Point Me In The Right Direction: Improving Visual Localization on UAVs with Active Gimballed Camera Pointing

Bhavit Patel, Michael Warren, and Angela P. Schoellig
University of Toronto Institute for Aerospace Studies
Toronto, Ontario, Canada
{*bhavit.patel, michaelwarren, angela.schoellig*}*@robotics.utias.utoronto.ca*

***Abstract*—Robust autonomous navigation of multirotor UAVs in GPS-denied environments is critical to enable their safe operation in many applications such as surveillance and reconnaissance, inspection, and delivery services. In this paper, we use a gimballed stereo camera for localization and demonstrate how the localization performance and robustness can be improved by actively controlling the camera's viewpoint. For an autonomous route-following task based on a recorded map, multiple gimbal pointing strategies are compared: off-the-shelf passive stabilization, active stabilization, minimization of viewpoint orientation error, and pointing the camera optical axis at the centroid of previously observed landmarks. We demonstrate improved localization performance using an active gimbal-stabilized camera in multiple outdoor flight experiments on routes up to 315 m, and with 6-25 m altitude variations. Scenarios are shown where a static camera frequently fails to localize while a gimballed camera attenuates perspective errors to retain localization. We demonstrate that our orientation matching and centroid pointing strategies provide the best performance; enabling localization despite increasing velocity discrepancies between the map-generation flight and the live flight from 3-9 m/s, and 8 m path offsets.**

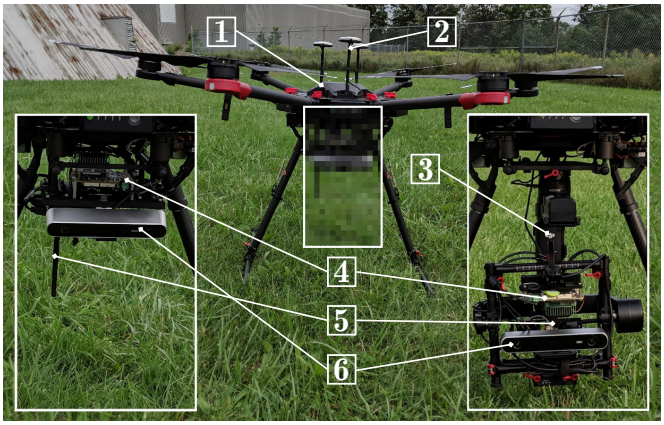


Figure 1: The hardware setup with a static camera (left) and gimballed camera (right) on a multirotor UAV: (1) DJI Matrice 600 Pro, (2) DJI A3 GPS module, (3) DJI Ronin-MX 3-axis gimbal, (4) NVIDIA Tegra TX2, (5) XBee Pro 900 MHz XSC S3B RF module, (6) StereoLabs ZED camera.

I. INTRODUCTION

The majority of Unmanned Aerial Vehicles (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 in the event of 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. There are many examples of vision-based flight using monocular [1]–[9] and stereo [10], [11] cameras. However, the majority use statically mounted cameras.

The goal of this work is to demonstrate the benefits of an actively controlled gimballed camera for visual localization on UAVs. In particular, we consider a visual teach and repeat scenario where a visual map is created during a human-piloted or autonomous GPS waypoint outbound flight and is used to safely fly back in the event of GPS loss by visually localizing against the map.

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) [12] 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 outbound and return flights unless the UAV follows an identical acceleration profile.

A 3-axis gimbal fully decouples the camera and vehicle orientations. Moreover, it 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.

In this paper, we use an active gimballed camera on a multirotor UAV in a similar manner as done in [13] for ground vehicles and in [11] 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 to handle path-following errors. Finally, we perform multiple outdoor flight experiments to i) highlight the robustness an active[1] gimballed camera adds over a static camera, ii) show that an off-the-shelf passively[2] 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.

II. RELATED WORK

The majority of vision-based autonomous navigation solutions for UAVs use static cameras [2]–[10]. Visual Simultaneous Localisation and Mapping (SLAM) techniques have been sucessfully demonstrated for GPS-denied environments in indoor [2], [3] and outdoor settings [4], [5], [10]. A proof-of-concept demonstration of Visual Teach and Repeat (VT&R) [14] on multirotor UAVs was shown in [6] followed by successful offline localization on a fixed-wing UAV [7]. Recently, there have also been demonstrations of similar teach-and-repeat style techniques for visual navigation using semantic objects [8], and offline map building [9].

