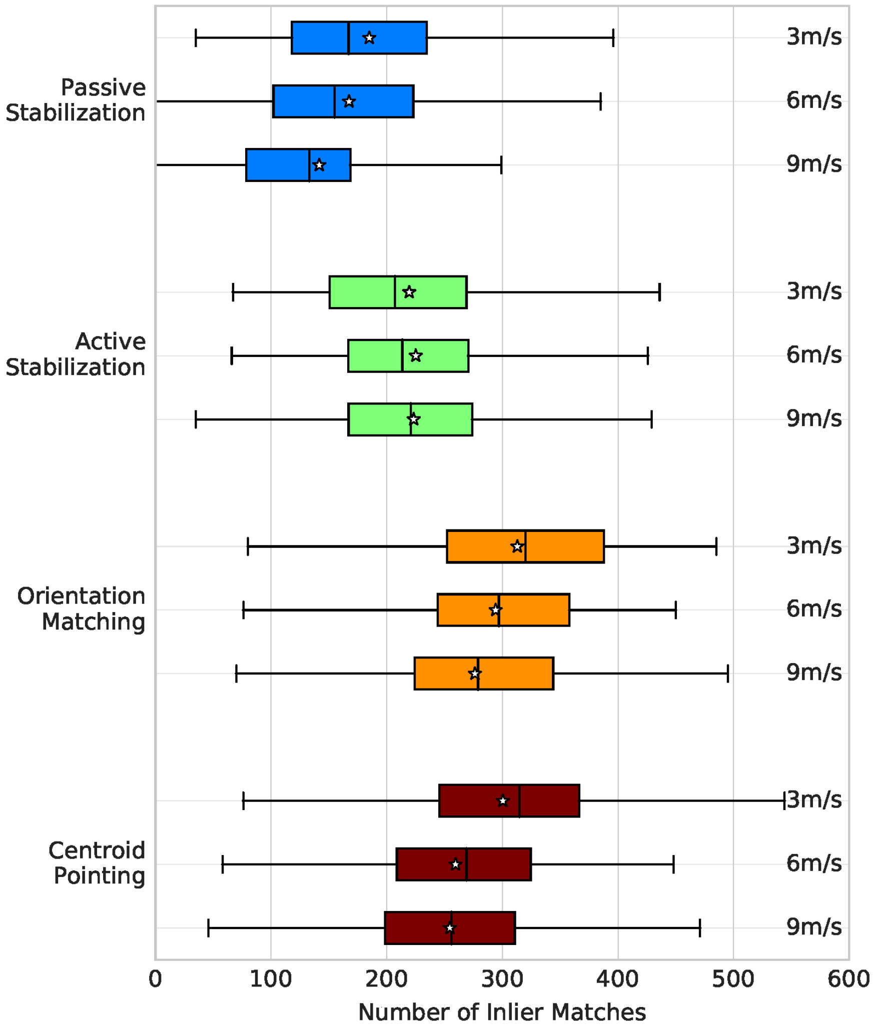


Figure 2.11: Active gimbal control prevents localization failures despite increasing speed discrepancies between teach and repeat flights. Incorporating visual information in the pointing strategy results in better localization performance.

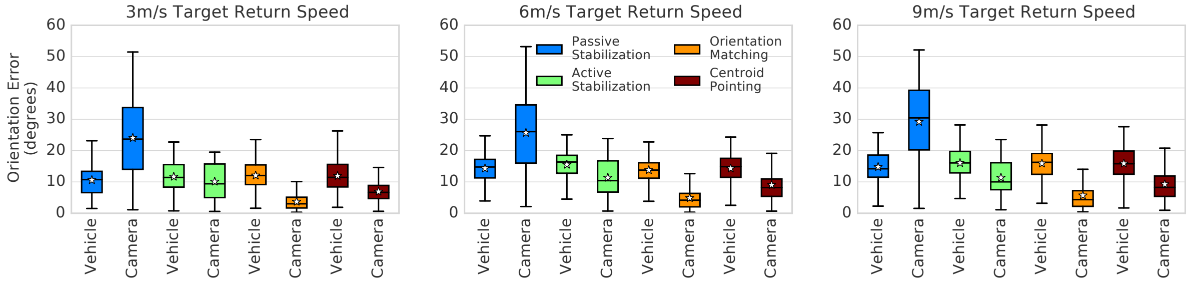


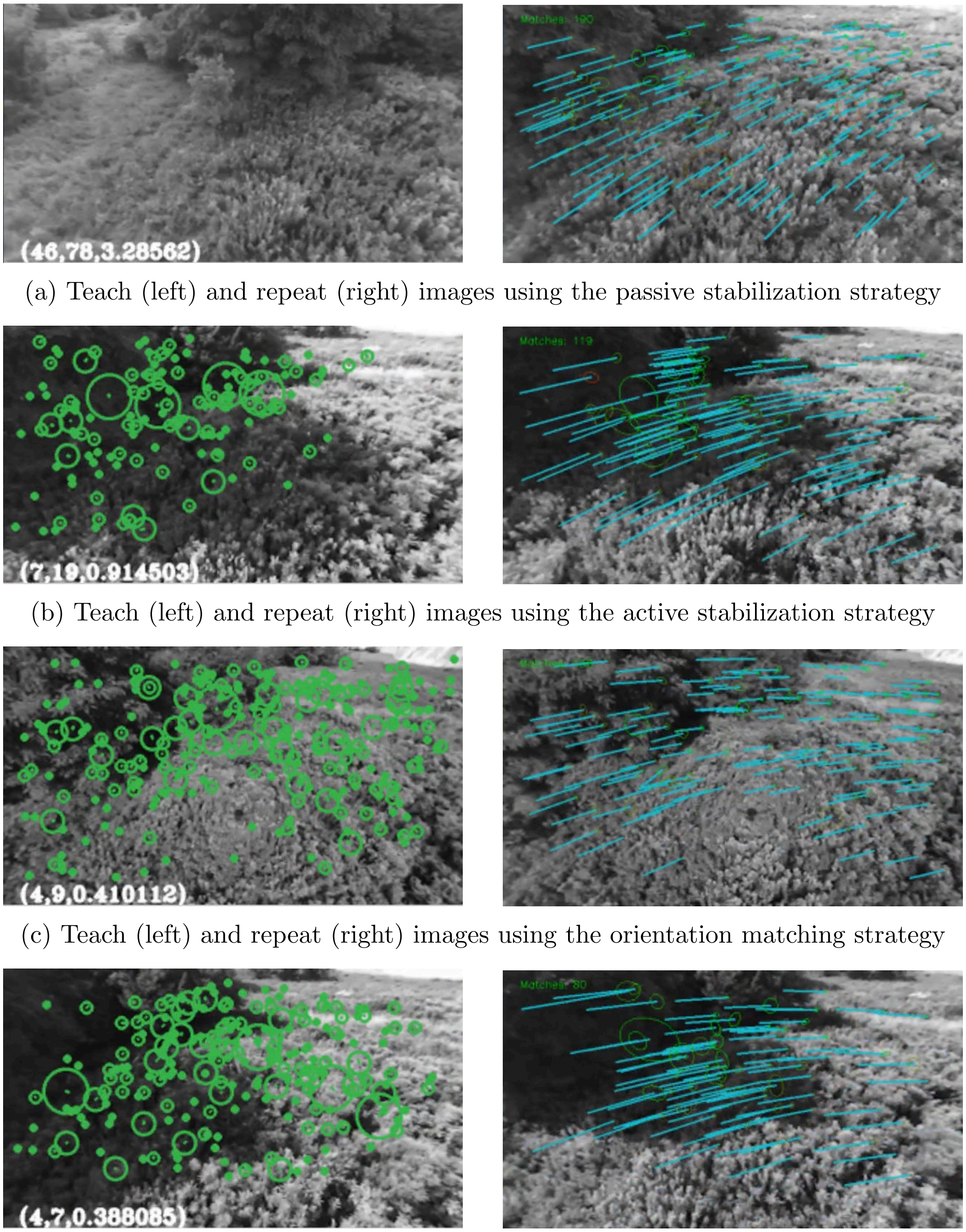
Figure 2.12: A gimballed camera with active pointing strategies reduces the camera perspective error that result from vehicle orientation errors between teach and repeats flights. The simple act of adding a gimbal is not enough as we see an off-the-shelf passive stabilization strategy actually increases the camera perspective error. Centroid pointing does not attempt to minimize viewpoint orientation error but it is compared for completeness.

2.5.3 Handling Path-following Errors

In this experiment, we intentionally add path offsets to the repeat path to further evaluate the localization performance of orientation matching and centroid pointing (see Figure 2.14). Intuitively, a centroid pointing strategy is more suitable for situations with large path-following errors since it attempts to compensate for the translation errors.

On segment 1, the vehicle descends from 10 m to 6 m altitude with lateral offsets up to 6.5 m. Since the scene structure is spatially close to the camera along this segment, the 6.5 m lateral offset creates perspective errors that orientation viewpoint manipulation alone cannot compensate. Landmarks simply fall out of view when matching orientations. With centroid pointing, the angle at which they are viewed dramatically changes resulting in difficulty with SURF feature matching. Along segment 2, the vehicle undergoes a pure vertical offset: climbing from 6 m to 10 m altitude. Segment 3 contains growing lateral and vertical offsets finishing with a −5 m altitude offset when it rejoins the original path. Segment 4 contains an 8 m lateral offset at 25 m altitude while segment 5 contains pure lateral offsets. Along segments 4 and 5, the scene structure is far enough away from the camera that both strategies can easily compensate for the large position offsets. Along segment 2 and parts of 3, the position offsets are large enough to reduce landmark visibility when orientation matching but small enough to be compensated by centroid pointing. Figure 2.15 shows an example of the viewpoints of both strategies during an altitude offset along segment 2. Landmarks in the bottom half of the map image are not present in the orientation matching view causing localization to be difficult. The same landmarks are visible in the centroid pointing view.

Although centroid pointing shows a slight performance benefit along certain segments



(a) Teach (left) and repeat (right) images using the passive stabilization strategy

(b) Teach (left) and repeat (right) images using the active stabilization strategy

(c) Teach (left) and repeat (right) images using the orientation matching strategy

(d) Teach (left) and repeat (right) images using the centroid pointing strategy

Figure 2.13: A comparison of the camera perspective and localization performance at the same position for each pointing strategy during 9 m/s returns. The repeat images show the SURF feature tracks while the teach images show the localization inliers. Passive stabilization has a slow yaw alignment resulting in it viewing the left side of the group of trees during the outbound flight and the right side during the return flight. This ultimately leads to a localization failure. All active strategies result in a similar camera viewpoint in this example and successfully localize.

of the path, it is important to note that the overall performance of both strategies is comparable. We aim to explore dynamic selection of pointing strategies during flight in future work. Orientation matching can be used when closely following the path while centroid pointing can be employed when the path offset is large enough to cause a substantial number of landmarks to fall out of the field of view.

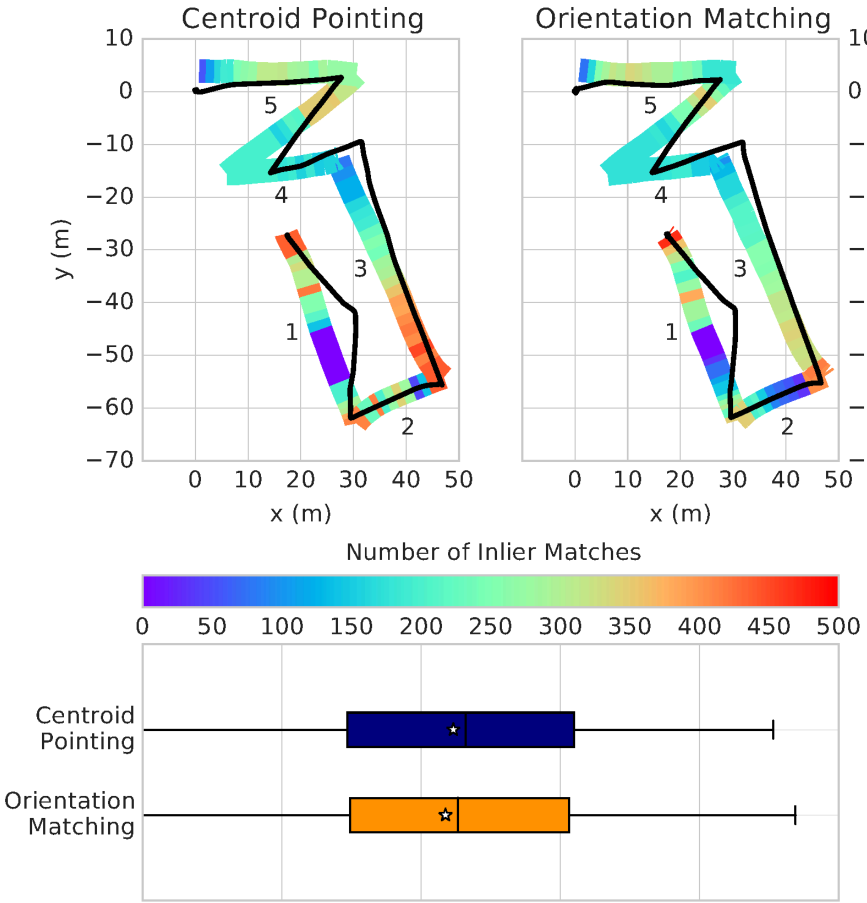
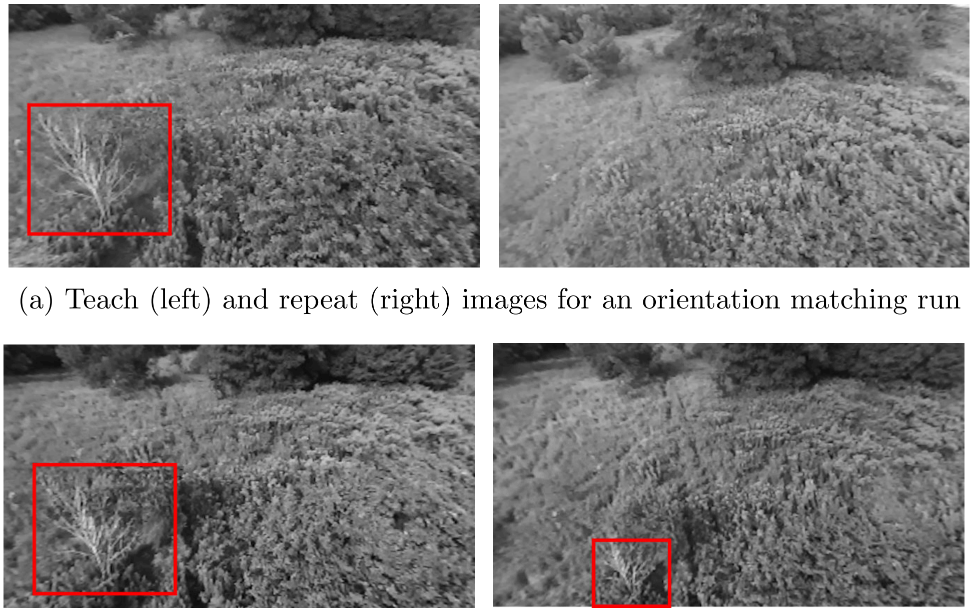


Figure 2.14: While the overall performance is similar, centroid pointing shows an advantage along segment 2 and parts of 3. The black line shows the outbound teach path while the inlier highlights are centered on the inbound repeat path.



(a) Teach (left) and repeat (right) images for an orientation matching run

(b) Teach (left) and repeat (right) images for a centroid pointing run

Figure 2.15: Comparison of orientation matching and centroid pointing return views (right) on segment 2. The spatially nearest images captured during the teach runs (left) are used for localization. The landmarks along the bottom of the image, such as the outlined shrubbery, are missing from the orientation matching view due to the altitude offset. Centroid pointing keeps the landmarks in the field of view.

2.6 Summary

We demonstrated improved visual localization performance using an active gimbal-stabilized camera within a VT&R framework on a multirotor UAV. We experimentally showed the need for a gimballed camera over a traditional statically-mounted camera. Multiple gimbal pointing strategies were evaulated including off-the-shelf passive stabilization, active stabilization, and two active strategies that use visual information to minimize the camera viewpoint orientation error (orientation matching) and point at the centroid of previously observed landmarks (centroid pointing). We showed that a passively stabilized gimbal can actually lead to localization failures. Finally, we demonstrated the ability of orientation matching and centroid pointing to enable visual localization despite velocity discrepancies and large path-following errors between the teach and repeat flights.

Chapter 3

Visual Localization with Google Earth Images

3.1 Motivation

Vision-based techniques involving VO are the most popular for Unmanned Aerial Vehicle (UAV) navigation in GPS-denied environments. However, pure odometry techniques are unreliable for accurate pose estimates since they drift over time in the absence of corrections. Visual SLAM corrects these drifts through loop closure and has been sucessfully demonstrated in GPS-denied environments [Blösch et al., 2010, Weiss et al., 2013, Shen et al., 2015] but requires revisiting locations. On the other hand, VT&R [Furgale and Barfoot, 2010] can enable safe navigation in the absence of GPS without requiring globally accurate poses but is limited to navigation along previously traversed routes. Such a technique is suitable to perform emergency return of UAVs in the event of GPS loss [Warren et al., 2019].

The aforementioned techniques require the vehicle itself to map an area either through a human-operated manual teach phase in the case of VT&R or a carefully developed safe exploration algorithm for autonomous SLAM. However, a 3D reconstruction of many parts of the world is already available in GE. The ability to use this 3D reconstruction as a map would enable global pose estimation without GPS, having to worry about safe exploration, or restricting navigation to a previously traversed route. One of the main challenges to using this map is the large appearance difference between the 3D reconstruction and the true world: lighting and seasonal changes, as well as recent structural changes to the environment all present difficulties for visual localization.

