AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-2201

Improving UAV-Based Target Geolocation Accuracy through Automatic Camera Parameter Discovery

Andy Fabian*
Virginia Commonwealth University, Richmond, VA, 23284

Peter Truslow†
Virginia Commonwealth University, Richmond, VA, 23284

Robert Klenke‡
Virginia Commonwealth University, Richmond, VA, 23284

Precise geolocation of image features in UAV-generated imagery is simple in theory, yet notoriously difficult in practice. Adverse factors such as GPS position error, aircraft attitude uncertainty, camera mounting error, lens distortion, and errors in the time domain accumulate to degrade the accuracy of geolocation calculations. Of particular note are errors related to the rotation (pitch/roll/yaw) of the camera and aircraft, since their effect on ground coordinate determination is multiplied as the aircraft gains altitude and also trigonometrically increases as the angle between aircraft and target moves away from "straight down."

This paper presents a case study wherein a fixed-wing UAV and photography system is tuned for optimal geolocation accuracy. All known sources of error are quantified and minimized, requiring changes at the physical, communications, and software levels.

Finally, a technique is proposed and demonstrated for online determination of several key calibration values, eliminating the need for advance calibration. By using multiple detections of the same target, an optimization problem is created, where camera parameters (pitch, roll, and yaw offsets) can be discovered by maximizing the consistency in the set of detections. Results from this automatic calibration are compared to a similar manually-calibrated system.

I. Nomenclature

D	=	target detection (longitude, latitude)
M	=	mean position of a group of target detections (longitude, latitude)
P	=	"Group Precision," the average standard deviation of a group of detections (feet)
i	=	Iterator
p	=	Pitch (degrees)
r	=	Roll (degrees)
Y	=	Yaw (degrees)
E_p	=	Position Error (feet)
E_t	=	Timing Error (seconds)
V_g	=	Ground Speed (knots)

II. Introduction

Accurate target locations are central to many drone tasks. How can a drone deliver a package if it doesn't know where to drop it? How can a quadcopter come in to land on a recharging station if it only has a vague notion of where that station is? Is a security drone more useful if it can't tell which side of a football field someone is on or if it knows their exact location and can relay that information in real time?

*Ph.D. Student, Department of Electrical and Computer Engineering, Virginia Commonwealth University
†Ph.D. Student, Department of Electrical and Computer Engineering, Virginia Commonwealth University
‡Professor, Department of Electrical and Computer Engineering, Virginia Commonwealth University

In recent years, drones have also been a triumph of powerful algorithms over expensive hardware and meticulous construction. Target geolocation is a logical extension of this philosophy: if an inexpensive and carelessly mounted camera can be made to precisely identify the location of targets through advanced software, then this is a capability that can be endowed to all camera drones.

This paper will present a summary of the major factors affecting target geolocation accuracy and mitigation strategies for each:

- *Lens Distortion* degrades the accuracy of position estimates by warping light as it enters the image sensor. Extracting a geolocation from a photo requires shooting a ray through that photo at a particular pixel to find its intersection point with the ground. If the camera's pixels are not linearly distributed with respect to ray angle, this non-linearity must be understood and corrected.
- *Timing* is critical in a system of interconnected parts: connected computer systems incur latency during communication which can translate into measurement error. Additionally, time-critical events such as camera shutter firing must be measured accurately.
- *Multiple detections* of the same target can be used to gain additional information about that target's location.
- *Camera angle calibration* is critical, since an error of 1 degree on the camera mount can lead to a measurement error of more than 8 feet when flying at 500 feet AGL.

Finally, the techniques described within have been assembled into a working system, results of which are also included. We show that an off-the-shelf fixed-wing drone with an inexpensive Raspberry Pi camera can determine a target position to an accuracy of 2 feet.