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 [15]–[17], and search and rescue [18]. 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 [19], and 3-axis gimballed monocular cameras were leveraged for autonomous landing on moving platforms [20], [21]. Recent work demonstrates the integration of gimballed cameras with Visual-Inertial Odometry (VIO) [22] and visual SLAM [1]. Work in [1] performs a reactive viewpoint selection strategy by panning the camera to areas of high feature density with

[1]Active strategies 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).

[2]Passive strategies require no user control input.

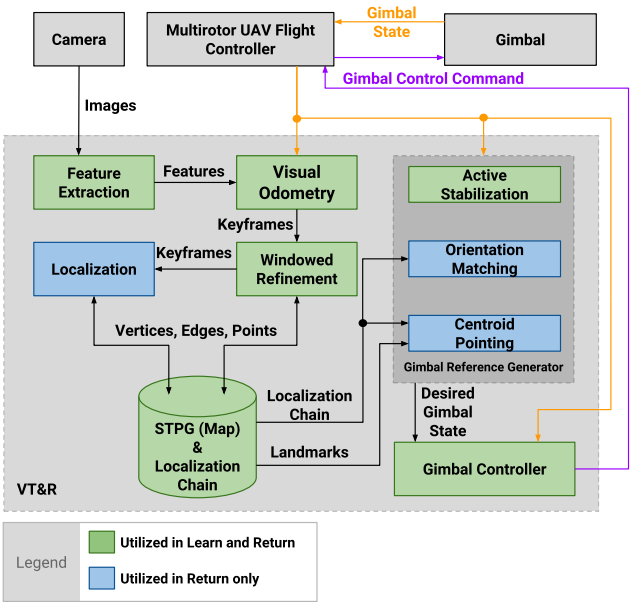


Figure 2: A simplified overview of the vision-based localization system with an active gimballed camera. During *learn*, the phase where the map is created, active stabilization can be performed while in *return*, the phase where the map is used for localization, any of the proposed gimbal pointing strategies can be selected.

the goal of improving localization accuracy of monocular visual SLAM using a two-axis gimbal. The most closely related work is our previous work demonstrating the use of VT&R as an emergency return system on multirotor UAVs [11] using the active gimballed camera implementation introduced in [13]. While work in [11] 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.

III. METHODOLOGY

VT&R is a route-following technique that enables long-range autonomous navigation without reliance on external positioning systems such as GPS [14]. While its development has largely focused on ground vehicles, we adapt it for multirotor UAVs to act as an emergency return system. During a human-piloted or autonomous GPS waypoint outbound flight, termed the *learn* phase, a visual map is generated using only a stereo camera and performing sparse feature-based Visual Odometry (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.

Figure 2 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 *learn* and *return* phases with faster response times. The new implementation also provides the ability to select different pointing methods to use in each phase.

A. VT&R Overview

During an outbound *learn* 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 angular positions 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 Fig. 3). Each vertex α stores the 3D positions of landmarks with associated covariances observed by the camera, $\{\mathbf{p}^\alpha, \boldsymbol{\sigma}^\alpha\}$, and the non-static vehicle-to-sensor transform, $\mathbf{T}_{sv}^\alpha$ (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 6 Degree of Freedom (DoF) $SE(3)$ transformation with uncertainty, $\{\mathbf{T}_{\alpha,\alpha-1}, \boldsymbol{\Sigma}_{\alpha,\alpha-1}\}$. The set of linked temporal edges represent the locally consistent path. During *learn*, this path is marked as privileged.

During an inbound *return* flight, the same visual odometry and map building as *learn* 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 (𝔩), twig (𝔴), branch (𝔟), trunk (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. With every successful VO update, the estimated transform from the trunk to the leaf, $\check{\mathbf{T}}_{\mathfrak{lt}} = \check{\mathbf{T}}_{hc} = \mathbf{T}_{hg}\mathbf{T}_{gd}\mathbf{T}_{dc}$ in Fig. 3, 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 Fig. 3 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