In this work, we present a technique to determine the full six DoF global pose of

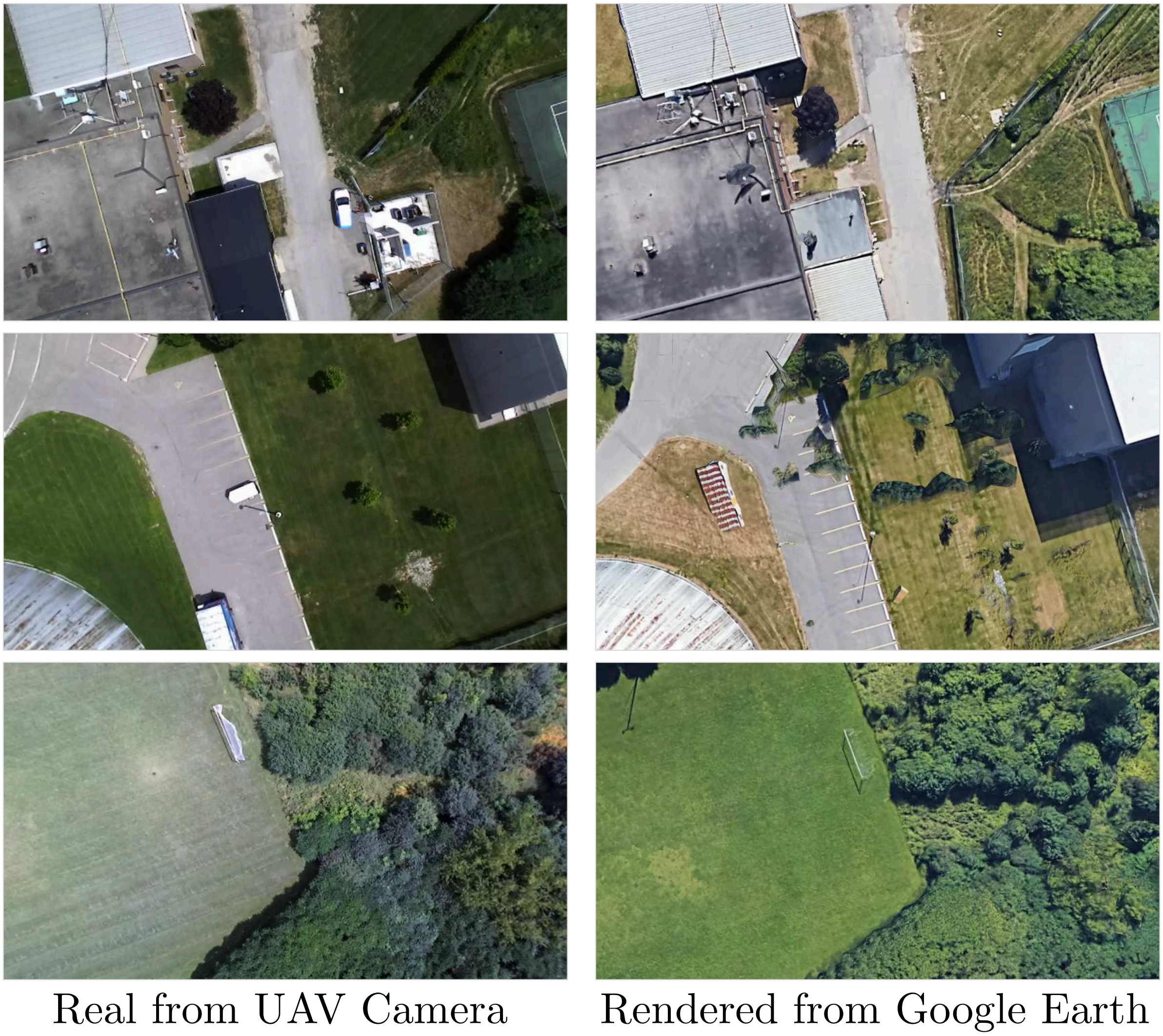


Figure 3.1: Comparison of real-world UAV images and rendered Google Earth images taken from the approximately same viewpoint at three locations along one of the flights. Large appearance changes, especially with vegetation, impermanent objects such as cars, poor 3D reconstructions (e.g., trees in middle pair), and structural changes to buildings (e.g., top pair) can all cause difficulties for visual localization.

a UAV in an area where the UAV itself has not mapped by using only a gimballed stereo camera, IMU, and georeferenced images from GE. These geoferenced images are rendered before the flight and stored onboard the UAV to enable navigation within the region covered by the images. The only limitations to the map size are the extents of the reconstruction coverage area and the available onboard storage.

Visual localization of the real images with the rendered images using traditional sparse features (e.g., SURF) is challenging due to the large appearance difference mentioned previously (see Figure 3.1). Therefore, we perform image registration using a dense technique that relies on MI. MI provides robustness to appearance changes allowing us to accurately register the images. We optimize the MI over warping parameters to align the real and rendered images. The result from this image registration is then fused with a pose estimated by a gimballed VO pipeline. The performance of this technique is evaluated on multiple datasets collected at the UTIAS.

The contribution of this work is a method to accurately estimate the global pose of a UAV in GPS-denied environments using pre-rendered images from a 3D reconstruction of the Earth. Our method allows accurate estimation at lower altitude flights compared to similar previous work described below. We also demonstrate robust estimation over an entire day in the presence of signficant lighting changes incurred from sunrise to sunset on 6.8 km of real-world data.

3.2 Related Work

Some of the earliest work using georeferenced satellite images used edges for registration. However, a simple edge detector resulted in only two successful matches along a 1 km trajectory [Conte and Doherty, 2008]. Building outlines extracted using local features were more successful in estimating the 6DoF pose of an aerial vehicle [Son et al., 2009]. Unfortunately, this technique cannot be employed for lower altitude flights where the outlines of multiple buildings are not visible in a single image.

Some recent work using local image features use street view images to estimate the pose of a ground robot [Agarwal and Spinello, 2015] and a UAV [Majdik et al., 2015]. In both cases, techniques similar to bag-of-words are first used for place recognition followed by image registration using SIFT keypoints in the matched images. However, even after finding the best matching georeferenced image, the feature matching can contain 80% outliers [Majdik et al., 2015] due to the large image appearance and viewpoint differences which makes it difficult to accurately localize.

Unsurprisingly, Convolutional Neural Networks (CNNs) have seen increased usage in recent years as image descriptors due to their ability to learn generic features that can be applied to a variety of tasks such as image classification [Krizhevsky et al., 2012, Simonyan and Zisserman, 2014] and object detection [Redmon et al., 2016]. Features from the middle layers of the CNNs have been shown to be robust against appearance changes while features from the final layers are robust to viewpoint changes and provide more semantic information about the structure of the scene [Sünderhauf et al., 2015]. Often pretrained CNNs are further trained for the task of place recognition allowing topological localization [Lin et al., 2015, Kim and Walter, 2017, Shetty and Gao, 2019] followed by filtering with VO in a particle filter [Kim and Walter, 2017] or Kalman filter [Shetty and Gao, 2019]. These whole-image descriptors only allow finding an image match and do not provide metric information about the relative pose between the query and map image. Often the pose of the matching map image is taken as the best estimate which ultimately limits the accuracy of the localizations to the spatial resolution of the georeferenced images.

We are interested in accurate metric localization using georeferenced images. To accomplish this, we use a dense image registration technique to align images captured by a camera mounted on the UAV with pre-rendered georeferenced images. Instead of minimizing the photometric error, we use a metric computed using MI to add robustness to appearance changes.

We adopt the use of the Normalized Information Distance (NID) [Pascoe et al., 2015,] which is computed from MI (3.13). The NID is a value between 0 and 1 that is not as dependent on the amount of information content in the images (i.e., the amount of image overlap) as MI. It has been shown to be able to robustly register images [Pascoe et al., 2015], and localize a ground vehicle equipped with a monocular camera using a textured 3D map generated from a LIDAR and camera [Pascoe et al., 2015]. One of the reasons for the high accuracy in [Pascoe et al., 2015] is their ability to generate synthetic images online from the textured 3D map allowing direct optimization over the $SE(3)$ pose parameters. GE has no online 3D view API, so we are required to semi-manually pre-render images at a limited number of poses before the flight. We then perform a warping online for interpolation.

Similar to our work is [Yol et al., 2014], which determines the global position and heading of a UAV by finding the optimal scale-rotation-translation (sRt) warping (3.11) that maximizes the MI of a query image taken by a nadir-pointed camera warped into a mosaic of satellite images. An sRt warping is a 4DoF image warping that performs a scaling (zoom), 1D rotation, and 2D translation. It assumes the scene is planar and parallel to the image plane. For a nadir-pointed camera this assumption becomes more

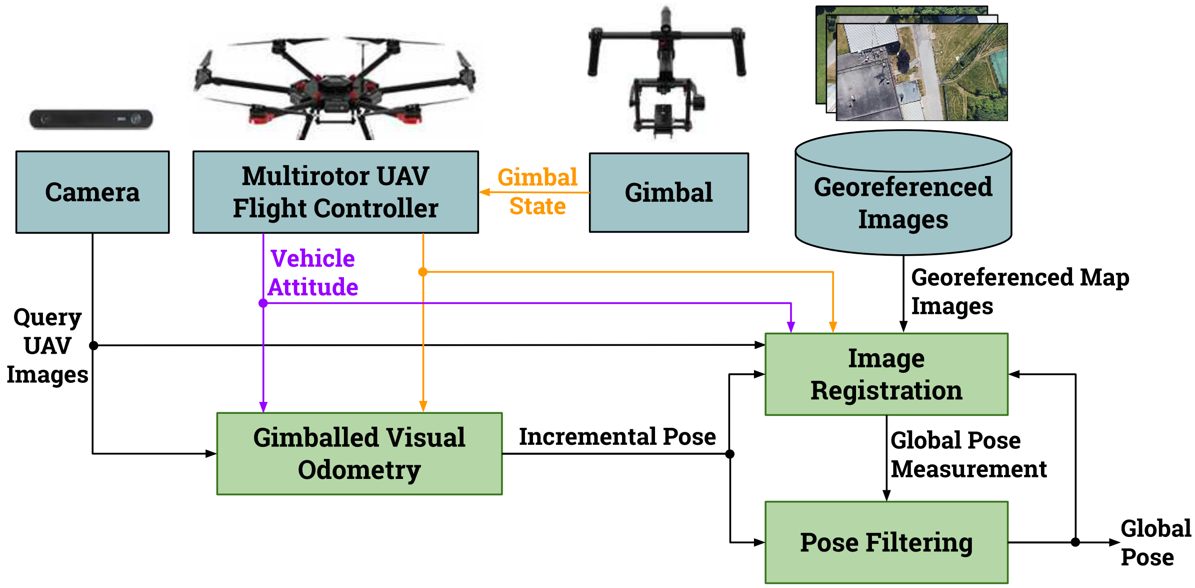


Figure 3.2: Gimballed VO is applied on the sequence of UAV images to obtain incremental pose estimates. For each keyframe, the associated query image is registered with a selected GE map image. The poses estimated by the image registrations are treated as measurements to apply corrections to VO in a filtering framework.

valid at higher altitudes since the building heights become small relative to the distance to the camera. In contrast to [Yol et al., 2014], we conduct lower altitude flights (e.g., 36 m Above Ground Level (AGL) compared to 150 m) where the scene is often non-planar. Despite this, we are able to use this warping due to our method of rendering images at mulitple nearby poses in the 3D reconstruction.

3.3 Pose Estimation

We estimate the global 6DoF $SE(3)$ pose of a multirotor UAV using only a gimballed stereo camera, an IMU (for vehicle attitude only), and a set of geoferenced GE images. Let

$$\mathbf{T}_{W,k} = \begin{bmatrix} \mathbf{C}_{W,k} & \mathbf{r}_W^{k,W} \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{3.1}$$

be the transformation from the vehicle at keyframe k to a world East-North-Up (ENU) frame. The position of the vehicle in the ENU frame is given by $\mathbf{r}_W^{k,W} = [x_W^{k,W} \; y_W^{k,W} \; z_W^{k,W}]^\top$ and the roll, pitch, and yaw ($\phi_{W,k}$, $\theta_{W,k}$, $\psi_{W,k}$, respectively) can be extracted from the 3×3 rotation matrix $\mathbf{C}_{W,k}$. Let $\mathcal{I}^q = (\mathbf{I}_1^q, \mathbf{I}_2^q, \ldots, \mathbf{I}_K^q)$ be the sequence of real UAV query images from each keyframe. We attempt to localize each keyframe image using a set of geoferenced map images, $\mathcal{I}^m = \{\mathbf{I}_1^m, \mathbf{I}_2^m, \ldots, \mathbf{I}_N^m\}$, where the global pose of map image n

is denoted $\mathbf{T}_{W,n_s}$ with s indicating the sensor (camera) frame. Figure 3.2 provides an overview of the estimation pipeline.

3.3.1 Gimballed Visual Odometry

The first step in the estimation pipeline is to perform VO on the UAV images. VO is performed using the VT&R 2.0 software system adapted for use on UAVs with gimballed cameras [Warren et al., 2019].

The inputs are rectified stereo greyscale images and a non-static vehicle-to-sensor transform, $\mathbf{T}_{f_s,f}$, computed at 10 Hz for each frame. It is computed by compounding transformations using the three gimbal angles and known translations between joints followed by a rotation into the standard camera frame. The roll and pitch axes of the gimbal are globally stabilized in a gravity-aligned inertial frame while the yaw follows the vehicle heading.

For each input stereo image pair, SURF features are extracted and descriptors generated. For each image we have SURF keypoints $\{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_N\}$ with associated descriptors $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ where $\mathbf{d}_i \in \mathbb{R}^{64}$, and $\mathbf{k}_i = (u, y)$ are image plane coordinates with uncertainty $\mathbf{R}_i$:

$$\mathbf{k}_i = \bar{\mathbf{k}}_i + \delta\mathbf{k}_i, \quad \delta\mathbf{k}_i \sim \mathcal{N}(0, \mathbf{R}_i). \tag{3.2}$$

Triangulation

Stereo triangulation is performed by matching keypoints in the left and right stereo image pair via their descriptors. From the matches we can triangulate the corresponding 3D landmark position, $\mathbf{p}_i = [x \ y \ z]^\top$. Stereo triangulation is unable to accurately triangulate landmarks at extreme depths when the baseline is small. Therefore, in our pipeline, landmarks are also triangulated from motion by matching keypoints between successive frames. This allows varied altitude flights as the system can rely more on these sequentially triangulated landmarks when flying at high altitudes and stereo triangulated landmarks are lower altitudes.

RANSAC

To estimate incremental motion, each feature is matched to the last keyframe via SURF descriptor matching. The raw matches and associated landmarks $\{\mathbf{y}_i, \mathbf{p}_i\}$ are sent through a 4-point RANSAC estimator to determine the relative transform between the current frame and last keyframe's pose in the sensor frames, $\mathbf{T}_{f_s,k_s}$. RANSAC samples 4 point correspondances and solves the P3P problem using 3 points with the 4th point used to select an appropriate solution. The reprojection errors along with a robust cost function are used to assess the solution.

A 3D landmark can be projected into pixel coordinates using the pinhole projection camera model:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{g}(\mathbf{p}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{3.3}$$

The reprojection errors are computed by transforming the landmarks into the current frame, projecting into image coordinates, and comparing to the measured image coordinates:

$$\mathbf{e}_i(\mathbf{T}_{f_s,k_s}) = \mathbf{y}_i - \mathbf{g}(\mathbf{C}_{f,k}\mathbf{p}_i + \mathbf{r}_f^{k,f}). \tag{3.4}$$

The weighted reprojection error for the ith raw match is given by:

$$u_i(\mathbf{T}_{f,k}) = \sqrt{\mathbf{e}_i(\mathbf{T}_{f,k})^\top \mathbf{R}_i^{-1} \mathbf{e}_i(\mathbf{T}_{f,k})}. \tag{3.5}$$

A raw match is an inlier if the weighted reprojection error is below a threshold $u_i < u_{thres}$. Furthermore, the overall cost of the solution is computed using the weighted reprojection errors from all matches as

$$J(\mathbf{T}_{f,k}) = \sum_{i=1}^{N} \sigma_i \rho(u_i(\mathbf{T}_{f,k})), \tag{3.6}$$

where σ_i is a scaling term and $\rho(x)$ is the Geman-McClure robust cost function:

$$\rho(x) = \frac{1}{2}\frac{x^2}{1+x^2}. \tag{3.7}$$

The cost and number of inliers are used to keep track of the best solution. The RANSAC algorithm finishes early if the number of inliers exceeds a desired threshold (e.g., 400 inliers).

The relative transform is then optimized in a nonlinear refinement step using the STEAM engine [Anderson and Barfoot, 2015] to minimize the weighted reprojection errors, and using a constant velocity model to smooth out the pose estimates.

The estimated incremental pose is still in the sensor frames thus the vehicle-to-sensor transform at the keyframe and current frame are used to recover the incremental pose estimate in the vehicle frame:

$$\mathbf{T}_{f,k} = \mathbf{T}_{f_s,f}^{-1} \mathbf{T}_{f_s,k_s} \mathbf{T}_{k_s,k}. \tag{3.8}$$

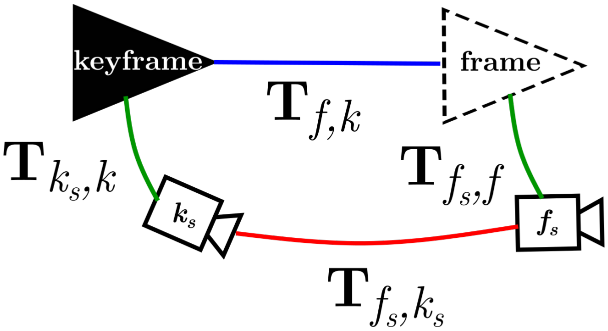


Figure 3.3: Gimballed VO uses the non-static vehicle-to-sensor transform to recover the incremental vehicle transformation from the estimated camera motion.

Keyframe Windowed Refinement

If the translation or rotation exceeds a threshold or the number of inliers drops below a minimum amount, a new keyframe is added. For each keyframe, all of the features, new landmarks that were observed, and the vehicle-to-sensor transform are stored as a vertex in a pose graph. The relative transform is stored as the edge between vertices in this pose graph. When a new vertex is inserted into the graph, windowed bundle adjustment is performed on the last 5 keyframes using a similar cost function as before but now adjusting landmark positions in addition to the relative poses.

The end result of gimballed VO is a set of linked dead-reckoned poses that represent the traversed path. As with all odometry techniques, incremental errors build up to cause drift in the estimated position and orientation that must be corrected with some other measurement.