III. Lens Distortion Removal

Lens distortion is non-linearity in the way that light enters the sensor. The most common types of lens distortion are Barrel Distortion and Pincushion distortion. Both of these types of geometric distortion produced by the lens are called radial distortion. The center of the image the lens projects onto the sensor serves as the reference point. The angular position of a point in the image is preserved, but the radial distance from the center has a non-linear relationship to the radius from the center of the subject. Barrel distortion causes the image to become compressed as the radius from the camera center increases. The effect is especially pronounced at the corners of the image. Pincushion causes the opposite, where the image is increasingly stretched at an increasing radius from the lens center. Barrel type distortion is generally observed in wide-angle lenses, while pincushion distortion is generally observed in narrow-angle or telephoto lenses. The effect of this distortion in this context is that when using lenses that exhibit barrel distortion, objects that are far from the center of the lens radially will be detected artificially close to the center of the image, while lenses with pincushion distortion will show objects far from the center of the image artificially far from the center.[1]

The Raspberry Pi camera's Image Signal Processor has a placeholder for lens compensation, but it is unused and empty. OpenCV has built-in lens distortion calibration. The calibration in OpenCV only needs to be performed once, and this calibration data may be common to all cameras with the same lens configuration. The calibration is performed using a chessboard pattern printed on a flat surface, where the size of the squares is consistent and known. Images of the pattern are collected at various distances and angles to cover the whole visual field of the camera. The images are passed through a camera calibration program included with OpenCV, which produces the constants necessary to correct for the distortion in images.

$$R_c = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \tag{1}$$

The equation for the radial distortion coefficient can be seen in Eq.(1) Where R_c is the coefficient of the radial error, and r is the ratio of the distance from the optical axis and the distance along the optical axis. The correction curve for radial distortion in the cameras used for testing can be found in Fig 1. As the center of the image is not necessarily the center of the lens distortion, lens calibration must be performed before the parameter discovery for camera angle calibration described later in this paper is performed.[2]

IV. Timing Concerns

A. Communication Frequency and Latency

A fixed-wing aircraft travelling at 50 knots covers 84 feet every second. Thus even small, tenth-of-second timing inaccuracies translate into significant position inaccuracies that can have deleterious effects on geolocation accuracy.