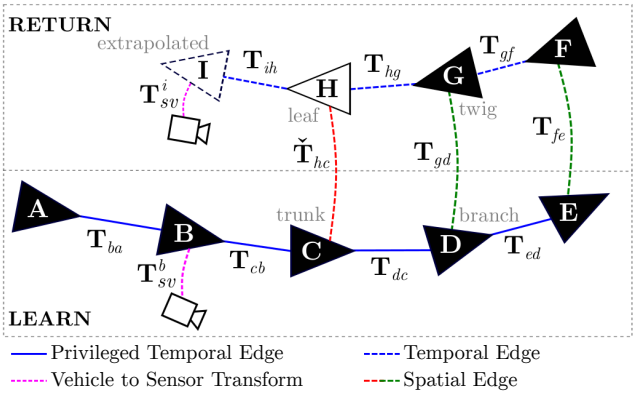


Figure 3: Depiction of an STPG with a single privileged experience. Active vision pointing strategies use the transforms from the live (*return*) path to the privileged (*learn*) path for gimbal control (e.g., $\mathbf{T}_{gd}$ which is the 6DoF transformation from vertex D to G). Some estimated transforms and uncertainties are omitted for clarity.

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. The spatial transform is optimized in a final stage. The localization chain is updated with the new spatial transform: $\mathbf{T}_{\mathfrak{wb}} \leftarrow \mathbf{T}_{hc}$.

We refer the reader to our previous work for a more detailed explanation of the the gimballed VO [13], localization [23], and the multirotor UAV emergency return adaptation with closed-loop vehicle control [11].

B. 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 $\boldsymbol{\Phi} = [\phi\ \theta\ \psi]^\mathsf{T}$ be the roll, pitch, and yaw angular positions of the gimbal, respectively. The angular velocity commands are computed as

$$\mathbf{u} = \mathbf{K}\left(\boldsymbol{\Phi}_d - \boldsymbol{\Phi}\right), \tag{1}$$

where $\mathbf{K}$ is a 3×3 matrix with proportional gains k_ϕ, k_θ, and k_ψ on the diagonal and zeros for the off-diagonals. The gimbal controller is run at 10 Hz to match the update rate of the gimbal state and VO.

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.

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) 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 *learn* phase, but we also test its use in the *return* phase for a full comparison.

3) Orientation Matching: The goal of orientation matching is to minimize the camera's viewpoint orientation error during *return*. 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 Simultaneous Trajectory Estimation And Mapping (STEAM) engine [24]. We denote the extrapolated pose as $\mathfrak{l}'$ with its associated trunk as $\mathfrak{t}'$ (vertex I and B, respectively, in Fig. 3). The pose of the camera at $\mathfrak{t}'$ with respect to the pose of the camera at $\mathfrak{l}'$ is given by:

$$\tilde{\mathbf{T}}_{\mathfrak{l}'\mathfrak{t}'} = \tilde{\mathbf{T}}_{ib} = \mathbf{T}_{sv}^{i}\mathbf{T}_{ih}\check{\mathbf{T}}_{hc}\mathbf{T}_{cb}\mathbf{T}_{sv}^{b\ -1}, \tag{2}$$

where $\tilde{\mathbf{T}}$ refers to transforms between the sensor (camera) frames, $\mathbf{T}_{ih}$ is obtained from extrapolation, and $\check{\mathbf{T}}_{hc} \leftarrow \mathbf{T}_{hg}\mathbf{T}_{gd}\mathbf{T}_{dc}$. 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., $\mathbf{T}_{sv}^{i} \leftarrow \mathbf{T}_{sv}^{h}$). The camera's viewpoint orientation error is extracted from $\tilde{\mathbf{T}}_{ib}$ to compute the desired gimbal angular position, $\mathbf{\Phi}_d$.

4) Centroid Pointing: Pointing the camera at the centroid of previously observed 3D landmarks accounts for vehicle path-following errors during *return*. 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, $\mathbf{T}_{\mathfrak{l}'\mathfrak{t}'} = \mathbf{T}_{ib}$. 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:

$$\hat{\mathbf{u}}_{\mathfrak{t}'\mathfrak{n}}^{\mathfrak{l}'} = \mathbf{C}_{\mathfrak{l}'\mathfrak{t}'}\frac{\mathbf{r}_{\mathfrak{t}'\mathfrak{n}}}{\|\mathbf{r}_{\mathfrak{t}'\mathfrak{n}}\|_2}, \tag{3}$$

where $\mathbf{r}_{\mathfrak{t}'\mathfrak{n}}$ 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 covariance from the pose covariance matrix $\mathbf{\Sigma}_{\mathfrak{l}'\mathfrak{t}'}$. The uncertainty along the path is given by

$$\boldsymbol{\sigma}_{\hat{\mathfrak{u}}} = \sqrt{\hat{\mathbf{u}}_{\mathfrak{t}'\mathfrak{n}}^{\mathfrak{l}'\ \mathsf{T}}\mathbf{\Sigma}_r\hat{\mathbf{u}}_{\mathfrak{t}'\mathfrak{n}}^{\mathfrak{l}'}}. \tag{4}$$