3.3.2 Image Registration

For every keyframe image, $\mathbf{I}_k^q$, the goal is to determine the relative $SE(3)$ pose between the query camera at k and a virtual GE camera that generated image n, $\mathbf{T}_{k_s,n_s}$. The global pose measurement of the vehicle is then obtained from

$$\mathbf{T}_{W,k} = \mathbf{T}_{W,n_s}\mathbf{T}_{k_s,n_s}^{-1}\mathbf{T}_{k_s,k}. \tag{3.9}$$

In this work, all real and rendered images are taken with the camera pointed in the nadir direction. The relative roll, ϕ_{k_s,n_s}, and pitch θ_{k_s,n_s}, are obtained from our gimbal, which keeps these angles at approximately 0 degrees. Therefore, we only need to estimate four pose parameters:

$$\boldsymbol{\eta} = [x_{k_s}^{n_s,k_s} \;\; y_{k_s}^{n_s,k_s} \;\; z_{k_s}^{n_s,k_s} \;\; \psi_{k_s,n_s}]^\top. \tag{3.10}$$

Since the image registration is estimating 4DoF, we use an sRt warping instead of a full homography:

$$\mathbf{x}' = w(\mathbf{x}, \boldsymbol{\mu}) = s\mathbf{R}(\psi) + \mathbf{t}, \tag{3.11}$$

where $\mathbf{x} = [x \ y]^\top$ is the query image plane coordinate warped to $\mathbf{x}' = [x' \ y']^\top$ for the map image, s is a scale, $\mathbf{R}(\psi)$ is a 1D rotation, $\mathbf{t} = [t_x \ t_y]^\top$ is a 2D translation, and $\boldsymbol{\mu} = [s \ \psi \ t_x \ t_y]^\top$. We can also directly warp pixel coordinates $\mathbf{u} = [u \ v]^\top$ from the query image into map image pixel coordinates $\mathbf{u}' = [u' \ v']^\top$:

$$\bar{\mathbf{u}}' = w(\bar{\mathbf{u}}, \boldsymbol{\mu}) = \mathbf{K}' \begin{bmatrix} s\mathbf{R}(\psi) & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{K}\bar{\mathbf{u}}, \tag{3.12}$$

where $\bar{\mathbf{u}} = [u \ v \ 1]^\top$ and K is the camera intrinsics matrix.

The NID between a query image and warped map image is

$$NID(\mathbf{I}_k^q, \mathbf{I}_n^m, \boldsymbol{\mu}) = \frac{H(\mathbf{I}_k^q, \mathbf{I}_n^m, \boldsymbol{\mu}) - MI(\mathbf{I}_k^q; \mathbf{I}_n^m, \boldsymbol{\mu})}{H(\mathbf{I}_k^q, \mathbf{I}_n^m, \boldsymbol{\mu})}, \tag{3.13}$$

where the MI is

$$MI(\mathbf{I}_k^q; \mathbf{I}_n^m, \boldsymbol{\mu}) = H(\mathbf{I}_k^q) + H(\mathbf{I}_n^m, \boldsymbol{\mu}) - H(\mathbf{I}_k^q, \mathbf{I}_n^m, \boldsymbol{\mu}). \tag{3.14}$$

The joint entropy is given by

$$H(\mathbf{I}_k^q, \mathbf{I}_n^m, \boldsymbol{\mu}) = -\sum_{a=1}^{N}\sum_{b=1}^{N} p_{qm}(a, b, \boldsymbol{\mu}) \ln(p_{qm}(a, b, \boldsymbol{\mu})), \tag{3.15}$$

where $p_{qm}(a, b, \boldsymbol{\mu})$ is the joint probability distribution of image intensities in $\mathbf{I}_k^q$ and $\mathbf{I}_n^m$ for N bins with bin indices a and b. An example of the joint probability distribution for two artificial images is shown in Figure 3.4e. Similarly, the individual entropies are

$$H(\mathbf{I}_k^q) = -\sum_{a=1}^{N} p_q(a) \ln(p_q(a)), \tag{3.16}$$

$$H(\mathbf{I}_n^m, \boldsymbol{\mu}) = -\sum_{b=1}^{N} p_m(b, \boldsymbol{\mu}) \ln(p_m(b, \boldsymbol{\mu})), \tag{3.17}$$

where $p_q(a)$ and $p_m(b, \boldsymbol{\mu})$ are the marginal probability distributions (e.g., $p_m(b, \boldsymbol{\mu})$ gives the probability that pixel $\mathbf{u}'$ in image $\mathbf{I}_n^m$ has intensity that falls into bin b). An example of the marginal probability distributions are shown in Figures 3.4b and 3.4d. These marginal probability distributions are simply the image histograms normalized by the total number of pixels. The entropies are scalar values computed using these distributions. For the example shown in Figure 3.4, the joint and both marginal entropies evaluate to a value of 2.83 since the artificial images contain the same information content. The resulting NID between the two images is therefore 0. MI is robust to appearance changes such as a linear shift in image intensity throughout the image as shown in this example.

To register the images we determine the optimal warping parameters, $\boldsymbol{\mu}^* = [s^* \; \psi^* \; t_x^* \; t_y^*]^\top$, to minimize the NID between a query image and selected map image:

$$\boldsymbol{\mu}_k^* = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \, NID(\mathbf{I}_k^q, \mathbf{I}_n^m, \boldsymbol{\mu}). \tag{3.18}$$

Since this optimization problem is non-convex, we solve it using Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm; a quasi-Newton method. Previous work has modified the discrete marginal and joint probabilty distribution functions to be analytically differentiable by using B-spline weights [Pascoe et al., 2015]. We instead use a simpler approach of a central difference numerical gradient. Furthermore, we apply a two-step optimization procedure. The first step applies a Gaussian blur on both images to smooth out the cost function and gradients and optimizes with the blurred images. The second step uses the optimal warping from the blurred optimization to initialize a refined optimization that operates on the raw images.

The query-to-map pose parameters (3.10) are recovered from the optimal warping:

$$x_{k_s}^{n_s,k_s} = -t_x^* s^* z_{n_s}^{g,n_s} \tag{3.19a}$$

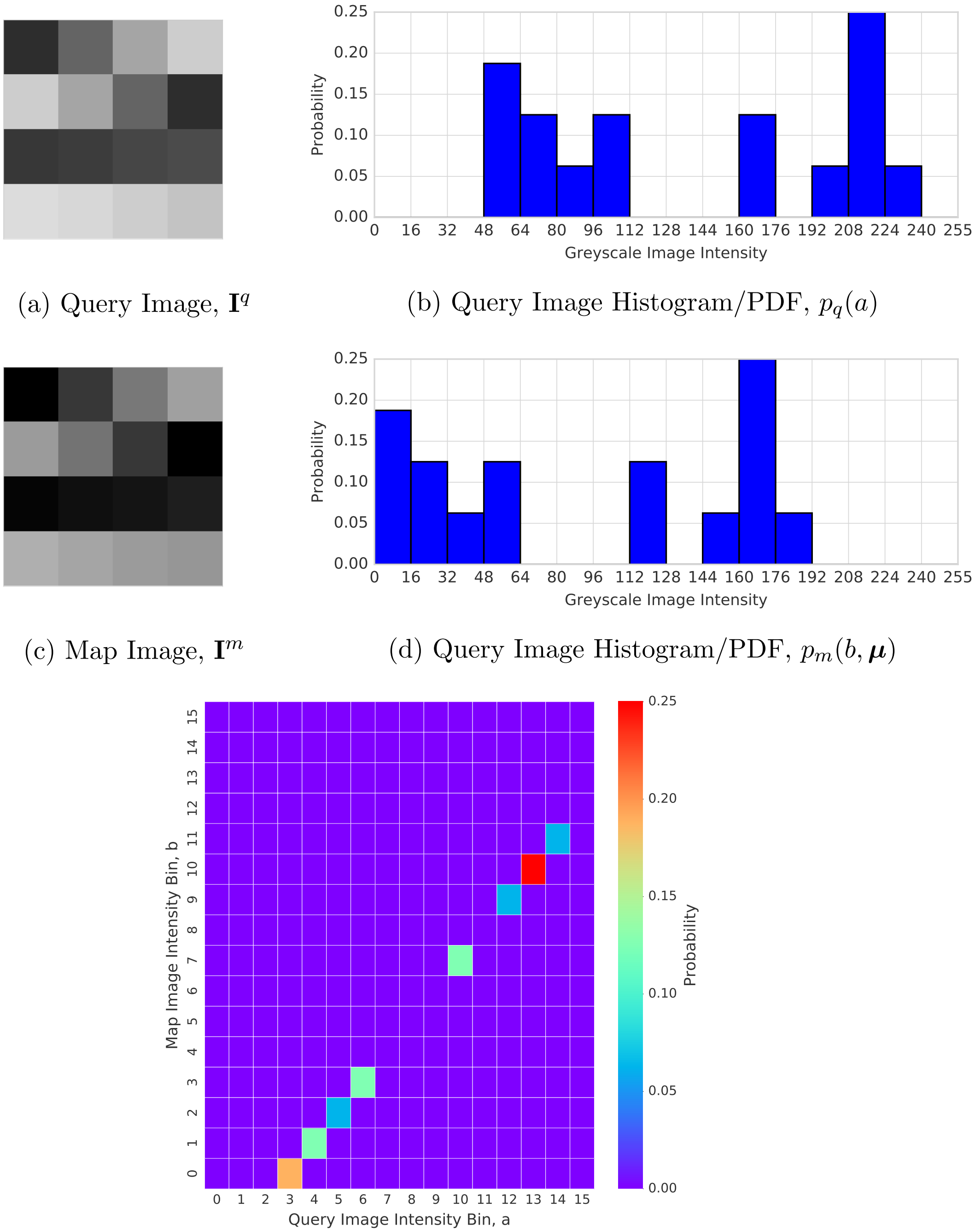
$$y_{k_s}^{n_s,k_s} = -t_y^* s^* z_{n_s}^{g,n_s} \tag{3.19b}$$

$$z_{k_s}^{n_s,k_s} = z_{n_s}^{g,n_s}(s^* - 1) \tag{3.19c}$$

$$\psi_{k_s,n_s} = -\psi^*, \tag{3.19d}$$

where $z_{n_s}^{g,n_s}$ is the distance from the nadir-pointed virtual camera to the ground. Therefore, for each successfully registered image we have an estimate for $\mathbf{T}_{k_s,n_s}$, which is used to obtain the global pose via (3.9).

It is important to select an appropriate map image $\mathbf{I}_n^m$ to register the query image $\mathbf{I}_k^q$. We compute the NID between the query image and unwarped map images in a radius around a predicted pose given by VO (3.28b). This strategy aims to provide the best aligned images before any warping. A simpler strategy is to select the spatially nearest map image to the predicted pose but this relies on having accurate predictions and is less robust to drift. We start with a larger search radius (e.g., 10 m) until VO is scaled



(a) Query Image, $\mathbf{I}^q$

(b) Query Image Histogram/PDF, $p_q(a)$

(c) Map Image, $\mathbf{I}^m$

(d) Query Image Histogram/PDF, $p_m(b, \boldsymbol{\mu})$

(e) Joint PDF, $p_{qm}(a, b, \boldsymbol{\mu})$

Figure 3.4: An example of the probability distributions with $N = 16$ bins for two artificial images. The query image is generated from the map image by increasing every greyscale intensity value by 48 with no clipping (i.e., it is a 'recolouring'). In this example, we can assume the images are aligned thus the sRt warping parameters are $\boldsymbol{\mu} = [1\ 0\ 0\ 0]^\top$. The joint probability distribution consists of only an off-diagonal since the two images are related by a shift in image intensity. Mutual Information is robust to this kind of appearance change; the resulting NID between these two images is 0.

and then reduce (e.g., to 4 m) for subsequent registrations. If registration is unsuccessful for multiple keyframes in a row, then we once again inflate the search radius. An image registration is deemed unsuccessful if the position distance or relative yaw from the registered pose to the predicted pose (3.28b) is too large or the optimizer fails to converge.

3.3.3 Pose Filtering

We follow the methods in [Barfoot, 2017] to compound uncertain transforms and fuse uncertain pose estimates. The estimation is performed in a local coordinate frame where $\mathbf{T}_{W,0}$ is constructed using the RTK position and vehicle attitude at the first VO keyframe.

VO provides a relative transform between keyframes, $\mathbf{T}_{k,k-1}$, which serves as an input to our filtering framework. We found the VO uncertainties to be overconfident so we define our own. Since VO is unscaled, we use a large uncertainty,

$$\mathbf{Q}_k = \text{diag}(0.04, 0.04, 0.04, 0.05, 0.05, 0.01),$$

until there are enough recent localizations to estimate a scale factor, λ, after which we reduce the position uncertainty by a factor of 10. For the prediction step of our filter, we have two uncertain poses that we want to compound: our posterior estimate from the previous timestep $\{\bar{\mathbf{T}}_{k-1,0}, \hat{\mathbf{P}}_{k-1}\}$ and the latest VO estimate $\{\bar{\mathbf{T}}_{k,k-1}, \mathbf{Q}_k\}$. We can use our $SE(3)$ perturbation scheme (1.5) to express our uncertain transforms as

$$\hat{\mathbf{T}}_{k-1,0} = \exp(\boldsymbol{\epsilon}_{k-1}^{\wedge})\bar{\mathbf{T}}_{k-1,0}, \quad \boldsymbol{\epsilon}_{k-1} \sim \mathcal{N}(0, \hat{\mathbf{P}}_{k-1}), \tag{3.20}$$

$$\mathbf{T}_{k,k-1} = \exp(\mathbf{w}_k^{\wedge})\bar{\mathbf{T}}_{k,k-1}, \quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k). \tag{3.21}$$

By compounding these transforms we obtain

$$\mathbf{T}_{k,0} = \mathbf{T}_{k,k-1}\mathbf{T}_{k-1,0} \tag{3.22a}$$

$$\exp(\boldsymbol{\epsilon}_k^{\wedge})\bar{\mathbf{T}}_{k,0} = \exp(\mathbf{w}_k^{\wedge})\bar{\mathbf{T}}_{k,k-1} \exp(\boldsymbol{\epsilon}_{k-1}^{\wedge})\bar{\mathbf{T}}_{k-1,0} \tag{3.22b}$$

$$\exp(\boldsymbol{\epsilon}_k^{\wedge})\bar{\mathbf{T}}_{k,0} = \exp(\mathbf{w}_k^{\wedge}) \exp\left((\bar{\boldsymbol{\mathcal{T}}}_{k,k-1}\boldsymbol{\epsilon}_{k-1})^{\wedge}\right) \bar{\mathbf{T}}_{k,k-1}\bar{\mathbf{T}}_{k-1}, \tag{3.22c}$$

where $\bar{\boldsymbol{\mathcal{T}}}_{k,k-1}$ is the $SE(3)$ adjoint of $\bar{\mathbf{T}}_{k,k-1}$ computed using (1.6). So our nominal motion model is

$$\bar{\mathbf{T}}_{k,0} = \bar{\mathbf{T}}_{k,k-1}\bar{\mathbf{T}}_{k-1,0}, \tag{3.23}$$

and we can derive the resultant uncertainty, $\mathbf{P}_k$, from the perturbations using a second-order approximation (see [Barfoot, 2017]):

$$\mathbf{P}_k = \mathbf{Q}_k + \bar{\boldsymbol{\mathcal{T}}}_{k,k-1} \hat{\mathbf{P}}_{k-1} \bar{\boldsymbol{\mathcal{T}}}_{k,k-1}^{\top}. \tag{3.24}$$

Each incremental transform is scaled with the result of a sliding-window scale estimator. The VO scale estimator determines a scale factor to minimize the uncertainty-weighted error between the incremental posterior and VO position estimates inside a window of size N keyframes. Let $\mathbf{u}_k = \mathbf{r}_{k-1}^{k,k-1}$ be the incremental VO position estimates, $\mathbf{x}_k = \hat{\mathbf{r}}_{k-1}^{k,k-1}$ be the incremental posterior estimates from the filter, and $\mathbf{Q}_{p_k}$ be the position component of the VO uncertainty then

$$J(\lambda_k) = \frac{1}{2} \sum_{j=k-N-1}^{k-1} (\mathbf{x}_j - \lambda_k \mathbf{u}_j)^{\top} \mathbf{Q}_{p_j}^{-1} (\mathbf{x}_j - \lambda_k \mathbf{u}_j), \tag{3.25}$$

is the cost function used to estimate the scale for each keyframe, provided there are N recent successful localizations. The cost function is quadratic in the scale factor allowing us to find the optimum in one step ($\frac{\delta J(\lambda)}{\delta \lambda} = 0$):

$$\left(\sum_{j=k-N-1}^{k-1} \mathbf{u}_j^{\top} \mathbf{Q}_{p_k}^{-1} \mathbf{u}_j \right) \lambda_k^{*} = \left(\sum_{j=k-N-1}^{k-1} \mathbf{x}_j^{\top} \mathbf{Q}_{p_k}^{-1} \mathbf{u}_j \right) \tag{3.26}$$

Our image registration provides a measurement of the vehicle pose for each keyframe, $\mathbf{T}_{k,0}$, which we use to apply corrections to VO. In the future, we aim to explore proper uncertainty quantification for the image registrations. One idea is to use inverse of the Hessian at the optimum of the MI optimization (3.18): this Hessian describes the curvature of a cost function at the optimal sRt parameters, and therefore can be used to give an indication of the quality or confidence of the optimum for each image registration. However, for now we use a fixed measurement covariance where

$$\mathbf{R}_k = \text{diag}(0.11, 0.11, 1.0, 0.01, 0.01, 0.01).$$

Therefore, the correction step fuses two uncertain poses with the error between them defined as

$$\mathbf{e}_k = \ln \left(\mathbf{T}_{k,0} \check{\mathbf{T}}_{k,0}^{-1} \right)^{\vee}, \tag{3.27}$$

where $\mathbf{e}_k \in \mathbb{R}^6$ is a pose vector obtained using our logarithmic mapping (1.4).