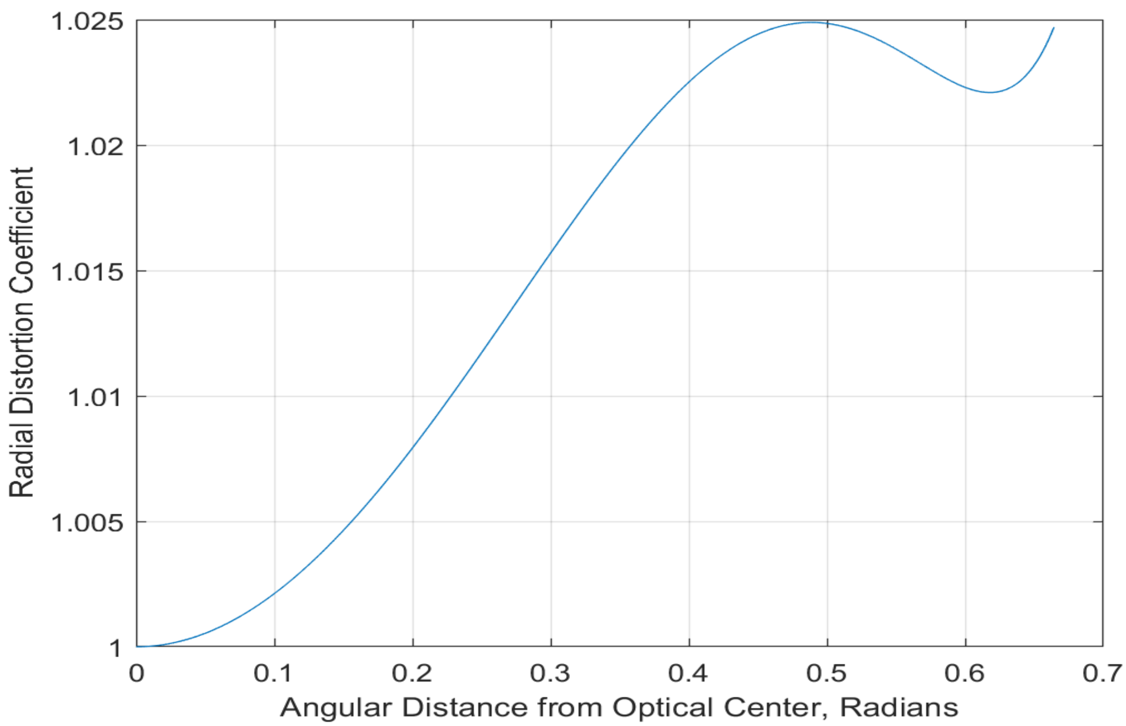

Fig. 1 Radial Distortion Correction Curve

As a gauge of this, timing inaccuracies can be translated into approximate positional inaccuracies via a simple scalar calculation, where E_p is the positional error in feet that results from E_t timing error (seconds) and V_g ground speed (knots).

$$E_p \approx V_g * E_t * 1.7 \tag{2}$$

This effect can be observed when using scheduled communication between system components. As an example, an aircraft flight controller might update its internal position estimate at 200 Hz, but only relay that information to a camera system at 10 or 20 Hz. In order to avoid positional errors related to intermittent position updates, the camera system must interpolate it's current position based on the last communicated position and velocity, and the time elapsed since then.

B. Image Capture Latency

Timing is also a consideration when determining the instant a photo is taken. Capturing the instant that the photo is requested (via shutter actuation or a software command) is typically not accurate, since there is a variable delay between that request and the image capture. This delay is dependant on the camera and its settings, such as auto-focus and auto-exposure. As an example, laboratory experiments determined that the capture latency of a Raspberry Pi camera is about 0.3 seconds.

The importance of this information lies in the determination of when to capture aircraft metadata. The aircraft state should be captured at the moment the image is recorded. Multiple approaches are available for this. Cameras with a firing indicator, such as a hot shoe attachment, can directly signal when the image is being captured. For other cameras, software can rely on latency estimation to capture aircraft metadata at the most likely moment for image capture.

V. Multi-Image Averaging

As one might expect, averaging multiple samples together can create a better estimate of target position. To make maximum use of this technique, one should understand the statistical output variance in relation to various factors. (Subscripts f and r refer to the aircraft's $forward$ and $right$ axis parallel to the ground.)

- Higher altitude results in higher variance. ($Var_{f,r} \propto A$)
- Higher ground speed results in higher variances, but only in the direction the airplane is moving. This is scaled by the system's timing accuracy, as discussed in section IV. ($Var_f \propto V_g K_t$)
- Large pitch and/or roll angles increase variance, as the camera's rays to the ground become longer and more parallel to the earth. ($Var_f \propto \frac{1}{cos(pitch)}, Var_r \propto \frac{1}{cos(roll)}$)
- The aircraft's pitch and roll stability each independently affect variance along those axes. ($Var_f \propto K_p, Var_r \propto K_r$)

Putting these together, the following formulas are obtained for the target location variance relative to the aircraft's forward and right directions. Var_i is the intrinsic variance of the camera system:

$$Var_f = Var_i * A * V_g K_t * \frac{1}{cos(pitch)} * K_p \tag{3}$$

$$Var_r = Var_i * A * \frac{1}{cos(roll)} * K_r \tag{4}$$

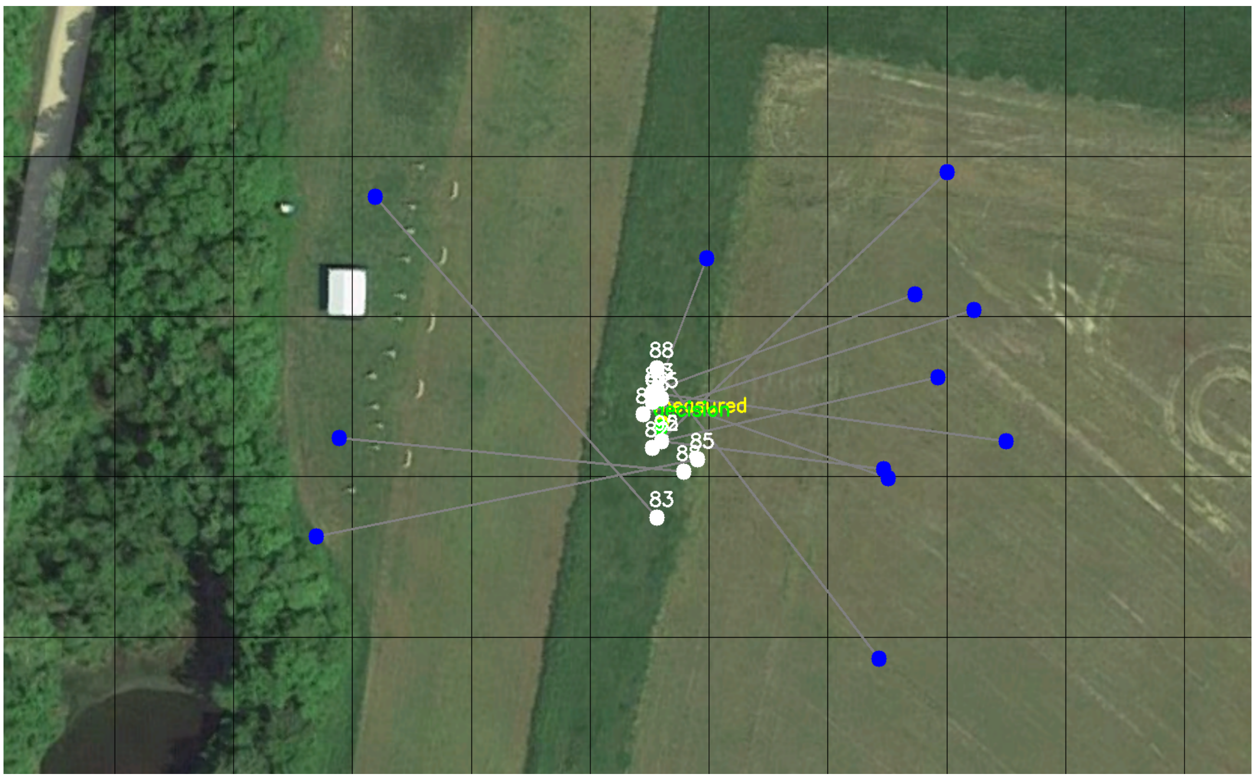

Fig. 2 Multiple Detection Averaging

In Fig. 2, the aircraft detects the same target multiple times. For each detection, the aircraft location is shown in blue, while the detection location estimate is shown in white. The average position (green) is very close to the measured position (yellow) - only 2 feet away!

VI. Parameter Discovery Via Aerial Imagery

In this section a method is presented for automatically and quickly determining the camera's pitch, roll, and yaw offset parameters, by creating an optimization problem from multiple images of a single target. This method relies on having an image processing system that can reliably identify the target and automatically extract the pixel coordinates of its centroid. It is further assumed that a geolocation pipeline is available and capable of running against stored imagery and aircraft metadata (aka "offline" execution) while varying the camera parameters.

Conceptually, we start by defining the "Mean Position" M as the mean coordinates (latitude and longitude) of all detections in the image set, possibly with outliers removed.

$$M = \frac{1}{n} \sum_{i=1}^{n} D_i \tag{5}$$

The term "Group Precision" P is defined to be the average standard deviation of all detections, relative to that mean. Group Precision is an indicator of how tightly grouped a set of detections are, though not of their accuracy in identifying the real-world coordinates of the target.

$$P = \frac{1}{n} \sqrt{\sum_{i=1}^{n} (D_i - M)^2} \tag{6}$$

Since detections D (and thus mean detection M) are functions of camera parameters, an optimization problem is formed: find the values for camera pitch, roll, and yaw offsets which minimize P.

$$P_{best} = \underset{p,r,y \in \mathbb{R}}{\arg\min} \left(\frac{1}{n} \sqrt{\sum_{i=1}^{n} (D_{i,p,r,y} - M_{p,r,y})^2} \right) \tag{7}$$

In algorithmic terms, this can be implemented as a brute-force search, which although slow, has the benefit of creating visualizations of the entire search space, helping the user to visually understand the trends in their system.