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, $\mathfrak{t}'$, using the privileged temporal transforms. The centroid of these landmarks is further transformed into the sensor frame at the extrapolated leaf, $\mathfrak{l}'$. Let $\tilde{\mathbf{p}}_j^{\alpha}$ be the jth landmark in the sensor frame of vertex $\alpha \in S$. Using the extrapolated leaf and trunk vertex in Fig. 3, the centroid in the sensor frame at the extrapolated leaf, denoted $\tilde{\mathbf{c}}$ is given by

$$\tilde{\mathbf{c}} = \tilde{\mathbf{T}}_{ib}\frac{\sum\limits_{\alpha\in S}\sum_{j=1}^{n_\alpha}\mathbf{T}_{sv}^{b}\mathbf{T}_{b\alpha}\mathbf{T}_{sv}^{\alpha\ -1}\mathbf{p}_j^{\alpha}}{\sum\limits_{\alpha\in S} n_\alpha}, \tag{5}$$

where n_α is the number of landmarks at vertex α and $\tilde{\mathbf{T}}_{ib}$ is computed by (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.

IV. 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 Fig. 4. 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.

Fig. 1 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 onboard by an NVIDIA Tegra TX2. A StereoLabs ZED camera is connected to the onboard computer to provide 672×376 grayscale stereo images. A 900 MHz XBee low-bandwidth,

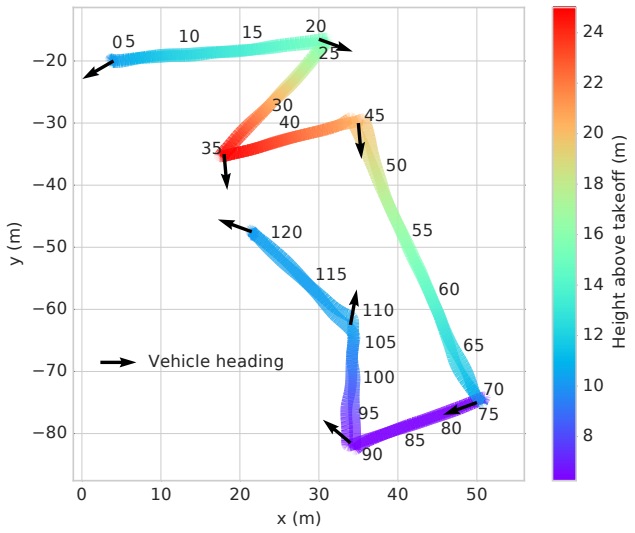


Figure 4: The dynamic path used for our gimbal pointing strategy comparisons which shows the height above takeoff, vehicle heading, and location of privileged vertices. Only the heading at each GPS waypoint and every fifth privileged vertex are shown for clarity. The vehicle heading rotates in the shortest 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 *return*).

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).

A. Gimballed Camera Robustness

The performance of a static and gimballed camera on a simple 315 m path at 15 m altitude learned at 3 m/s and returned at 9 m/s is shown in Fig. 5. 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. Fig. 6 shows a zigzag pattern flight path highlighted 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

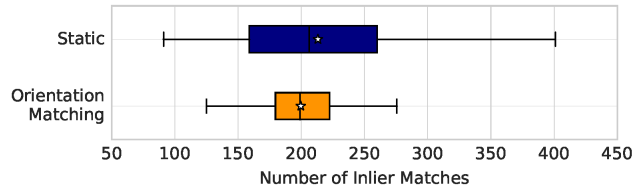


Figure 5: For a simple path with few dynamic motions, a static camera localizes even when returning at a faster speed (from 3 m/s outbound to 9 m/s target inbound speed). However, a gimbal reduces the variance in localization inliers by maintaining similar perspective.