As a result, our filtering equations are

$$\check{\mathbf{P}}_k = \mathbf{Q}_k + \boldsymbol{\mathcal{T}}_{k,k-1}\hat{\mathbf{P}}_{k-1}\boldsymbol{\mathcal{T}}_{k,k-1}^{\top} \tag{3.28a}$$

$$\check{\mathbf{T}}_{k,0} = \mathbf{T}_{k,k-1}\hat{\mathbf{T}}_{k-1,0} \tag{3.28b}$$

$$\mathbf{K}_k = \check{\mathbf{P}}_k \left(\check{\mathbf{P}}_k + \mathbf{R}_k\right)^{-1} \tag{3.28c}$$

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k)\check{\mathbf{P}}_k \tag{3.28d}$$

$$\hat{\mathbf{T}}_{k,0} = \exp\left(\left(\mathbf{K}_k \ln(\mathbf{T}_{k,0}\check{\mathbf{T}}_{k,0}^{-1})^{\vee}\right)^{\wedge}\right)\check{\mathbf{T}}_{k,0}, \tag{3.28e}$$

where $\boldsymbol{\mathcal{T}}_{k,k-1}$ is the adjoint of $\mathbf{T}_{k,k-1}$ computed using (1.6), the prior uncertainty $\check{\mathbf{P}}_k$ is a second-order approximation, and $\mathbf{K}_k$ is the Kalman gain. We refer the reader to [Barfoot, 2017] for more detail. For unsuccessful registrations, the predicted position and uncertainties are propagated (i.e., $\hat{\mathbf{T}}_{k,0} = \check{\mathbf{T}}_{k,0}$ and $\hat{\mathbf{P}}_k = \check{\mathbf{P}}_k$). The posterior global vehicle pose at each keyframe is obtained by

$$\hat{\mathbf{T}}_{W,k} = \mathbf{T}_{W,0}\hat{\mathbf{T}}_{k,0}^{-1}. \tag{3.29}$$

3.4 Experimental Setup

3.4.1 UAV Dataset Collection

The experiments are conducted with data collected at UTIAS using the hardware setup shown in Figure 3.5. We use the DJI Matrice 600 Pro multirotor UAV with a 3-axis DJI Ronin-MX gimbal. A StereoLabs ZED camera is connected to the onboard NVIDIA Tegra TX2 computer to provide 1280 × 720 RGB stereo images at 10FPS. These images are downscaled to 560 × 315 and converted to greyscale for VO and image registration. The gimbal connects to the flight controller to provide angular positions from joint encoders at 10 Hz. The RTK-GPS system provides the vehicle position at 5 Hz, and an IMU provides the vehicle attitude at 50 Hz.

The first dataset is a simple 303 m rectangular path flown with height variations beween 45−48 m AGL. It was collected in the fall during an overcast day and is the primary dataset used for development and tuning of our method. We also collected six datasets during a sunny summer day on a more complicated 1132 m path flown with height variations between 36 − 42 m AGL to show the ability of our method to localize a) at lower altitudes, and b) using a single map image database despite significant lighting changes in the real-world images. We collect a dataset near distinctive times of the day: sunrise (`06:17 AM`), morning (`08:50 AM`), noon (`11:54 AM`), afternoon (`02:50 PM`), evening

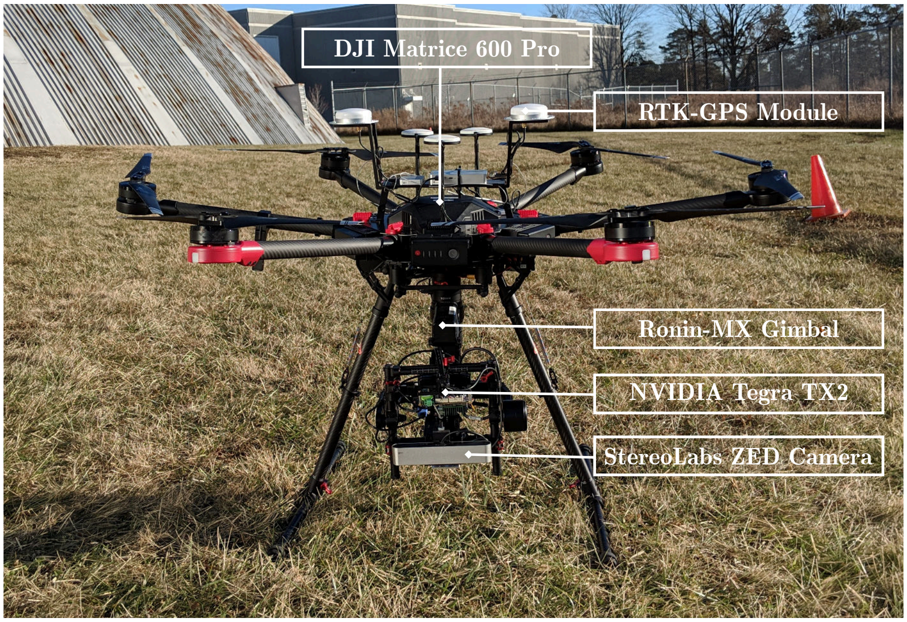


Figure 3.5: A 3-axis gimballed stereo camera on a multirotor UAV with an onboard computer is used for our data collection.

(`05:50 PM`), and sunset (`08:24 PM`). Figure 3.6 shows examples of the extreme lighting changes that occur throughout the day at two locations. These flights are over both man-made structure and significant stretches of vegetation to evaluate the performance in different environments.

3.4.2 Map Images

The set of geoferenced map images, $\mathcal{I}^m$, is generated from the 3D view in Google Earth at desired camera poses in an offline step. We define a virtual camera at each pose with the same focal length as the UAV-mounted camera so that query and map images taken at the same pose can have a nearly perfect alignment when the 3D reconstruction is precise. We also use GE elevation data to obtain the height of the camera AGL at each pose.

After planning the UAV path, we render images at discrete poses along the nominal path. For this work, all images are rendered with the camera facing east and pointed in the nadir direction. A multirotor UAV is able to travel in all directions with the same heading allowing this restriction to be feasible for many applications such as drone delivery. It is also possible to use a second gimbal to orient an application-specific sensor. Images are generated every 3 m along the nominal path to match our gimballed VO pipeline, which creates a new keyframe at approximately the same spacing. We extend

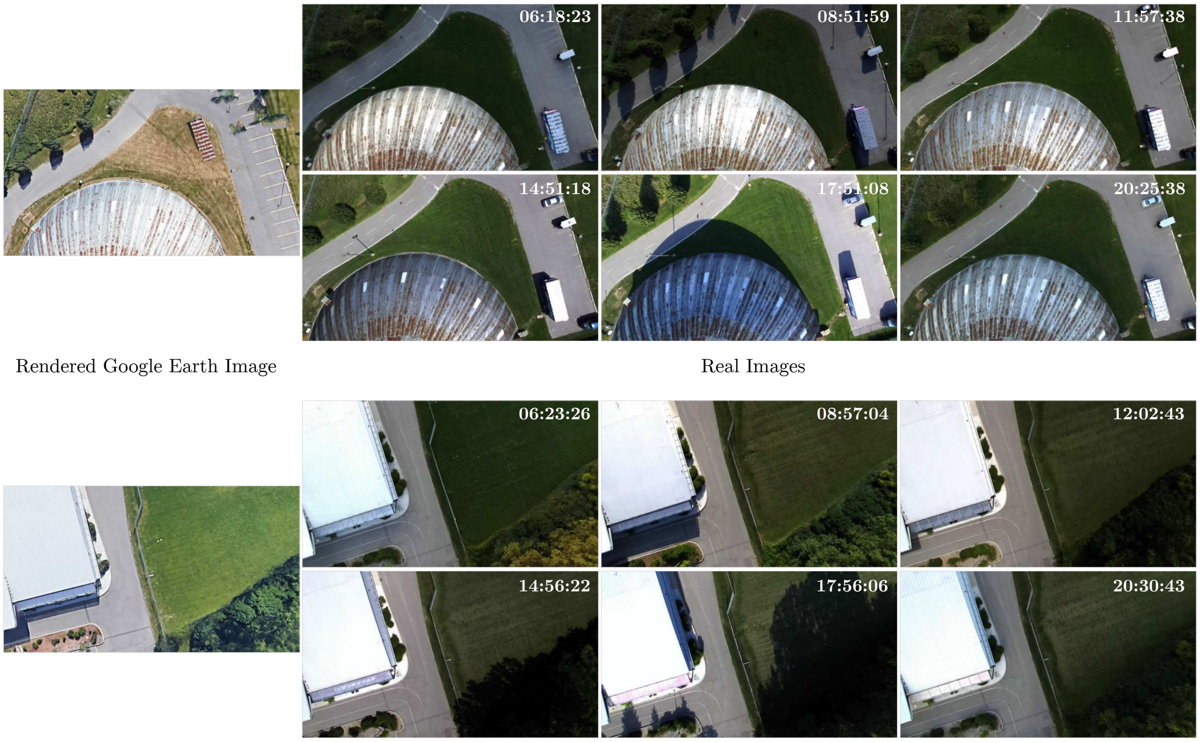


Figure 3.6: Examples of lighting changes that occur from sunrise to sunset east of the dome (top) and north-west of the soccer field (bottom). The Google Earth reconstruction appears to contain late morning to early noon shadows.

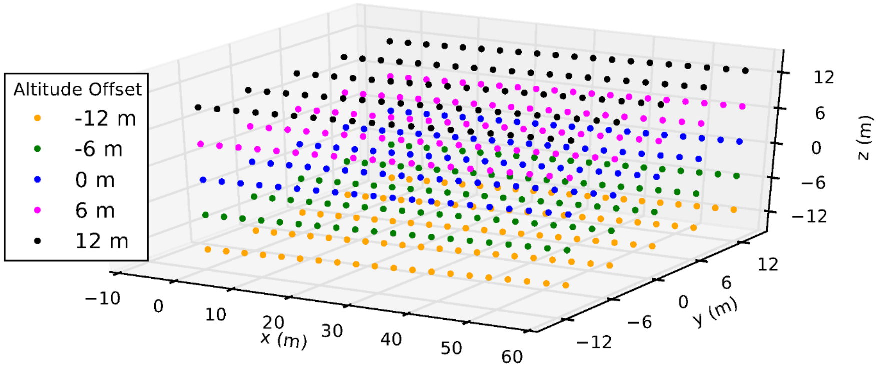


Figure 3.7: Location of the georeferenced map images relative to the starting position of a desired path. Only a short, straight part of the path is shown for clarity. We render images at multiple locations around the nominal path to ensure we capture non-planar changes to the scene.

the images 12 m to the left and right, and above and below the path with a 6 m spacing. The end result is a rectangular tube of images centered along the path with a 24 × 24 m cross-section and 6 × 6 × 3 m spacing between images. Figure 3.7 shows an example of the position of map images for a short part of the beginning of a path. The tube of images ensures that if the vehicle deviates off the nominal path, there is a map image taken from a nearby pose that captures the non-planar changes in the scene (e.g., side of a building becoming visible). This allows us to accurately localize with *sRt* warping at lower altitudes where the planar scene assumption is less valid. The spacing was chosen heuristically: the largest possible distance to any map image is 4.5 m when inside the tube, which is approximately the width of our convergence basin at the altitude flown in these experiments.

The map images could be extended beyond 12 m or cover an entire flight area. The only limitation is the storage available on the UAV. Although we save high-resolution RGB images, the image registration algorithm only uses 560 × 315 4-bit greyscale images (the NID is computed using 16-bin histograms of the greyscale intensities). Our 313 m and 1.1 km paths contain 2393 and 8992 map images, respectively, which would require approximately only 212 Mb and 794 Mb if saved in the minimum required format. With today's large capacity and inexpensive storage, map images covering several square kilometres could easily be stored onboard.

3.4.3 Ground Truth

It is important to note that the RTK-GPS and GE global coordinate frames, $\underrightarrow{\mathcal{F}}_{W'}$ and $\underrightarrow{\mathcal{F}}_{W}$, respectively, do not perfectly align. Therefore, we uniformly sample 10% of the posterior pose estimates along the path and use these to align the coordinate frames with a transform, $\mathbf{T}_{W',W}$, that is determined by miniziming the uncertainty-weighted relative pose errors between the RTK-GPS poses and posterior pose estimates. We report all image registration and filtered errors on the remaining 90% of the path.

From our filter we have the transform from the GE world frame to the estimated pose with uncertainty at keyframe k, $\{\bar{\mathbf{T}}_{k,W}, \mathbf{P}_k\}$. Also, we have the transform from the vehicle pose to the RTK-GPS world frame with uncertainty for each keyframe, $\{\bar{\mathbf{T}}_{W',k}, \mathbf{S}_k\}$. These two uncertain transforms can be compounded to obtain a nominal or mean transform,

$$\bar{\mathbf{T}}_{W',W} = \bar{\mathbf{T}}_{W',k}\bar{\mathbf{T}}_{k,W}, \tag{3.30}$$

with uncertainty

$$\boldsymbol{\Sigma}_k = \mathbf{S}_k + \bar{\boldsymbol{\mathcal{T}}}_{W',k}\mathbf{P}_{k,W}\bar{\boldsymbol{\mathcal{T}}}_{W',k}^{\top}. \tag{3.31}$$

Note that $\bar{\mathbf{T}}_{W',W}$ is not associated with $\mathbf{T}_{W',W}$, which is the transform we wish to estimate to align the coordinate frames.

We can define the error for each keyframe as

$$\mathbf{e}_k(\mathbf{T}_{W',W}) = \ln\left(\bar{\mathbf{T}}_{W',k}\bar{\mathbf{T}}_{k,W}\mathbf{T}_{W',W}^{-1}\right)^{\vee}. \tag{3.32}$$

Using our $SE(3)$ perturbation scheme we have $\mathbf{T}_{W',W} = \exp(\boldsymbol{\epsilon}^{\wedge})\mathbf{T}_{\text{op}}$ so that

$$\begin{aligned}\mathbf{e}_k(\mathbf{T}_{W',W}) &= \ln\left(\bar{\mathbf{T}}_{W',k}\bar{\mathbf{T}}_{k,W}\mathbf{T}_{\text{op}}^{-1}\exp(-\boldsymbol{\epsilon}^{\wedge})\right)^{\vee} && (3.33\text{a})\\ &\approx \mathbf{e}_k(\mathbf{T}_{\text{op}}) - \mathbf{G}_k\boldsymbol{\epsilon}, && (3.33\text{b})\end{aligned}$$

where $\mathbf{G}_k = \boldsymbol{\mathcal{J}}(-\mathbf{e}_k(\mathbf{T}_{\text{op}}))^{-1}$ is the inverse left Jacobian [Barfoot, 2017]. Our approximation is valid when $\bar{\mathbf{T}}_{W',k}\bar{\mathbf{T}}_{k,W}\mathbf{T}_{\text{op}}^{-1}$ is small which is true provided we start with a good intial guess. Our cost function is defined as

$$\begin{aligned}J(\mathbf{T}_{W',W}) &= \frac{1}{2}\sum_{j=0}^{K}\mathbf{e}_k(\mathbf{T}_{W',W})^{\top}\boldsymbol{\Sigma}_k^{-1}\mathbf{e}_k(\mathbf{T}_{W',W}) && (3.34\text{a})\\ &\approx \frac{1}{2}\sum_{j=0}^{K}(\mathbf{e}_k(\mathbf{T}_{\text{op}}) - \mathbf{G}_k\boldsymbol{\epsilon})^{\top}\boldsymbol{\Sigma}_k^{-1}(\mathbf{e}_k(\mathbf{T}_{\text{op}}) - \mathbf{G}_k\boldsymbol{\epsilon}). && (3.34\text{b})\end{aligned}$$

We can solve this iteratively using Gauss-Newton procedure where

$$\left(\sum_{j=0}^{K}\mathbf{G}_k^{\top}\boldsymbol{\Sigma}_k^{-1}\mathbf{G}_k\right)\boldsymbol{\epsilon}^{*} = \left(\sum_{j=0}^{K}\mathbf{G}_k^{\top}\boldsymbol{\Sigma}_k^{-1}\mathbf{e}_k(\mathbf{T}_{\text{op}})\right) \tag{3.35}$$

is used to update our operating point:

$$\mathbf{T}_{\text{op}} \leftarrow \exp(\boldsymbol{\epsilon}^{*\wedge})\mathbf{T}_{\text{op}}. \tag{3.36}$$

After convergence we have an optimal $\mathbf{T}_{W',W} = \mathbf{T}_{\text{op}}^{*}$ that is used to align the GE and RTK-GPS coordinate frames.

3.5 Results and Discussion

3.5.1 MI-based Image Registration

We first show an example of aligning two images with sRt warping using the NID cost function. Figure 3.8 shows the cost function values swept over the four warping parameters for the first image in the overcast dataset. The warping that generates the minimum NID is often the best image alignment as shown in this example. However, it is not guaranteed that there will be a single minimum or even that the global minimum corresponds to the best alignment. Furthermore, the absolute NID value highly depends on the scene. The near-perfect alignment in the example occurs at nearly 0.93 even though the NID is a value between 0 and 1 with a lower value indicating more similarity. For this reason, we use a geometric criterion for classifying image registration failures instead of thresholding the NID.

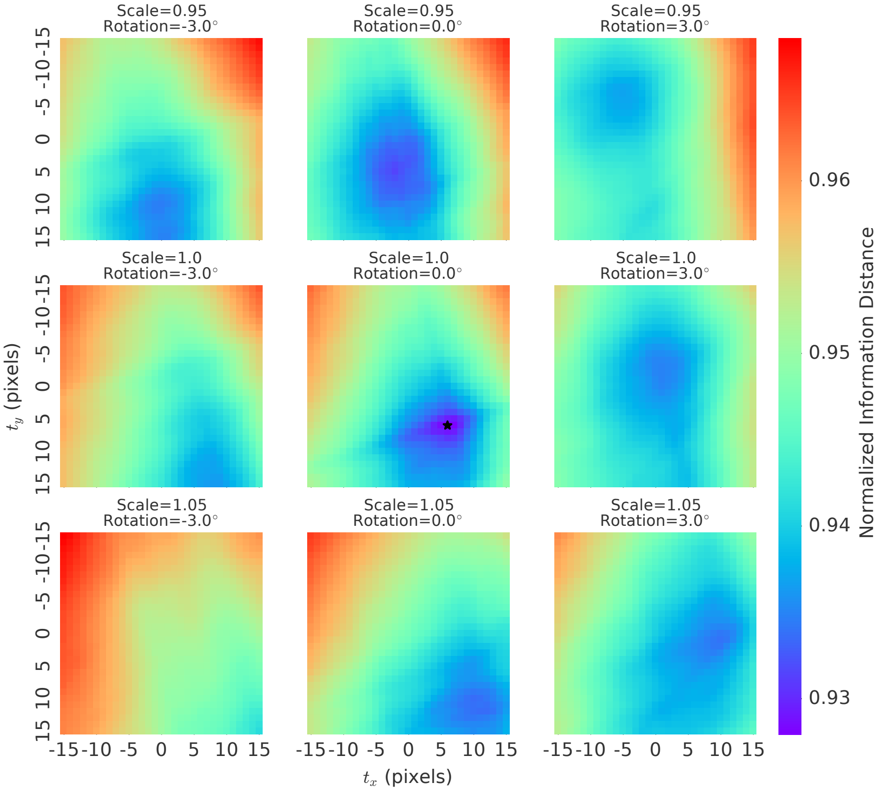
Next, we show the image alignment using our two-step optimization approach using the same two images as in Figure 3.8 to compare with the grid search results. Figures 3.9a and 3.9b show the cost function value per iteration during this optimization. The optimal warping parameters from the refined optimization are a scale of 0.99, rotation of 0.14 degrees, and translation of $(5.6, 6.1)$ pixels, which result in an NID of 0.921. The grid search resulted in an optimum at a scale of 1.0, rotation of 0 degrees, and translation of $(6, 6)$ pixels with a NID of 0.928. Our two step optimization approach successfully finds the optimum with fewer function evaluations and provides sub-pixel and sub-degree warping parameters.