A faster approach, and the one used in our reference system, is to do a gradient descent. This is an iterative approach, wherein the system computes the partial derivatives of the calibration quality at successive positions, using each position and the gradient direction to determine the next position. The system continues until the gradient approaches zero. Using this algorithm, our current single-threaded Python code base can solve camera parameters from a 250 image set in 15 seconds.

Gradient descent is sensitive to the problem of local minimums, so care must be taken to ensure that data is suitable for gradient descent in the first place. This can be satisfied if all dimensions of the gradient are convex, and thus cannot have local minimums. From inspection of the brute-force calibration solver output (Fig. 5), this appears to be the case for this camera calibration problem.

Once the calibration parameters are discovered, they can be applied to all future imagery as long as the camera mounting, flight controller mounting, and software parameters related to orientation remain constant.

VII. Results

In testing this procedure, data was gathered by setting up a 4x4 foot red square target on the ground at a known location. This target was chosen to be compatible with image-processing software that can rapidly and reliably identify this target, including its pixel coordinates in captured images. Bench testing was also done for lens distortion calibration and system delay compensation.

A. Camera Angle Calibration

Two test flights were performed, with two sets of airplane, camera, and computing hardware. Each aircraft was a small fixed-wing autonomous airplane. The airplanes are based on a ReadyMadeRC SkyHunter [3] frame, running VCU Aries Autopilot [4], and a Raspberry Pi camera. The airplanes were made to fly over the target on a variety of headings for approximately 45 minutes at altitudes of 200, 300, and 400 feet above ground level (AGL). This resulted in the acquisition of approximately 200 images of the target from each aircraft, along with metadata files containing the aircraft state at the moment each image was captured. This state, which includes latitude, longitude, altitude, heading, pitch, and roll, is essential for geolocation.

Each set of images and metadata was fed into the calibration solver, which produced the calibrations shown in Fig. 3 and 4. The target geolocation algorithm was run before and after solving for the camera angles in order to measurement what improvement (if any) there was in the geolocation accuracy. Results are in Table 1, and showed a 2x or more improvement in accuracy.

Using this camera calibration data, all detections are fed into a target location algorithm, which performs a weighted average of detection coordinates as described in section V. This gives us the essential performance metric: improvement in geolocation accuracy.