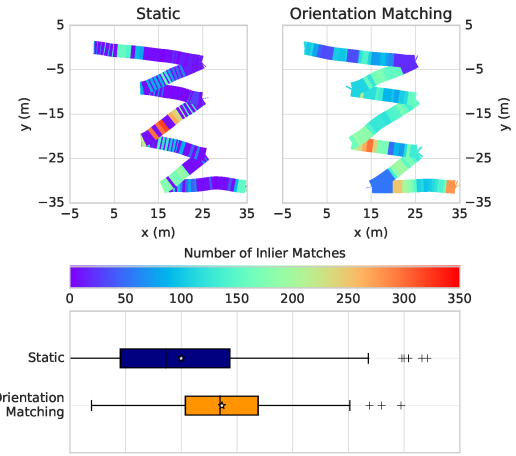


Figure 6: For highly dynamic paths, the static camera system has trouble localizing due to large viewpoint changes. The active gimballed camera system minimizes perspective error during the dynamic motions to improve localization performance with a 37% increase in the mean number of inliers.

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. The gimbal with active camera pointing increases the mean number of inliers by 37% over a static camera.

B. 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 *learn* flights conducted at 3 m/s. Fig. 4 shows the altitude-varying 170 m flight path used for these tests. The CDF of the localization uncertainties is shown in Fig. 7 while Fig. 8 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

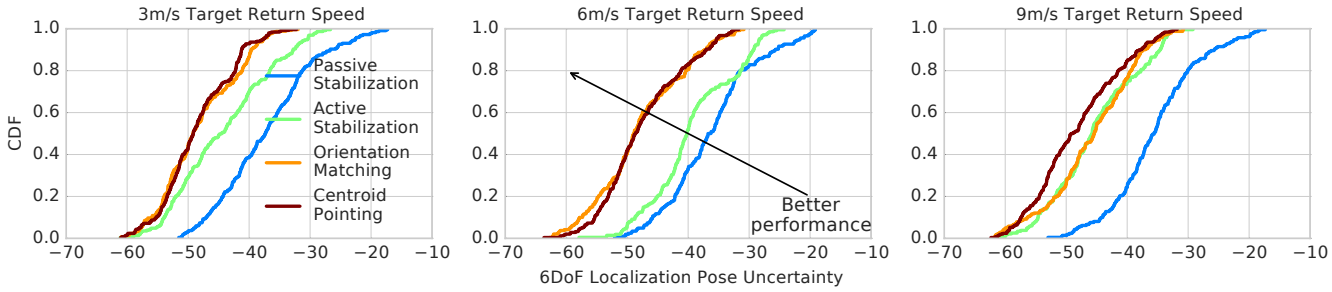


Figure 7: The CDF of the localization pose uncertainties show that active pointing strategies result in more confident localization estimates with centroid pointing showing slightly more confidence than orientation matching. The localization uncertainty is computed as the log determinant of the 6DoF spatial pose uncertainty matrix for each localization update (i.e., $\mathbf{\Sigma}_{t,t}$).

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 *learn* and *return* paths can be reduced using active pointing strategies. Fig. 9 shows the root mean square (RMS) 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., $\mathbf{T}_{gd}$ and $\tilde{\mathbf{T}}_{gd}$ in Fig. 3). As the velocity discrepancy between *learn* and *return* flight 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 between *learn* and *return*, 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. Active stabilization removes the yaw lag but does not account for vehicle yaw error between *learn* and *return* 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 RMS orientation error from the vehicle to camera frame for 3 m/s, 6 m/s, and 9 m/s return flights, respectively.

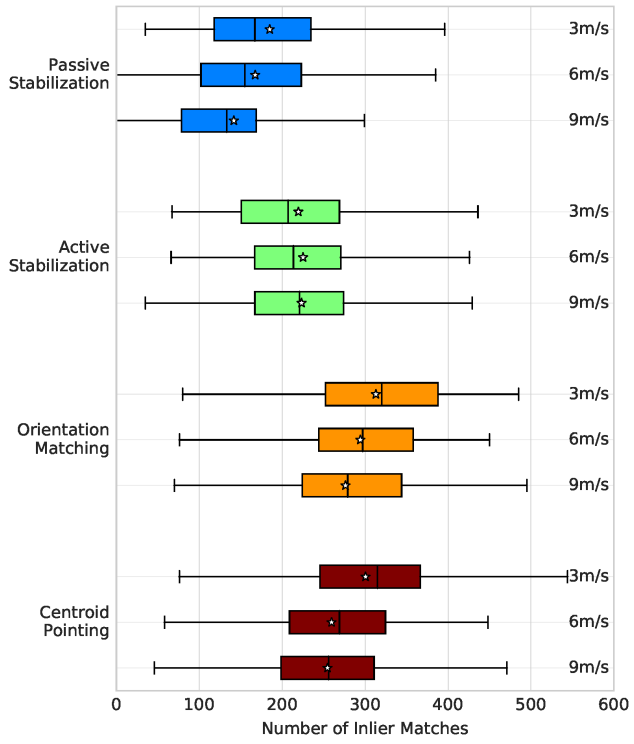


Figure 8: Active gimbal control prevents localization failures despite increasing speed discrepancies between *learn* and *return* flights. Incorporating visual information in the pointing strategy results in better localization performance.