We now present our image registration results on all datasets using our two-step optimization approach. Table 3.1 shows the success rate and Root-Mean-Square Error (RMSE) computed using all registrations and only successful ones. The optimizer always converged to a solution thus all failures were instead due to poor image alignment.

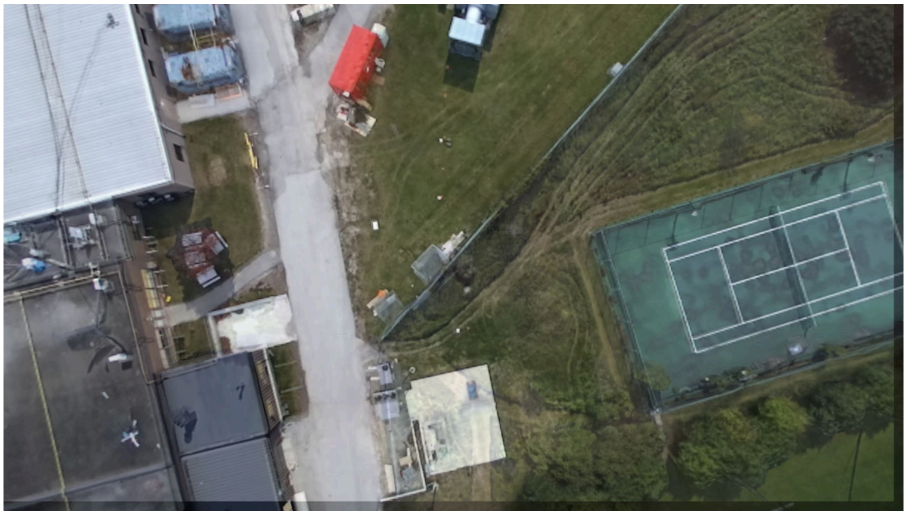
Table 3.1: Summary of MI-based Image Registration Results

Lighting Condition	**Registration Success (%)**	**Successful Registrations RMSE (m)**				**All Registrations RMSE (m)**			
		long.	lat.	alt.	heading	long.	lat.	alt.	heading
Overcast	100	0.69	0.46	0.50	0.89	0.69	0.46	0.50	0.89
Sunrise	94.7	1.10	0.71	1.17	2.28	1.87	1.47	1.73	2.80
Morning	95.1	1.02	0.58	0.78	2.57	2.24	1.39	1.20	2.97
Noon	97.8	0.78	0.61	1.01	1.82	1.26	1.02	1.40	2.70
Afternoon	96.0	1.69	0.92	1.17	1.71	2.14	1.57	1.54	2.63
Evening	81.3	3.03	1.32	1.35	2.49	4.09	3.63	2.98	5.25
Sunset	87.5	1.95	1.12	1.55	2.64	3.03	1.95	2.54	3.06

For the overcast flight we successfully register every keyframe and achieve sub-metre position and sub-degree orientation errors. This is in part due to the higher altitude flight (although this is still relatively low compared to previous work), which provides more objects and boundaries to aid in the alignment. An example of the optimal NID values for each keyframe is shown in Figure 3.10. This further showcases the difficulty in using the absolute value to classify failures. For the sunrise to sunset flights, the registration

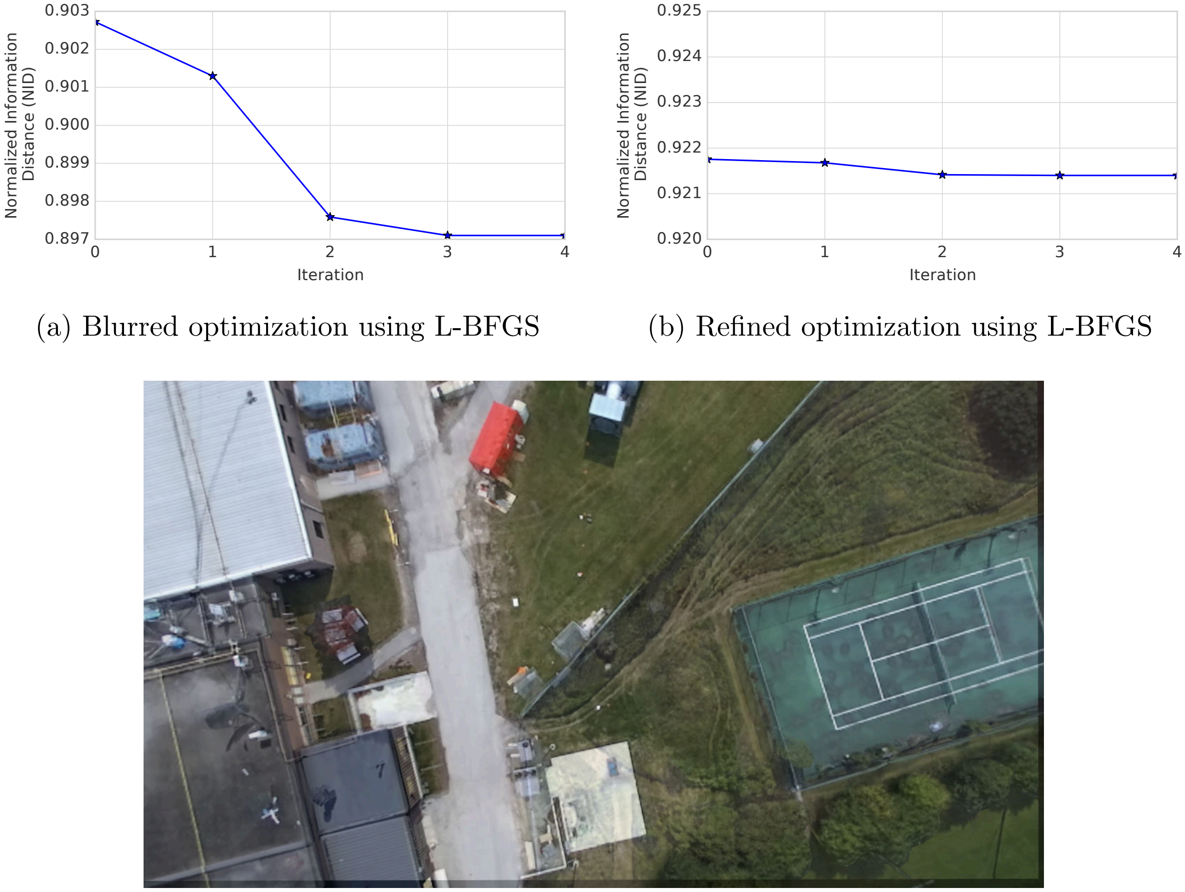


(a) NID cost function swept from -15 to 15 pixel translations in t_x and t_y at three different scales and rotations.



(b) Alpha blended image of the UAV query image (prominent) and the Google Earth map image warped with the optimal sRt parameters.

Figure 3.8: An example sRt alignment using the NID cost function showing the smoothness over the warping parameters with a clear optimum that corresponds to a nearly perfect alignment.



(a) Blurred optimization using L-BFGS

(b) Refined optimization using L-BFGS

(c) Alpha blended image of the UAV query image (prominent) and the Google Earth map image warped with the optimal sRt parameters from the refined optimization.

Figure 3.9: Optimizing the NID cost function over the sRt warping parameters using our two-step optimization approach produces sub-pixel and sub-degree sRt parameters that generate a lower NID than our costly grid search.

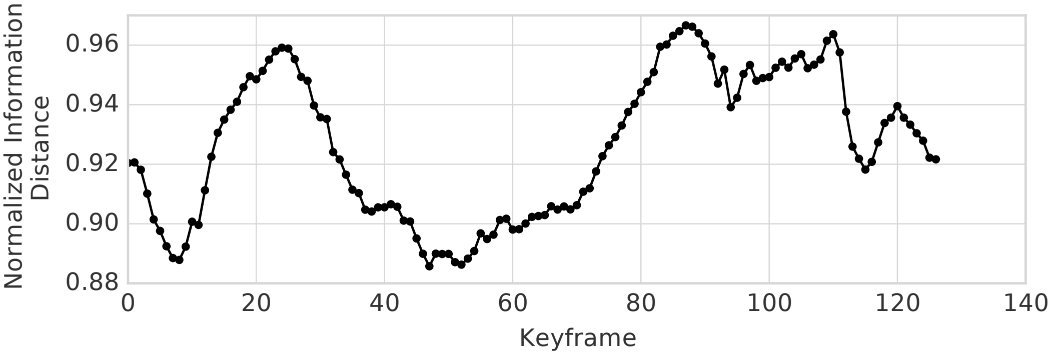
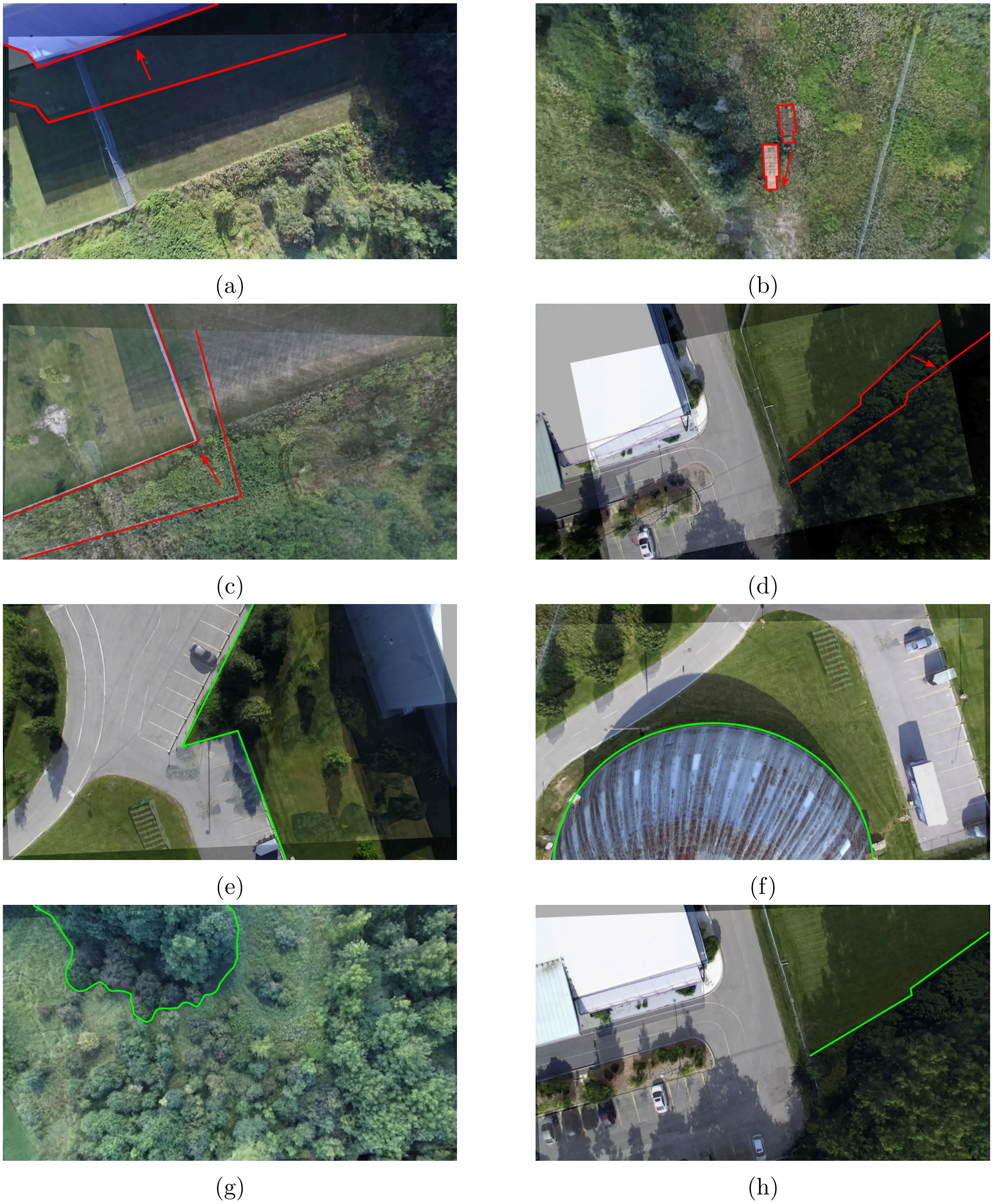


Figure 3.10: The optimal NID value from image registration for each keyframe in our overcast dataset.

performs the best at noon as expected; the GE 3D reconstruction in our flight area resembles early noon. The image registrations alone were able to achieve nearly less than 3 m and 3° position and heading RMSE.

There are two types of scenes that are particularly difficult for our image registration: scenes with lots of self-similar texture (e.g., vegetation in Figures 3.11b, 3.11c), and scenes with large shadows (e.g., Figures 3.11a, 3.11d). Self-similar texture results in many local minima in the registration cost function. Shadows can trick the MI into associating the shadow with its caster resulting in a strong local minimum that may even be a global minimum. These shadows were most prevalent in the evening flight resulting in its lower success rate. While our blurred optimization provides robustness to shallow local minima, we depend on good initial guesses to handle the aforementioned problematic areas. Figures 3.11f, 3.11g, 3.11h show examples of when the MI optimizer can settle in the correct local minimum with an initial guess given by VO near the true alignment. Another method to handle these scenes is to optimize over a window of keyframes. Although this may produce a suboptimal alignment for each individual keyframe, it prevents large jumps in the measured poses introduced by these additional minima.

The number of cost function evaluations required per image registration is presented in Figure 3.12. Currently we place a very loose constraint on the number of iterations allowed for optimization. If necessary, we can reduce this number as, in many cases, the additional iterations do not provide a significant improvement to the accuracy. The current offline implementation uses the `SciPy` library for optimization thus is not able to run in real-time. However, we strongly believe our upcoming C++ implementation will enable the image registration to run at rate of at least 1 Hz.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3.11: The top two rows show examples of bad alignments due to (a) alignment of the building with its shadow, (b) and (c) almost no structure to aid in alignment, and (d) alignment of the trees with their shadows. The bottom two rows show good alignments despite: (e) poor 3D reconstructions, (f) and (h) large shadows, and (g) very little structure. A better initial pose guess and slightly more structure in (h) compared to (d) allows the MI optimizer to correctly align the images.

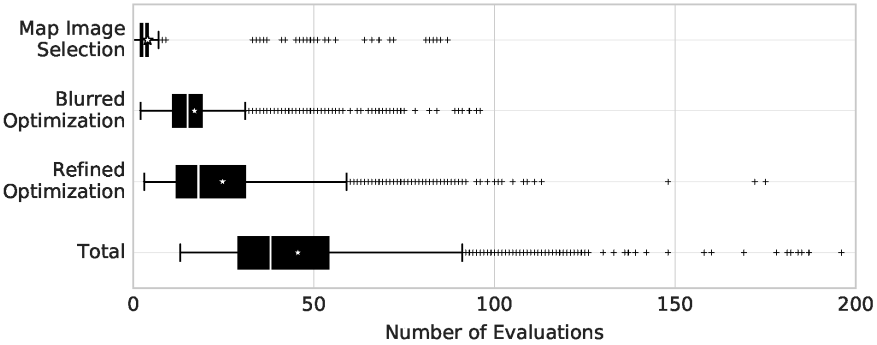


Figure 3.12: Number of cost function evaluations per image registration at each step of the proposed method.

3.5.2 Comparison with Feature-based Registration

We briefly present the results from a feature-based image registration scheme for comparison. We use the aforementioned VT&R framework with SURF. The GE images along the nominal path are used for the teach run. Repeats are then attempted with each of the sunrise to sunset flights but the result is a poor registration performance. The features are only capable of producing less than 7% successes per repeat where the registration is declared a failure if the number of MLESAC inliers is below 30.

Since it is quite obvious that feature matching across the rendered and real-world images will struggle, we also briefly evaluate the performance of a typical teach-and-repeat without the use of GE images. The sunrise flight is used as the teach with subsequent flights used as repeats. This results in 34.9%, 30.3%, 15.0%, 8.4%, and 72.4% success rate for morning to sunset. It is clear that the dramatic changes in lighting makes feature matching unreliable. The sunset flight is able to localize the most frequently due to the similar brightness and minimal shadows that appear during sunrise and sunset.

3.5.3 Filtered Pose Estimation

Finally, we highlight the accuracy we can achieve by fusing VO and our MI-based real-to-rendered image registration. The pure VO, image registration measurements, and filtered position estimates for the overcast flight are shown in Figure 3.13 alongside the ground truth. A histogram of the position and orientation errors is presented in Figure 3.14. It is clear that the combination of scaled VO to smooth out registrations and the registrations to correct for drifts in VO results in an accurate filtered global pose estimate. Figures 3.15 - 3.20 show the position estimates for each of our six flights from sunrise to sunset, and Figures 3.21, 3.22 show histograms of position and orientation errors, respectively.