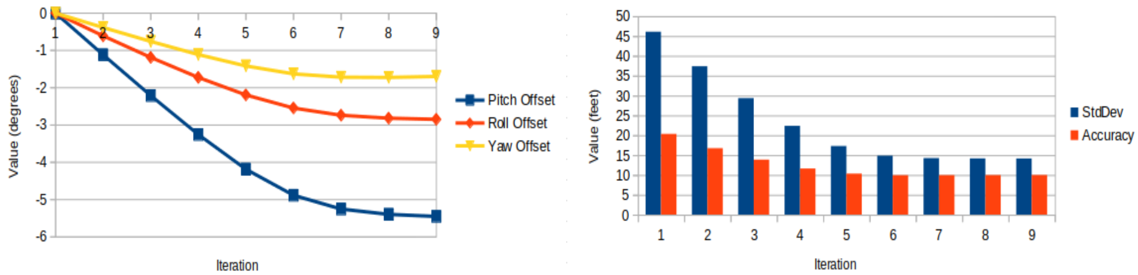

Fig. 3 Calibration Discovery via Gradient Descent - Airplane #5

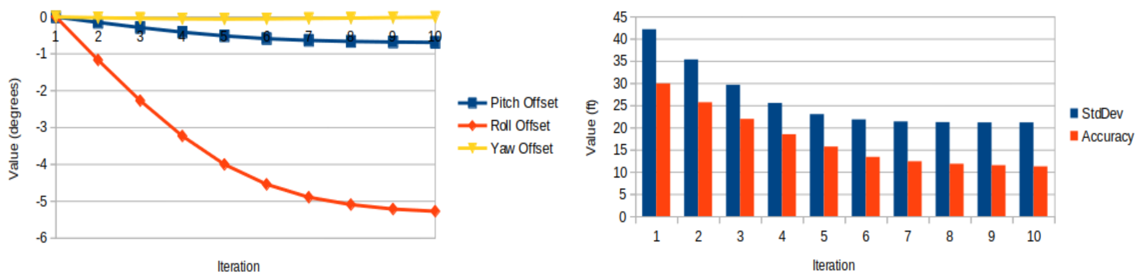

Fig. 4 Calibration Discovery via Gradient Descent - Airplane #6

B. Lens Distortion

A large set of images was fed into the gradient-descent algorithm twice, once with the lens distortion calibration enabled and once with it disabled. The results are below in Table 2. The results do not show a significant improvement in this case, as the camera used does not have significant distortion. In fact the results show that the calibration made the data somewhat worse, which may be due to some loss in quality when undistorting an image that had already been compressed. If the camera used were significantly distorted, it is likely that the distortion correction would have show a significant improvement, but this is dependent on the camera used.

C. Timing

To address the problem of camera firing latency, we chose to measure the average latency, and then program compensation for this into the camera control and metadata acquisition software. Using the Raspberry Pi and Pi Camera, we developed a utility that triggered the camera and simultaneously started displaying a timer to a monitor. By pointing the camera at the screen, the camera captures an image of the elapsed time (subject to rendering and monitor latency). Using this technique, we estimated the Pi camera's latency at 0.3s in our configuration.

Based on this measurement, our data acquisition software triggers the camera, and then in a separate thread, waits 0.3 seconds before capturing the current metadata state.

D. Multiple Detections

With regard to sample size, we've found that variety, not just quantity is important. Fig. 6 shows how an increasing sample size can increase the standard deviation of the individual samples. This is an indication that the initial set of samples was grouped together in a way not representative of the total population. Note that since samples typically come from aircraft flights, they are not randomly distributed, but are arranged in wandering linear patterns. We hypothesize that to optimize the ratio of accuracy to image count, one should design a flight pattern that creates a set of images from varied altitudes, headings, and alignment to the target (left/right/center of the aircraft). For best results, one should avoid including multiple images for similar locations and attitudes.