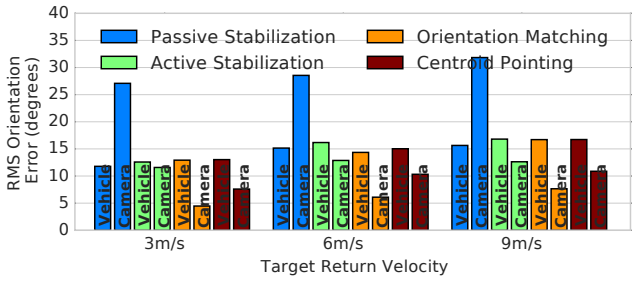


Figure 9: A gimballed camera with active pointing strategies reduces the camera perspective error that result from vehicle orientation errors between *learn* and *return* flights. The simple act of adding a gimbal is not enough as we see an off-the-shelf passive stabilization strategy actually increases the camera perspective error. Centroid pointing does not attempt to minimize viewpoint orientation error but it is compared for completeness.

C. Handling Cross-track Errors

In this experiment, large lateral and vertical cross-track errors are added to the return path to evaluate the localization performance of orientation matching and centroid pointing (see Fig. 10). Intuitively, a centroid pointing strategy is more suitable for situations with large path-following errors since it attempts to compensate for the translational 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 translational offsets. Along segment 2 and parts of 3, the translational offsets are large enough to reduce landmark visibility when orientation matching but small enough to be compensated by centroid pointing. Fig. 11 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 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.

V. CONCLUSIONS

In this paper, we demonstrated improved visual localization performance using an active gimbal-stabilized camera within a VT&R framework on multirotor UAVs. 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).

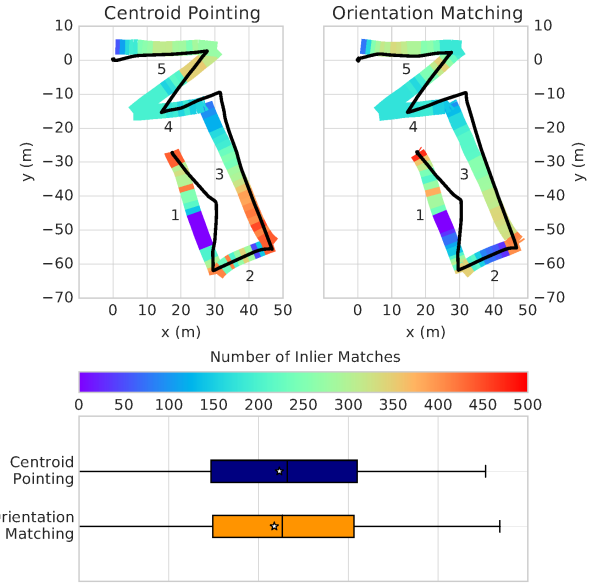
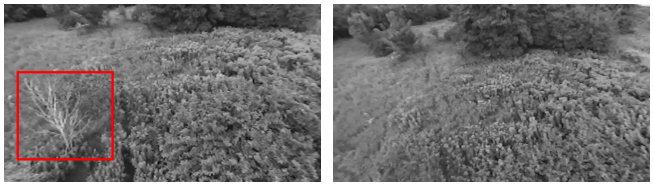
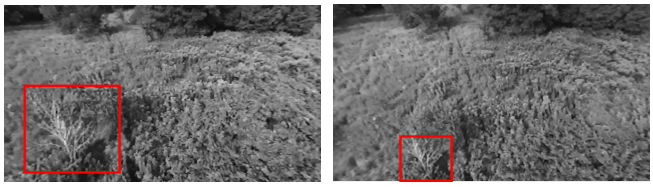


Figure 10: While the overall performance is similar, centroid pointing shows an advantage along segment 2 and parts of 3. The black line shows the outbound *learn* path while the inlier highlights are centered on the *return* path.