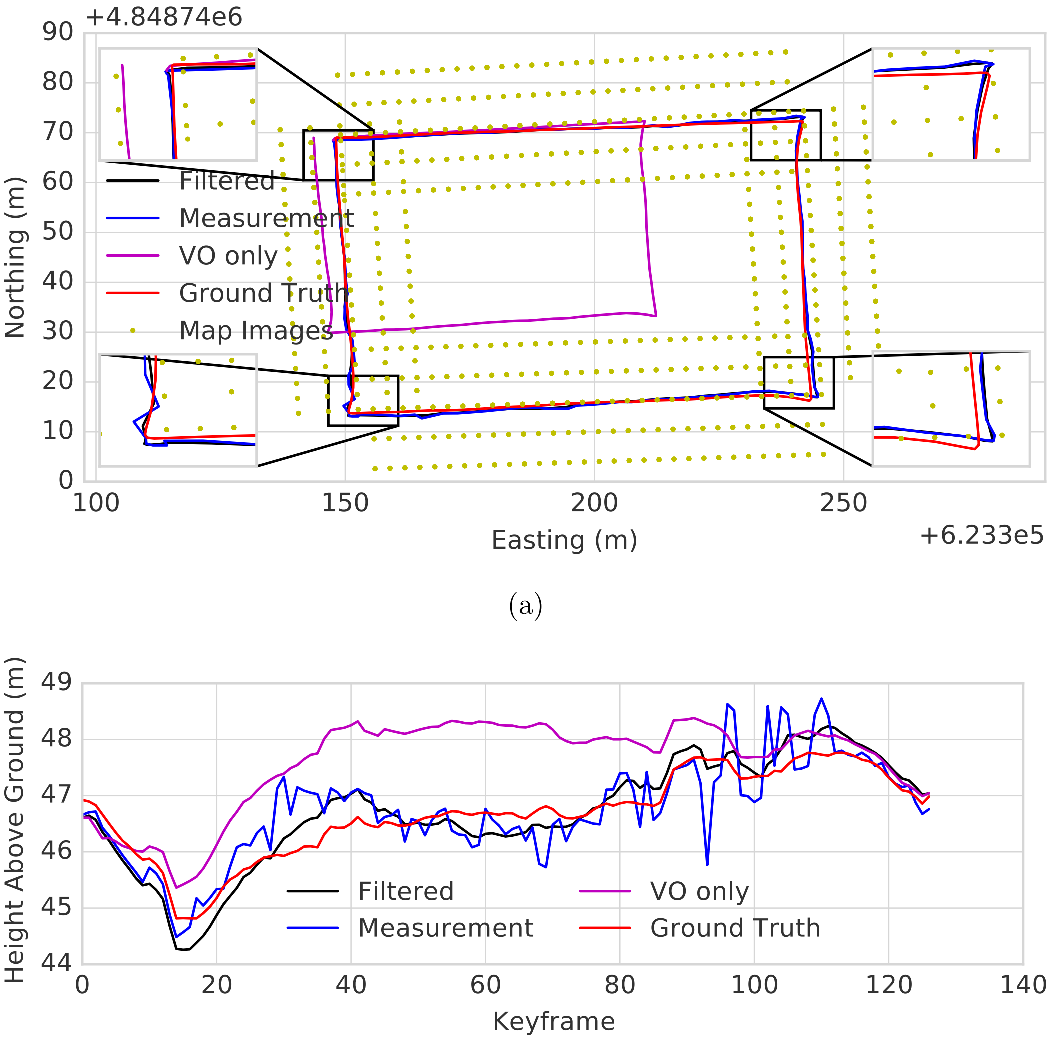
As we saw previously, a few particular areas were problematic for image registration in the presence of lighting changes. However, VO was able to carry the estimation through these small stretches ($5 - 10$ keyframes) of failures that predominantly occurred during the evening and sunset flights. Overall, our method is able to estimate a global pose throughout the day with a position accuracy that rivals (non-differential) GPS.

Table 3.2: Summary of Filtered Results

Lighting	**RMSE (m)**					
Condition	long.	lat.	altitude	roll	pitch	heading
Overcast	0.61	0.42	0.32	0.27	0.29	0.84
Sunrise	1.10	0.76	0.32	0.31	0.69	2.19
Morning	1.15	0.80	0.31	0.35	0.46	2.67
Noon	0.91	0.82	0.30	0.55	0.78	1.76
Afternoon	1.51	0.86	0.47	0.52	0.61	1.54
Evening	2.73	1.64	0.45	0.52	0.82	2.48
Sunset	1.78	0.76	0.51	0.52	0.84	2.55

3.6 Summary

We presented a method for global pose estimation of a UAV by visually localizing real-world images with pre-rendered images from a 3D reconstruction of the Earth. We used a MI-based dense image registration scheme to align the real and rendered images for metric localization. The registrations were then used to apply corrections to gimballed VO in a filtering framework. On multiple flights totaling 7.1 km of data with altitudes as low as 36 m AGL, we estimated the full pose with an accuracy on the order of a few metres and degrees. We also showed the ability to consistently localize over the course of a sunny summer day using a single database of pre-rendered images despite dramatic changes in lighting. Our method enables global pose estimation with a position accuracy on par with GPS.



(a)

(b)

Figure 3.13: Position estimates using our method for the 303 m overcast flight. The VO drifts significantly but our accurate image registrations allow us to estimate the scale and apply corrections.

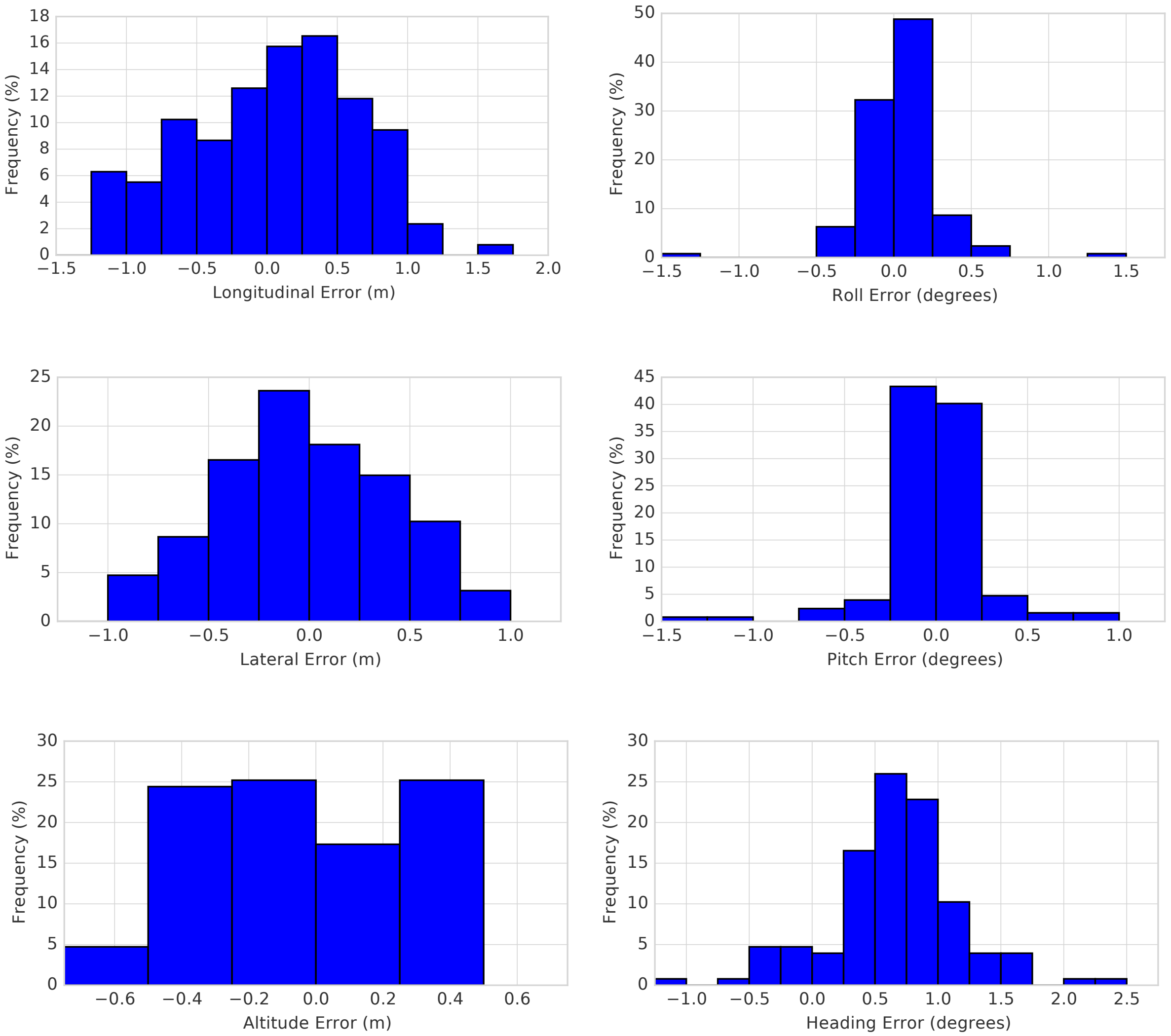
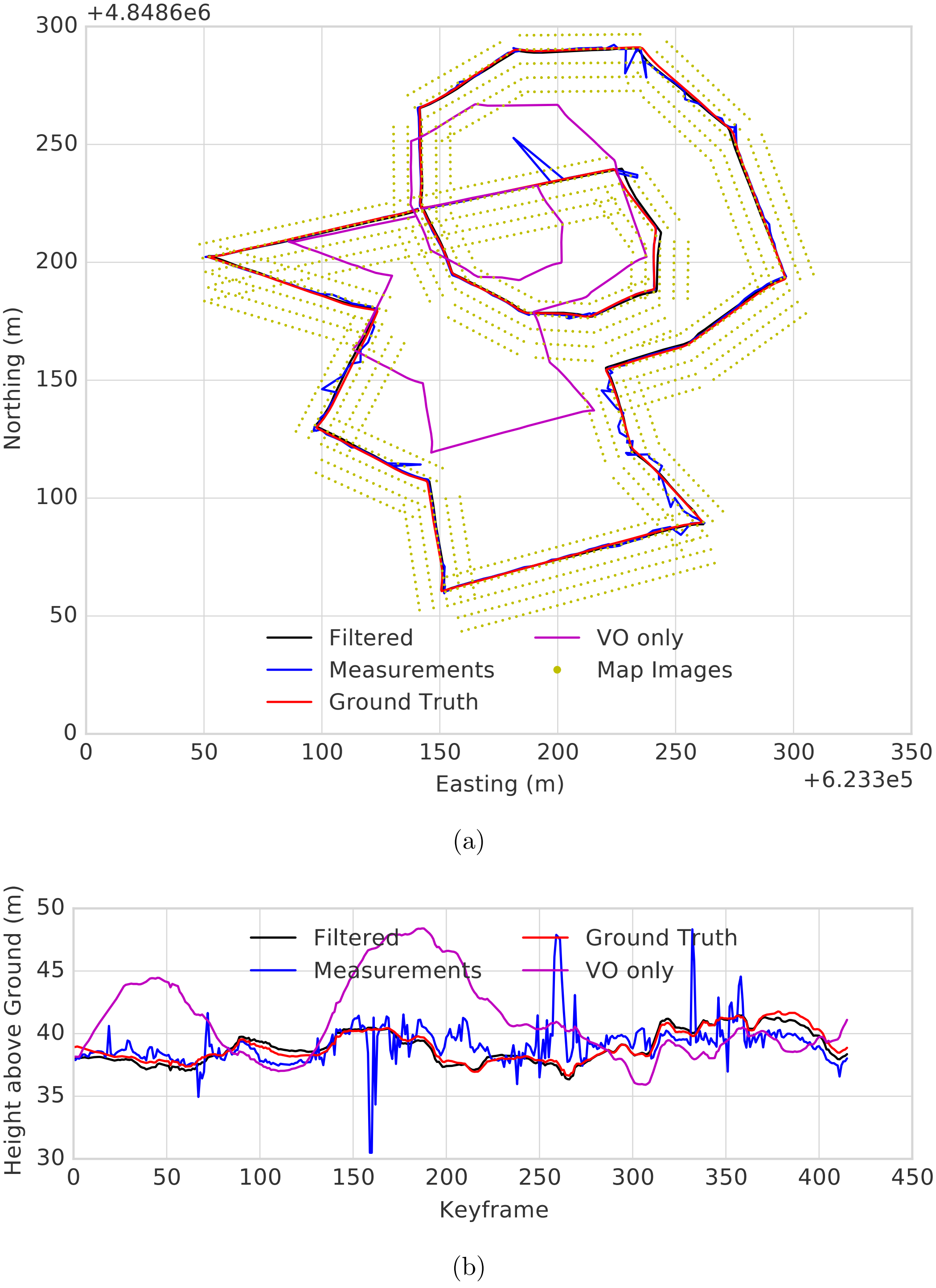


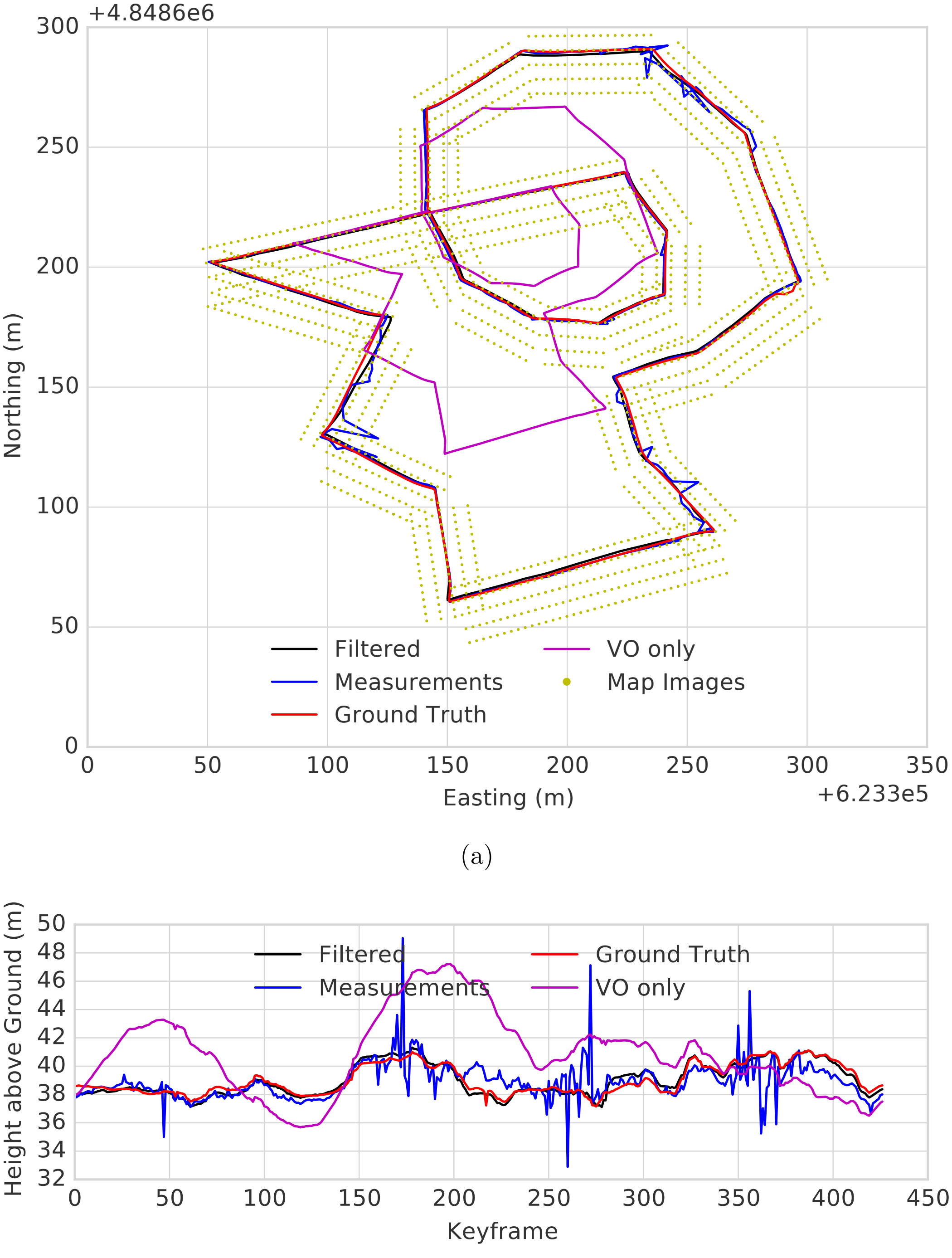
Figure 3.14: Histogram of filtered position and orientation errors for our overcast flight.



(a)

(b)

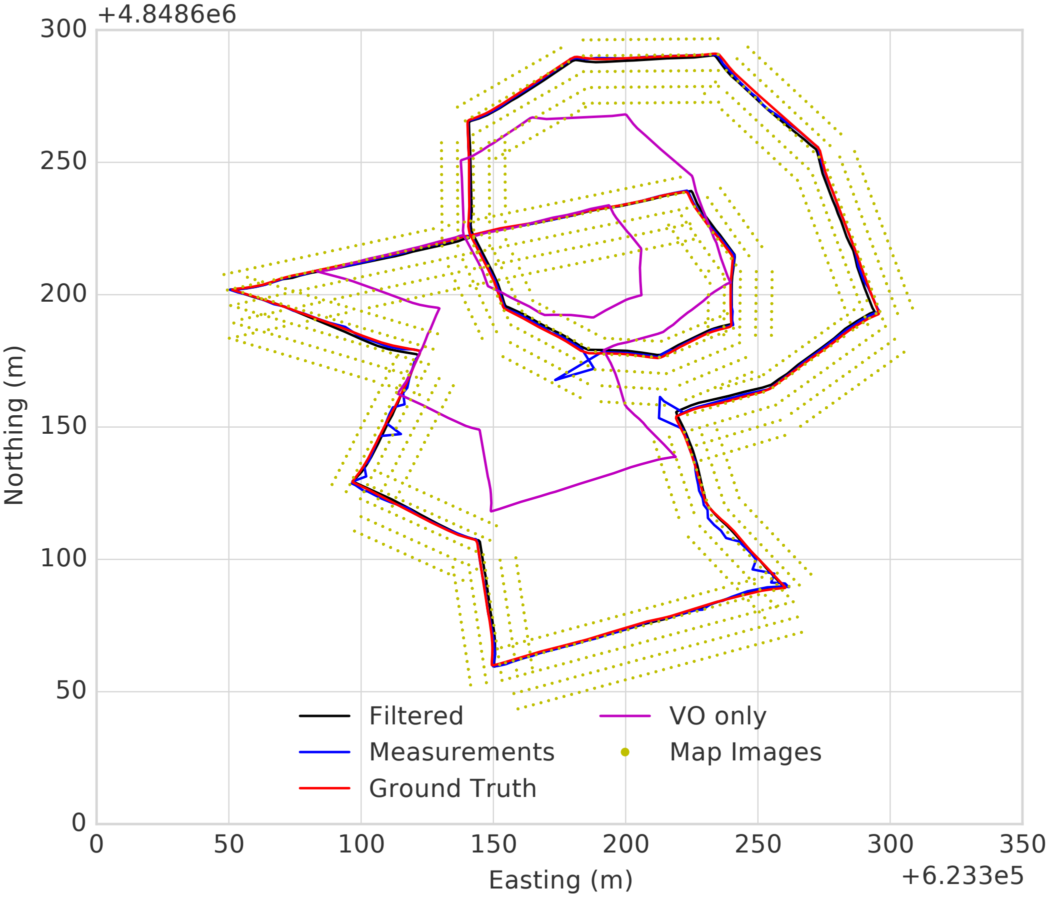
Figure 3.15: Position estimates for our 1132 m sunrise flight. The path starts at the intersection and spirals outward clockwise. Note that we show all image registration results here, even if they were classified as a failure. Our estimator performs well despite the drifting, unscaled VO, and image registrations using real images at sunrise that appear darker than the GE images.