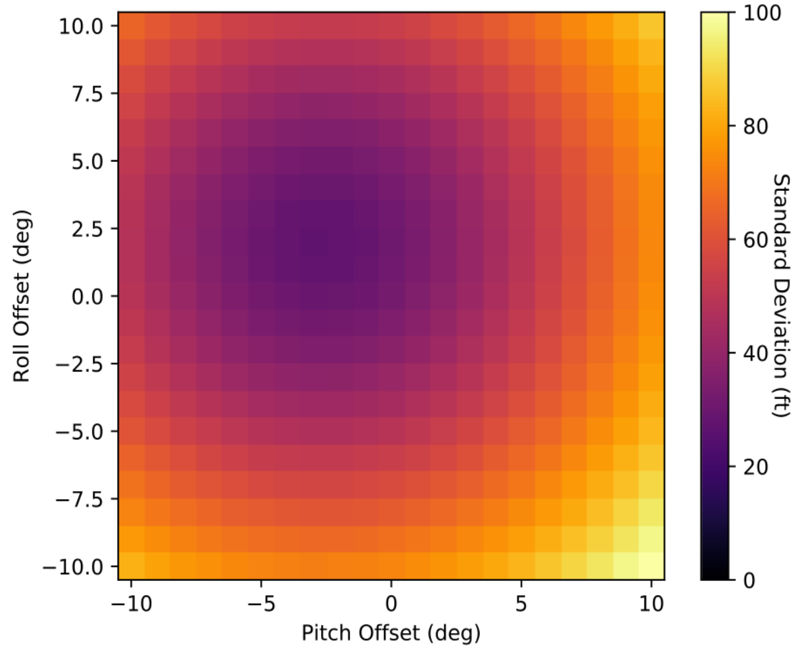

Fig. 5 Heatmap Visualization of Brute-Force Solver

Table 1 Accuracy Improvement Due To Camera Angle Calibration

Aircraft	Uncalibrated Accuracy	Calibrated Accuracy	Improvement
Skyhunter #5	20.34	10.00	103%
Skyhunter #6	29.90	11.22	129%

VIII. Related Works

A series of papers [5] [6] [7] on geolocation via flocks of coordinated UAVs have been published by William Whitacre and Mark Campbell of Cornell University, and Matt Wheeler of the Insitu Group. Wheeler's work is focused on flight control for a fleet of aircraft so that a tracked target always remains in view of one or more aircraft. Whitacre's paper focuses on errors in AHRS estimation (a topic not addressed in this paper) as a source of uncertainty in geolocation.

In another paper [8], Whitacre covers critical concepts to accurate geolocation, and their effects on geolocation precision. This is directly comparable to section IV, though Whitacre's study included moving targets, and high-rate image acquisition with an active target-tracking mechanism.

IX. Future Work

Following the calibrations detailed in this paper, the residual error present in the localization of ground objects is primarily parallel to the direction of travel of each aircraft. This error would likely stem from error in vehicle pitch, or proper synchronization of images with telemetry.The error still present is not consistently detected in either direction of the plane as would occur with a static pitch error, and static pitch error is already compensated for. Therefore the likely culprit for the residual error is jitter in the timing of telemetry data to the capture of images. In order to tightly synchronize the telemetry with the image capture, the Flash control on the camera could be used to generate an interrupt on the flight controller, triggering it to produce a unique telemetry packet containing the position at the instant the image is taken.

Additional parameters could be added to the set of parameters to be automatically discovered. "Camera trigger delay" or "altitude offset" might be good candidates, but any value that can be calibrated would be a candidate. This would come at the expense of requiring larger data sets with a wide distribution of flight conditions for accurate determination of all parameters.

Table 2 Camera Lens Calibration Std. Deviation Improvements

Aircraft	Uncalibrated StdDev	Calibrated StdDev	Improvement
Skyhunter #4	18.005	18.10886	-0.575%