(a) Map and return images for an orientation matching run



(b) Map and return images for a centroid pointing run

Figure 11: Comparison of orientation matching and centroid pointing return views (right) on segment 2. The spatially nearest images captured during the learn 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.

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 outbound and return flights.

REFERENCES

[1] N. Playle, "Improving the Performance of Monocular Visual Simultaneous Localisation and Mapping Through the Use of a Gimballed Camera," Master's thesis, University of Toronto, 2015.

[2] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision Based MAV Navigation in Unknown and Unstructured Environments," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 21–28, 2010.

[3] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based Navigation for Autonomous Micro Helicopters in GPS-denied Environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[4] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3056–3063, may 2011.

[5] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803–831, 2013.

[6] A. Pfrunder, A. P. Schoellig, and T. 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*, pp. 238–245, 2014.

[7] M. Warren, M. Paton, K. MacTavish, A. Schoellig, and T. Barfoot, "Towards Visual Teach & Repeat for GPS-Denied Flight of a Fixed-Wing UAV," *Field and Service Robotics*, pp. 481–498, 2017.

[8] A. G. Toudeshki, F. Shamshirdar, and R. Vaughan, "UAV Visual Teach and Repeat Using Only Semantic Object Features," *arXiv preprint arXiv:1801.07899*, 2018.

[9] J. Surber, L. Teixeira, and M. Chli, "Robust Visual-Inertial Localization with Weak GPS Priors for Repetitive UAV Flights," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6300–6306, 2017.

[10] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based State Estimation for Autonomous Rotorcraft MAVs in Complex Environments," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[11] M. Warren, M. Greeff, B. Patel, J. Collier, A. P. Schoellig, and T. 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*, vol. 4, no. 1, pp. 161–168, Jan 2019.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[13] M. Warren, A. P. Schoellig, and T. D. Barfoot, "Level-Headed: Evaluating Gimbal-Stabilised Visual Teach and Repeat for Improved Localisation Performance," *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2018.

[14] P. Furgale and T. D. Barfoot, "Visual Teach and Repeat for Long Range Rover Autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.

[15] M. Quigley, M. A. Goodrich, S. Griffiths, A. Eldredge, and R. W. Beard, "Target Acquisition, Localization, and Surveillance Using a Fixed-Wing Mini-UAV and Gimbaled Camera," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2005, pp. 2600–2606, 2005.

[16] P. Skoglar, "Modelling and Control of IR / EO-gimbal for UAV Surveillance Applications," Ph.D. dissertation, Linköping Institute of Technology, Linköping, Sweden, 2002.

[17] P. Skoglar, U. Orguner, D. T. Ornqvist, and F. Gustafsson, "Road Target Search and Tracking with Gimballed Vision Sensor on an Unmanned Aerial Vehicle," *Remote Sensing*, vol. 4, no. 7, pp. 2076–2111, 2012.

[18] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting Wilderness Search and Rescue using a Camera-Equipped Mini UAV," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.

[19] C. S. Sharp, O. Shakernia, and S. S. Sastry, "A Vision System for Landing an Unmanned Aerial Vehicle," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1720–1727, 2001.

[20] A. Borowczyk, D.-T. Nguyen, A. P.-V. Nguyen, D. Q. Nguyen, D. Saussié, and J. Le Ny, "Autonomous Landing of a Quadcopter on a High-Speed Ground Vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2378–2385, 2017.

[21] Z. Wang, H. She, and W. Si, "Autonomous Landing of Multirotors UAV with Monocular Gimbaled Camera on Moving Vehicle," *Proc. of the IEEE International Conference on Control and Automation (ICCA)*, pp. 408–412, 2017.

[22] C. L. Choi, J. Rebello, L. Koppel, P. Ganti, A. Das, and S. L. Waslander, "Encoderless Gimbal Calibration of Dynamic Multi-Camera Clusters," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[23] M. Paton, K. Mactavish, M. Warren, and T. D. Barfoot, "Bridging the Appearance Gap: Multi-Experience Localization for Long-Term Visual Teach and Repeat," *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[24] S. Anderson and T. D. Barfoot, "Full STEAM Ahead: Exactly Sparse Gaussian Process Regression for Batch Continuous-Time Trajectory Estimation on SE(3)," *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015.