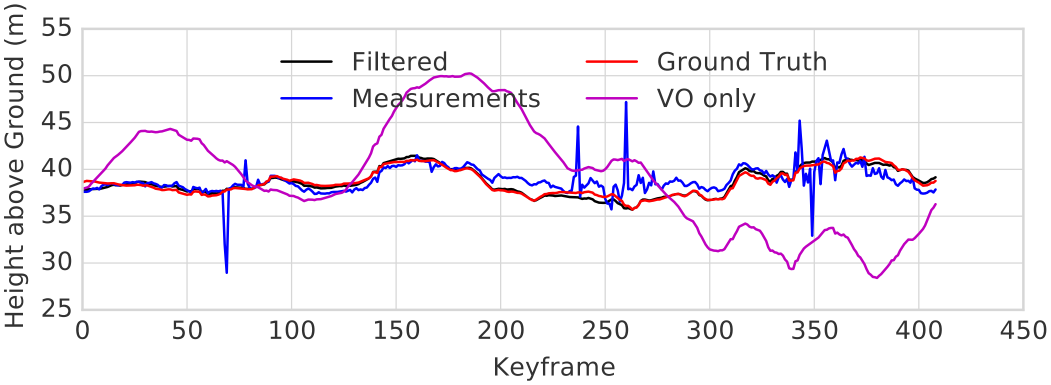
(a)

(b)

Figure 3.16: Position estimates for our 1132 m morning flight. The path starts at the intersection and spirals outward clockwise. Note that we show all image registration results here, even if they were classified as a failure. Our estimator performs well despite the drifting, unscaled VO.

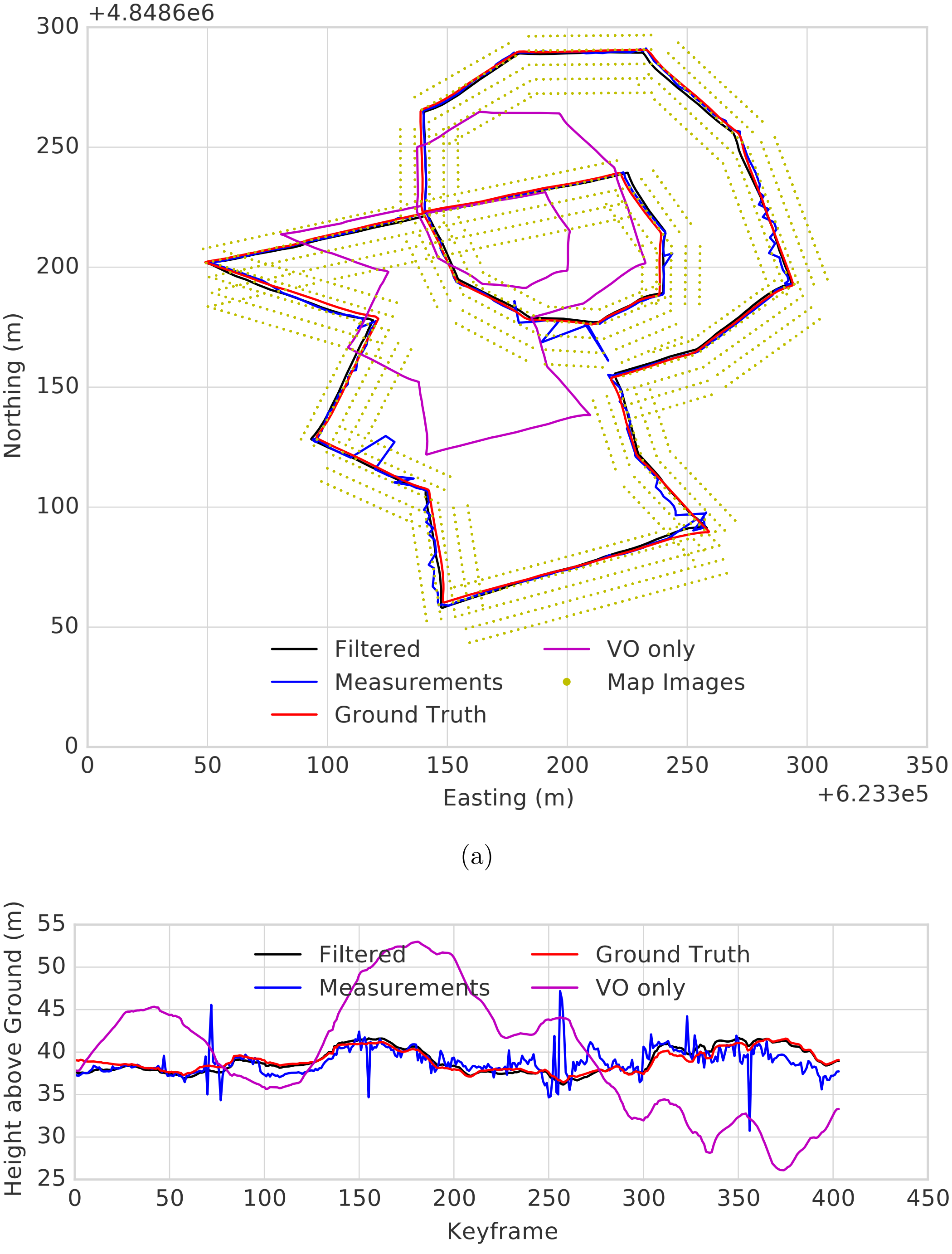


(a)



(b)

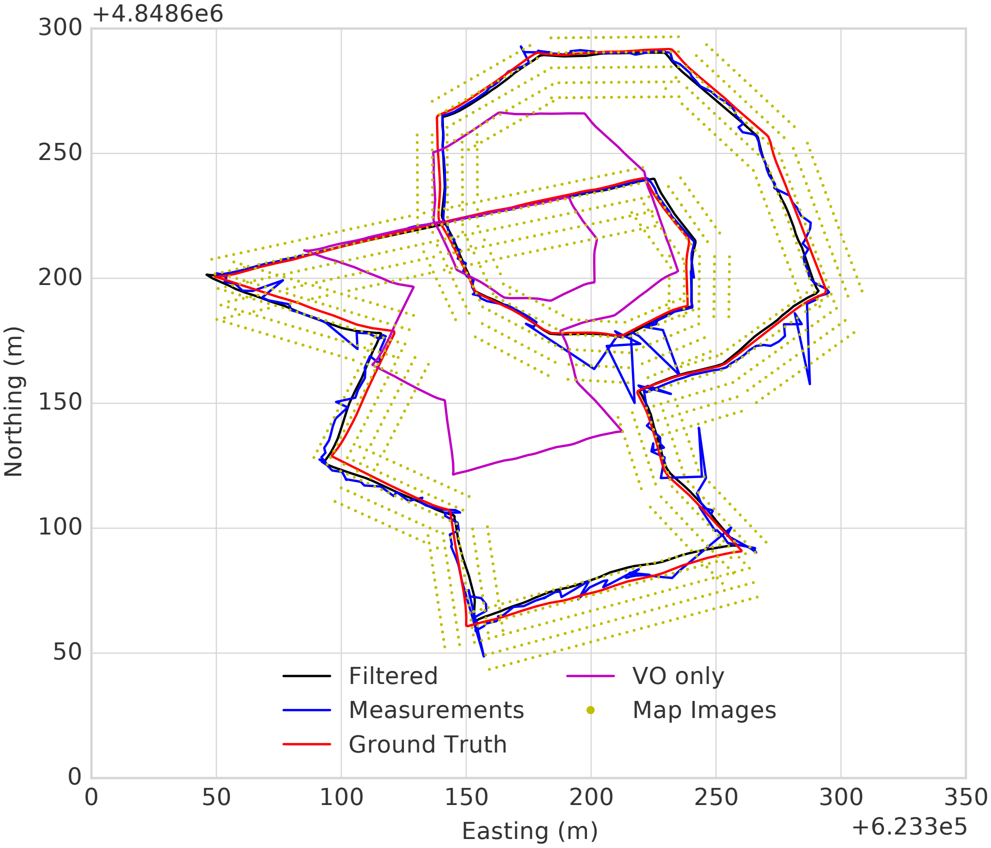
Figure 3.17: Position estimates for our 1132 m noon flight which shows our best localization performance. The path starts at the intersection and spirals outward clockwise. Note that we show all image registration results here, even if they were classified as a failure. Our estimator performs well despite the drifting, unscaled VO. Our image registrations during the noon flight are quite accurate due to the similar apperance between the real images at noon and the GE images.



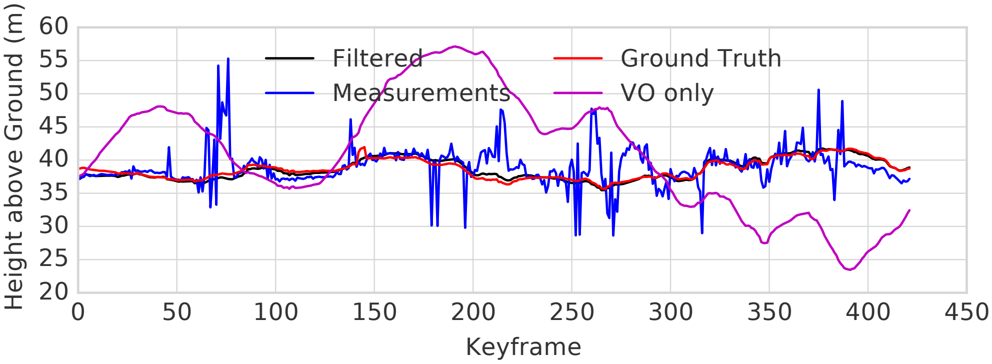
(a)

(b)

Figure 3.18: Position estimates for our 1132 m afternoon flight. The path starts at the intersection and spirals outward clockwise. Note that we show all image registration results here, even if they were classified as a failure. Our estimator performs quite well despite the drifting, unscaled VO.

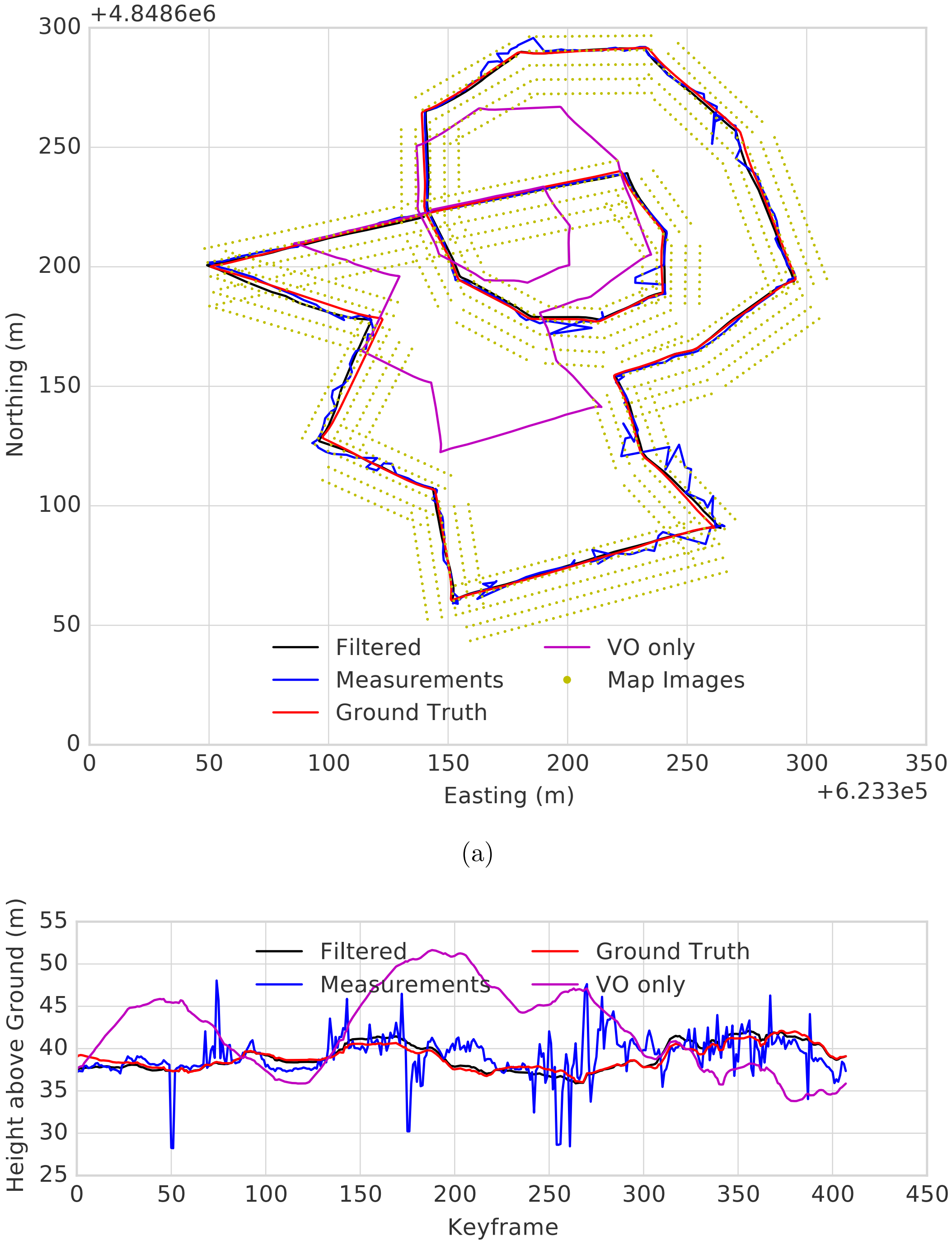


(a)



(b)

Figure 3.19: Position estimates for our 1132 m evening flight which shows our worst localization performance. The path starts at the intersection and spirals outward clockwise. Note that we show all image registration results here, even if they were classified as a failure. The evening flight had the largest number of image registration failures due to the large shadows that appear in the real images. The worse performance can clearly be seen by the large jumps in the measured positions. However, the image registrations were good enough to apply corrections and scale VO to result in a fairly accurate filtered estimate.



(a)

(b)

Figure 3.20: Position estimates for our 1132 m sunset flight. The path starts at the intersection and spirals outward clockwise. Note that we show all image registration results here, even if they were classified as a failure. While the image registrations had the 2nd lowest success rate due to the darker real images at sunset compared to the GE images, the overall estimation was still fairly accurate.

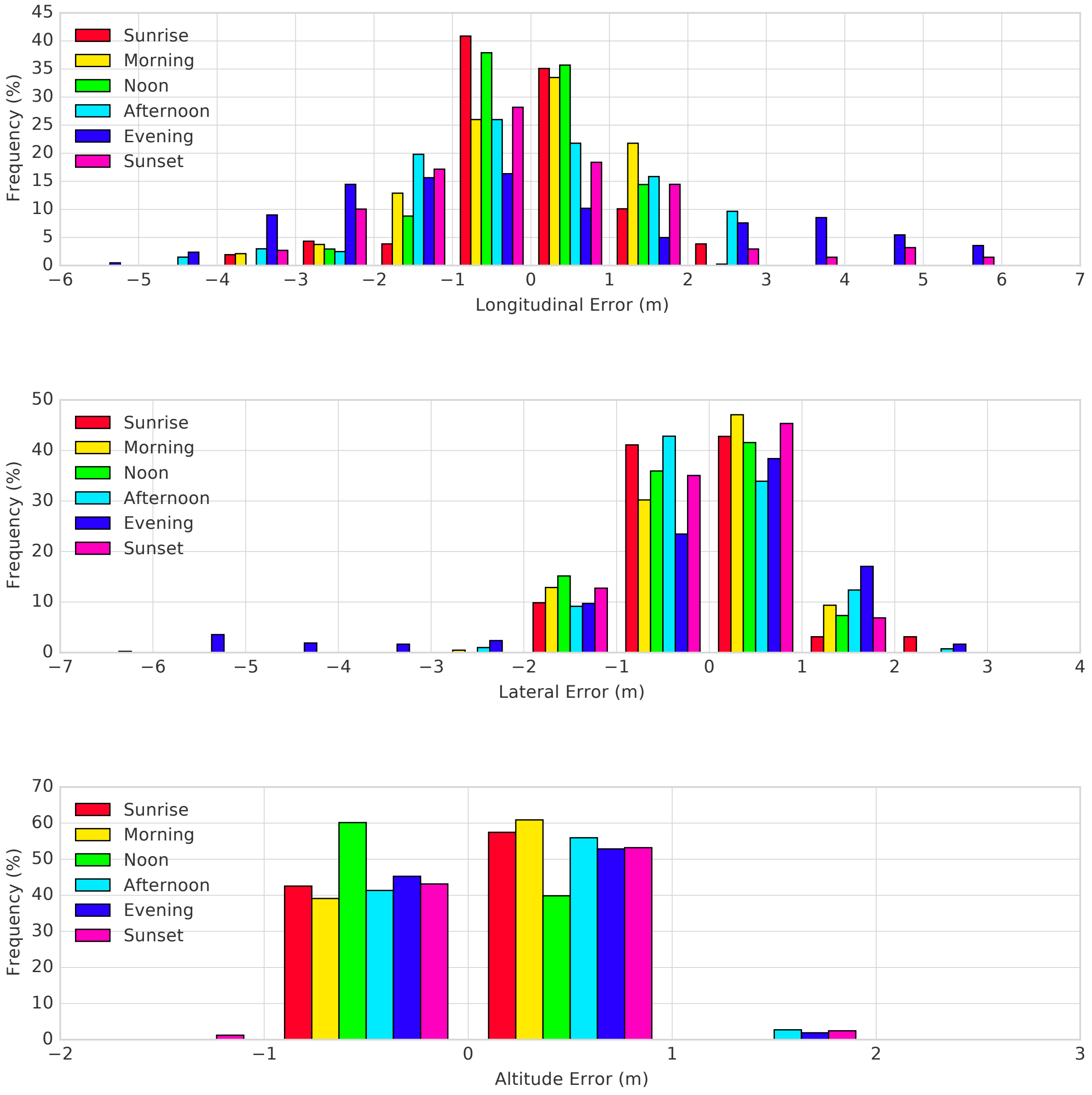


Figure 3.21: Histogram of filtered position errors for our sunrise to sunset flights. The majority of errors are less than a few metres.