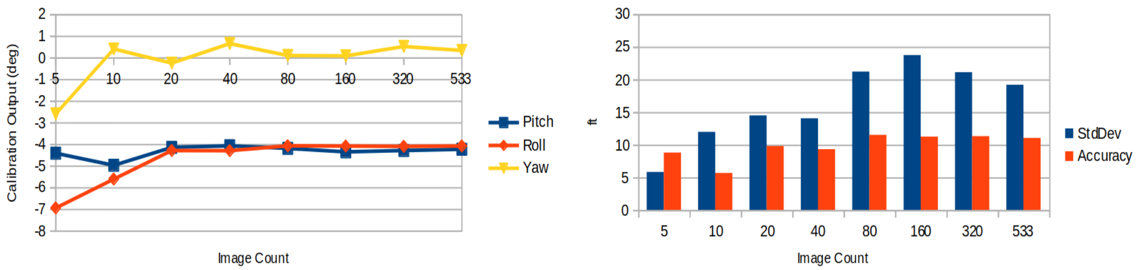

Fig. 6 Effects of Image Count on Calibration Output and Mean Error - Airplane #5

This paper has not focused on the necessary step of identifying targets in images and determining their pixel coordinates, but this is worthy of discussion. For maximum accuracy, the target locator must have sub-pixel accuracy, since a 1-pixel resolution limit is increasingly problematic as altitude and distance from the camera to the target increase. With large targets and sharp edges, edge- and corner-refinement technologies should be used to generate sub-pixel-accurate pixel coordinates. Further work could be done to determine the limits of geolocation accuracy with respect to pixel sizes and image processing strategies. Along similar lines, image intake steps that affect individual pixels should be examined. Compression, either in transmission of the image from the camera to the computer, or when the computer stores images to disk, will degrade image quality. De-warping of the image, as required by the lens distortion correction operation, also constitutes a re-rendering of the image in such a way that some pixel information is lost. It may be preferable to de-warp the output of the image detector, rather than the image itself.

Finally, a distinction could be made between parameters expected to have long-lasting calibrations (ex. pitch/roll/yaw offset), and those where calibrations are only accurate in the immediate temporal vicinity. Altitude offset, for example, may be driven by the immediate atmospheric conditions affecting the aircraft's altimeter. Such an effect could be calibrated for within a single set of images, but the calibration would be irrelevant to future flights.

X. Conclusion

Ultimately, flight testing demonstrated the ability to consistently recognize targets with an accuracy of several feet, given a flight altitude in the hundreds of feet. The methods described in this paper resulted in >100% improvements in geolocation accuracy without requiring any additional hardware or physical alterations to the drones. Room for improvement remains in the areas of single-detection accuracy and optimal flight patterns for target detection and camera calibration. More work could be put into identifying additional configuration values via gradient descent - for example possible altitude offsets or camera timing errors.

Flight testing is essential for obtaining good geolocation performance. Note that all of the methods described in this paper are solutions to problems that shouldn't exist in a perfect world. Given a perfect camera with perfect timing and perfect metadata from the aircraft, the geolocation problem reduces to simply shooting a ray through the image and seeing where it strikes the ground. What this paper supports is the ability to utilize inexpensive real-world systems, with a variety of sources of error, in the best way possible.

Acknowledgments

The authors would like to thank the Richmond Area RC club for the generous use of their flight field. Financial support for this research was provided by the Virginia Commonwealth University Department of Electrical and Computer Engineering.

References

[1] Alvarez, L., Gómez, L., and Sendra, J. R., "An Algebraic Approach to Lens Distortion by Line Rectification," *Journal of Mathematical Imaging and Vision*, Vol. 35, No. 1, 2009, pp. 36–50. doi:10.1007/s10851-009-0153-2, URL `https://doi.org/10.1007/s10851-009-0153-2`.

[2] Team, O. D., "Camera Calibration and 3D Reconstruction," , ???? URL `https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`.

[3] ReadyMadeRC, L., "Skyhunter FPV Plane," , 2019. URL `https://www.readymaderc.com/products/details/rmrc-skyhunter-kit`.

[4] Elmore, J., "Design of an All-In-One Embedded Flight Control System," *Theses and Dissertations*, 2015. URL `https://scholarscompass.vcu.edu/etd/3981`.

[5] Wheeler, M., Schrick, B., Whitacre, W., Campbell, M., Rysdyk, R., and Wise, R., "Cooperative Tracking of Moving Targets by a Team of Autonomous UAVs," *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, 2006, pp. 1–9. doi:10.1109/DASC.2006.313769.

[6] Whitacre, W., and Campbell, M., "Cooperative Geolocation with UAVs Under Communication Constraints," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2007. doi:10.2514/6.2007-6759, URL `https://arc.aiaa.org/doi/10.2514/6.2007-6759`.

[7] Campbell, M., and Wheeler, M., "A Vision Based Geolocation Tracking System for UAV's," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2006. doi:10.2514/6.2006-6246, URL `https://arc.aiaa.org/doi/10.2514/6.2006-6246`.

[8] Whitacre, W., Campbell, M., Wheeler, M., and Stevenson, D., "Flight Results from Tracking Ground Targets Using SeaScan UAVs with Gimballing Cameras," *2007 American Control Conference*, 2007, pp. 377–383. doi:10.1109/ACC.2007.4282696.