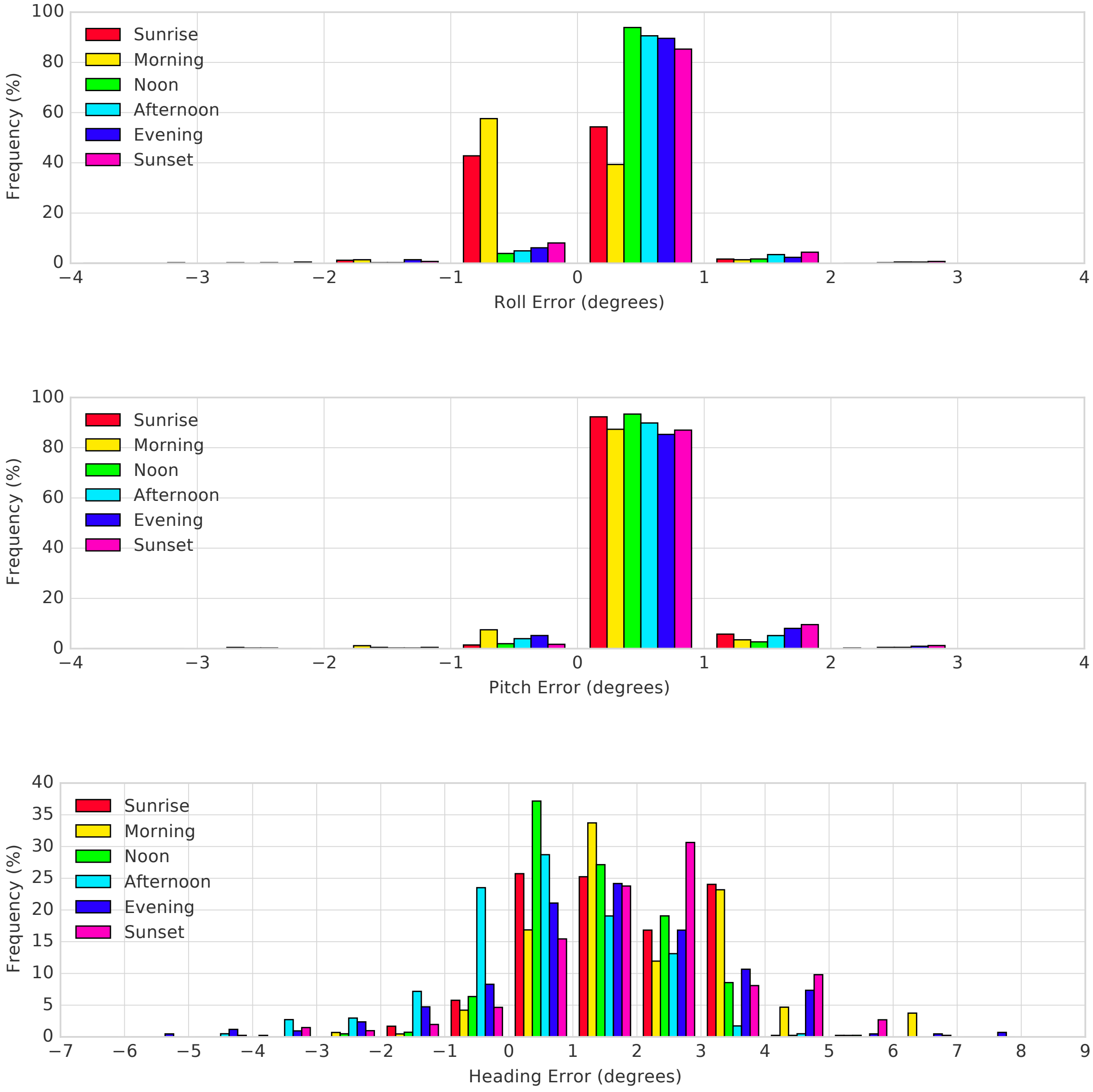


Figure 3.22: Histogram of filtered orientation errors for our sunrise to sunset flights. The majority of errors are less than a few degrees.

Chapter 4

Conclusion and Future Directions

4.1 Conclusion

The first contribution of this thesis is aiding in the development of a vision-based route-following system for the emergency return of a multirotor UAV in the event of GPS loss; this was developed using the VT&R system as a foundation. The primary contribution was assisting in the software development, hardware setup, and experimental validation over months of outdoor flight tests at UTIAS; Koffler Scientific Reserve; Suffield, Alberta; and downtown Montreal. An emphasis was placed on contributing to the visual localization algorithm and gimballed camera pointing. Using a slower absolute angle gimbal controller with an orientation matching strategy enabled successful localizations at speeds up to 15 m/s on a 450 m path flown at 12 m AGL with return flights performed under GPS control. We also showed successful localizations despite increasing altitude errors with the mapping flights performed at 12 m AGL and returns at up to 18 m AGL. Finally, we demonstrated closed-loop vehicle control using a vision-based PID path-following controller on the same 450 m path at 12 m AGL with path-following errors equivalent to the onboard GPS controller. This work resulted in a journal publication [Warren et al., 2019].

The next contribution of this thesis is a modular gimbal controller that resides inside the VT&R software system (Chapter 2). The user is able to select an absolute angle command controller or an angular rate controller for a faster response. The user can additionally select a desired camera pointing strategy for the teach and repeat phases. In Chapter 2, we evaluated multiple gimbal pointing strategies including off-the-shelf passive stabilization, active stabilization, orientation matching to minimize the camera viewpoint orientation error, and centroid pointing to point at the centroid of previously observed landmarks. We highlighted the robustness a gimballed camera adds to visual localization by showing a statically-mounted frequently failing on a dynamic path while orientation matching was able to minimize the perspective errors to retain localization. We showed that active pointing is necessary; simply adding a gimbal with a passive control strategy can actually lead to localization failures. Finally, we showed our orientation matching and centroid pointing strategies enable successful localizations on a 170 m flight path with $6 - 25\,\mathrm{m}$ altitude variations despite large velocity discrepancies (teaching at 3 m/s and repeating at 9 m/s) and large path-following errors (up to 8 m) between the outbound and return flights. This work resulted in a best paper award [Patel et al., 2019].

Lastly, this thesis contributes a global pose estimation method for a multirotor UAV using a set of pre-rendered images from a 3D reconstruction of the Earth (Chapter 3). We use an information theoretic approach for dense image registration of real-world images with rendered georeferenced images. Using a MI-based cost instead of photometric allows accurate registration despite large appearance differences between the 3D reconstruction and the true world. On 7.1 km of flight data over multiple flights with altitudes as low as 36 m AGL, we achieved less than a few metres and few degrees global RMSE. We were able to consistently localize from sunset to sunrise during a 16 hour sunny summer day using a single database of pre-rendered images from Google Earth despite the dramatic changes in lighting in the real-world images. Our method is able to provide global position estimates with an accuracy on par with GPS. This work resulted in a journal paper submission [Patel et al., 2020].

4.2 Future Directions

For active camera pointing we considered fairly simple and intuitive pointing strategies for the repeat runs. Although these provide performance that is good enough for closed loop control at slower speeds, future work could explore more advanced pointing strategies. One idea is to point the camera to maximize the probabilty of acheiving a minimum number of inliers informed by an inlier model that is a function the angular offset from the camera to the landmark caused by a position offset between the teach and repeat paths. As this angle increases, the probability that the landmark will be matched via SURF features descreases. Future work could also explore pointing strategies for the teach phase to improve the visual map. One such strategy is an altitude-based pitch control to point the camera towards the horizon during low altitude flights and towards the ground during high altitude flights to maximize useful visual information.

The global pose estimation method presented here opens the door to many research and engineering opportunities. First, the estimation is currently running offline in a

Python implementation so the next step is to implement it in C++ and within the VT&R software system. We expect this will provide at least an order of magnitude speedup, and in combination with parellization, we strongly believe that real-time performance is possible. Furthermore, the integration with the VT&R software system would enable the pose estimates from the MI-based image registration to be used in the existing bundle adjustment. Second, the estimation would greatly benefit with proper uncertainty quantification. The uncertainty could then be used to constrain the search radius for map image selection and reject registrations outside the three standard deviation uncertainty bounds. Third, GE opens the door to perform path planning for good localizability in advance of flight. Images generated in GE along potential flight paths could be used to design a path that maximizes the chance of localization. Fourth, the performance of the estimator needs to be evaluated in more scenarios (lower altitudes, over mulitple seasons, in different locations, etc.) to compare its reliability with GPS. Finally, it needs to be integrated with a vision-based flight controller to show closed-loop vision-based autnomous navigation. Ultimately, we envision this technique will enable any time, any place flights.

Bibliography

[1] Markus Achtelik, Michael Achtelik, Stephan Weiss, and Roland Siegwart. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3056–3063. IEEE, may 2011. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980343. URL `http://ieeexplore.ieee.org/document/5980343/`.

[2] Pratik Agarwal and Luciano Spinello. Metric Localization using Google Street View. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3111–3118, 2015. ISBN 9781479999934.

[3] Sean Anderson and Timothy D. Barfoot. Full STEAM ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on SE(3). *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 157–164, 2015. ISSN 21530866. doi: 10.1109/IROS.2015.7353368.

[4] Abraham Bachrach, Ruijie He, and Nicholas Roy. Autonomous Flight in Unknown Indoor Environments. *International Journal of Micro Air Vehicles*, 1(4):126, 2009. ISSN 10709932. doi: 10.1109/MRA.2005.1411416. URL `http://dspace.mit.edu/bitstream/handle/1721.1/54222/599812275.pdf?sequence=1`.

[5] Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. RANGE-Robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28 (5):644–666, 2011. ISSN 15564959. doi: 10.1002/rob.20400.

[6] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. ISSN 10773142. doi: 10.1016/j.cviu.2007.09.014.

[8] Michael Blösch, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Vision based MAV navigation in unknown and unstructured environments. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 21–28, 2010. ISBN 9781424450381. doi: 10.1109/ROBOT.2010.5509920.

[9] Alexandre Borowczyk, Duc-Tien Nguyen, André Phu-Van Nguyen, Dang Quang Nguyen, David Saussié, and Jerome Le Ny. Autonomous Landing of a Quadcopter on a High-Speed Ground Vehicle. *Journal of Guidance, Control, and Dynamics*, 2017. ISSN 0731-5090. doi: 10.2514/1.G002703.

[10] Christopher L. Choi, Jason Rebello, Leonid Koppel, Pranav Ganti, Arun Das, and Steven L. Waslander. Encoderless gimbal calibration of dynamic multi-camera clusters. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2126–2133, 2018. doi: 10.1109/ICRA.2018.8462920. URL `https://doi.org/10.1109/ICRA.2018.8462920`.

[11] Gianpaolo Conte and Patrick Doherty. An integrated UAV navigation system based on aerial image matching. In *IEEE Aerospace Conference Proceedings*, 2008. ISBN 1424414881. doi: 10.1109/AERO.2008.4526556.

[12] Paul Furgale and Timothy D Barfoot. Visual Teach and Repeat for Long Range Rover Autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010. doi: 10.1002/rob.20342. URL `http://dx.doi.org/10.1002/rob.20342`.

[13] Michael A. Goodrich, Bryan S. Morse, Damon Gerhardt, Joseph L. Cooper, Morgan Quigley, Julie A. Adams, and Curtis Humphrey. Supporting wilderness search and rescue using a camera-equipped mini UAV. *Journal of Field Robotics*, 25(1-2):89–110, 2008. ISSN 15564959. doi: 10.1002/rob.20226.

[14] Sungsik Huh, David Hyunchul Shim, and Jonghyuk Kim. Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3158–3163, 2013. ISBN 9781467363587. doi: 10.1109/IROS.2013.6696805.

[15] Dong Ki Kim and Matthew R. Walter. Satellite image-based localization via learned embeddings. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2073–2080, 2017. ISBN 9781509046331. doi: 10.1109/ICRA.2017.7989239.

[16] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*, 2007. ISBN 9781424417506. doi: 10.1109/ISMAR.2007.4538852.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] Tsung-yi Lin, James Hays, and Cornell Tech. Learning Deep Representations for Ground-to-Aerial Geolocalization. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[19] András L. Majdik, Damiano Verda, Yves Albers-Schoenberg, and Davide Scaramuzza. Air-ground Matching: Appearance-based GPS-denied Urban Localization of Micro Aerial Vehicles. *Journal of Field Robotics*, 2015. ISSN 15564967. doi: 10.1002/rob.21585.

[20] Geoffrey Pascoe, Will Maddern, and Paul Newman. Robust Direct Visual Localisation using Normalised Information Distance. In *British Machine Vision Conference (BMVC)*, pages 70.1–70.13, Swansea, Wales, 2015. doi: 10.5244/c.29.70.

[21] Geoffrey Pascoe, Will Maddern, Alexander D. Stewart, and Paul Newman. FARLAP: Fast robust localisation using appearance priors. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6366–6373, 2015. ISBN 9781479969234. doi: 10.1109/ICRA.2015.7140093.

[22] Bhavit Patel, Michael Warren, and Angela P. Schoellig. Point me in the right direction: Improving visual localization on UAVs with active gimballed camera pointing. In *Proc. of the Conference on Computer and Robot Vision (CRV)*, 2019.

[23] Bhavit Patel, Timothy D. Barfoot, and Angela P. Schoellig. Visual localization with google earth images for robust global pose estimation of UAVs. *IEEE Robotics and Automation Letters*, 2020. Submitted.

[24] Michael Paton, Kirk Mactavish, Michael Warren, and Timothy D. Barfoot. Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. *IEEE International Conference on Intelligent Robots and Systems*, 2016-Novem:1918–1925, 2016. ISSN 21530866. doi: 10.1109/IROS.2016.7759303.

[25] Andreas Pfrunder, Angela P. Schoellig, and Timothy D. Barfoot. A proof-of-concept demonstration of visual teach and repeat on a quadrocopter using an altitude sensor and a monocular camera. In *Proc. of the Conference on Computer and Robot Vision (CRV)*, pages 238–245, 2014. ISBN 9781479943388. doi: 10.1109/CRV.2014.40.

[26] Nicholas Playle. *Improving the Performance of Monocular Visual Simultaneous Localisation and Mapping through the use of a Gimballed Camera*. MASc, University of Toronto, 2015.

[27] Morgan Quigley, Michael A. Goodrich, Stephen Griffiths, Andrew Eldredge, and Randal W. Beard. Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2005, pages 2600–2606, 2005. ISBN 078038914X. doi: 10.1109/ROBOT.2005.1570505.

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[29] C S Sharp, O Shakernia, and S S Sastry. A Vision System for Landing an Unmanned Aerial Vehicle. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1720–1727, 2001. ISBN 0780364759. doi: 10.1109/ROBOT.2001.932859.

[30] Shaojie Shen, Yash Mulgaonkar, Nathan Michael, and Vijay Kumar. Vision-based state estimation for autonomous rotorcraft MAVs in complex environments. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013. ISBN 9781467356411. doi: 10.1109/ICRA.2013.6630808.

[31] Shaojie Shen, Nathan Michael, and Vijay Kumar. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2015-June, pages 5303–5310, 2015. ISBN 978-1-4799-6923-4. doi: 10.1109/ICRA.2015.7139939.

[32] Akshay Shetty and Grace Xingxin Gao. UAV Pose Estimation using Cross-view Geolocalization with Satellite Imagery. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Per Skoglar. Modelling and control of IR / EO-gimbal for UAV surveillance applications. *Electrical Engineering, Linköping Institute of Technology, Linköping, Sweden*, 2002. URL `http://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:Modelling+and+control+of+IR/EO-gimbal+for+UAV+surveillance+applications{\#}0`.

[35] Per Skoglar, Umut Orguner, David T. Ornqvist, and Fredrik Gustafsson. Road Target Search and Tracking with Gimballed Vision Sensor on an Unmanned Aerial Vehicle. *Remote Sensing*, 4(7):2076–2111, 2012. ISSN 20724292. doi: 10.3390/rs4072076.

[36] Kil-ho Son, Youngbae Hwang, and In-so Kweon. UAV global pose estimation by matching forward-looking aerial images with satellite images. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3880–3887, 2009. ISBN 9781424438044.

[37] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of ConvNet features for place recognition. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4297–4304, 2015. ISBN 9781479999941. doi: 10.1109/IROS.2015.7353986.

[38] Julian Surber, Lucas Teixeira, and Margarita Chli. Robust visual-inertial localization with weak GPS priors for repetitive UAV flights. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6300–6306, 2017. ISBN 9781509046331. doi: 10.1109/ICRA.2017.7989745.

[39] Amirmasoud Ghasemi Toudeshki, Faraz Shamshirdar, and Richard Vaughan. UAV Visual Teach and Repeat Using Only Semantic Object Features. 2018. URL `http://arxiv.org/abs/1801.07899`.

[40] Zhanglong Wang, Haoping She, and Weiyong Si. Autonomous landing of multi-rotors UAV with monocular gimbaled camera on moving vehicle. *IEEE International Conference on Control and Automation, ICCA*, pages 408–412, 2017. ISSN 19483457. doi: 10.1109/ICCA.2017.8003095.

[41] Michael Warren, Michael Paton, Kirk MacTavish, Angela P. Schoellig, and Tim D. Barfoot. Towards visual teach & repeat for GPS-denied flight of a fixed-wing UAV. In *Proc. of the 11th Conference on Field and Service Robotics (FSR)*, pages 481–498, 2017.

[42] Michael Warren, Angela P. Schoellig, and Tim D. Barfoot. Level-headed: gimbal-stabilised visual teach & repeat for improved high-speed path-following. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[43] Michael Warren, Melissa Greeff, Bhavit Patel, Jack Collier, Angela P. Schoellig, and Timothy D. Barfoot. There's no place like home: Visual teach and repeat for emergency return of multirotor UAVs during GPS failure. *IEEE Robotics and Automation Letters*, 4(1):161–168, 2019. doi: 10.1109/LRA.2018.2883408.

[44] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011. ISSN 15564959. doi: 10.1002/rob.20412.

[45] Stephan Weiss, Markus W. Achtelik, Simon Lynen, Michael C. Achtelik, Laurent Kneip, Margarita Chli, and Roland Siegwart. Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium. *Journal of Field Robotics*, 30(5): 803–831, 2013. ISSN 14746670.

[46] Aurelien Yol, Bertrand Delabarre, Amaury Dame, and Jean-emile Dartois. Vision-based Absolute Localization for Unmanned Aerial Vehicles. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3429–3434, 2014. ISBN 9781479969340.