2017 IEEE International Symposium on
Safety, Security and Rescue Robotics (SSRR)
Shanghai, China, October 11-13, 2017

Visual Pose Stabilization of Tethered Small Unmanned Aerial System to Assist Drowning Victim Recovery

Jan Dufek[1], Xuesu Xiao[1], and Robin Murphy[1]

Abstract— **This paper proposes a method for visual pose stabilization of Fotokite, a tethered small unmanned aerial system, using a forward facing monocular camera. Conventionally, Fotokite stabilizes itself only relative to its tether and not relative to the global frame. It is, therefore, susceptible to environmental disturbances (especially wind) or motion of its ground station. Related work proposed visual stabilization for unmanned aerial systems using a downward facing camera and homography estimation. The major disadvantage of this approach is that all the features used in the homography estimation must be in the same plane. The method proposed in this paper works for features in different planes and can be used with a forward-facing camera. This paper is the part of a bigger project on saving drowning victims using lifesaving unmanned surface vehicle visually servoed by Fotokite to reach the victims. Some of the used algorithms are motion sensitive and, therefore, it is desirable for Fotokite to keep its pose relative to the world. The method presented in this paper will enable to prevent gradual drifting of Fotokite in windy conditions typical for coastal areas or when the ground station is on a boat. The quality of pose stabilization was quantitatively analyzed in 9 trials by measuring metric displacement from the initial pose. The achieved mean metric displacement was 34 cm. The results were also compared to 3 trials with no stabilization.**

I. Introduction

This research is the part of a long-term project aimed at helping first responders saving drowning victims in mass marine casualty events using a team of an unmanned surface vehicle (USV) and a small unmanned aerial system (sUAS) [1], [2].

The motivation for this research is the European refugee crisis. Many refugees drown because the main migratory route involves a journey across the Mediterranean Sea.

The first responders face two major problems. First, there are some areas which are inaccessible by rescue boats. Second, the number of responders immediately available at the particular location at the moment of the event is too low to immediately help everybody.

A partial solution currently tested by the Hellenic Coast Guard is to use a USV to operate in otherwise unreachable areas. Emergency Integrated Lifesaving Lanyard (EMILY), a USV by Hydronalix, was tested in teleoperation mode by Murphy et al. at Lesbos, Greece in 2016. EMILY is a 1.2 m long robotic flotation device that can reach speeds up to $35\,\mathrm{km\,h^{-1}}$ and can keep up to 6 people afloat.

Teleoperating EMILY addresses the first problem by enabling responders to reach otherwise unreachable areas. However, it makes the second problem even worse. Teleoperation reduces the number of responders that can assist in person since somebody has to teleoperate the USV. Therefore, the idea of this research project is to make EMILY autonomous and send it to less urgent cases while responders can help refugees who could be facing more urgent issues in person.

To make EMILY autonomous, it was combined with Fotokite Pro, a tethered sUAS from Perspective Robotics AG. sUASs are widely used in disaster response [3]. Having a tethered sUAS has two advantages for this project. First, the tether provides power from the ground station enabling unlimited flight time. Second, the tether serves as a leash so Fotokite cannot fly away and does not require constant supervision.

Current work is enabling the operator to select victims in the live video feed provided by Fotokite. EMILY then autonomously approaches the victims being visually servoed by the Fotokite.

Unfortunately, Fotokite cannot hold its position in the global frame of reference because it does not have any global positioning system. Fotokite position stabilization is done only relative to the tether by keeping it under constant length, elevation angle, and azimuth[1]. This leads to two problems. First, Fotokite follows any motion of the ground station, which is a problem if the ground station is on a moving platform such as a boat. Second, Fotokite behaves similar to a regular kite in the wind, even though it is a quad rotor. It gradually drifts with the wind until the tether is stretched in the direction of the wind.

Holding the position relative to the world frame is crucial for this project because the video feed from Fotokite is used in motion-sensitive visual servoing algorithms to control EMILY. Based on the stable visual feedback, a variety of robotic platforms can be controlled, such as wheeled robots or even snakes [4]. This paper presents a method for visual pose stabilization of Fotokite relative to the environment using an onboard forward facing monocular camera. The method is based on feature tracking, essential matrix estimation, pose reconstruction, and coordinate conversions to compute control signals canceling any unwanted rotation and translation.

From the viewpoint of technology, the project integrates various matured technologies to help first responders. From

[1] Jan Dufek, Xuesu Xiao, and Robin Murphy are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA dufek@tamu.edu, xiaoxuesu@tamu.edu, murphy@cse.tamu.edu

[1] Despite the fact that Fotokite cannot hold position relative to the global frame, it can still hold orientation relative to the global frame using inertial measurement unit. It is used to stabilize Fotokite's yaw and camera gimbal tilt and roll relative to the global frame.

978-1-5386-3923-8/17/$31.00 ©2017 IEEE

the hardware perspective, a robotic flotation device is teamed with a tethered sUAS. From the software perspective, the technologies used popularly in monocular SLAM are applied to stabilize the tethered sUAS.

The paper is organized as follows. First, related work will be discussed. Second, pose recovery of Fotokite will be presented. Third, the control of Fotokite will be explained. Fourth, experimental design and results will be presented, analyzed and discussed.

II. Related Work

The visual pose stabilization of a sUAS using an onboard monocular camera was explored in the past. The majority of the studies used a downward facing camera to detect features on a ground plane [5], [6], [7]. A homography was computed from feature correspondences from multiple views. This approach has two disadvantages. First, it assumes the camera is facing down. Second, it assumes that all the features are in the same plane. This is usually the case only for high altitude sUAS.

These two assumptions are not satisfied in this project. First, the camera of Fotokite is facing the rescue scenario at an oblique angle to provide the view of EMILY and the victims. Therefore, the features will be in different planes. Second, in the case of a tethered sUAS, the altitude is limited by the tether length. For example, Fotokite's maximum tether length is 20 m. From altitude this low, it is not safe to assume all the features will be in the same plane even if the camera was facing down.

A Jacobian-based technique for visual stabilization was proposed in [8]. While using a different technique, this method suffers from the same problems as the methods above, particularly it assumes the features will be in the same plane.

Some studies used even simpler approach measuring pixel displacement of features across multiple views [9]. This technique was also used in commercial applications [10]. This method does not enable full pose recovery but can be used to detect motion. The advantage is low computational cost. The major problem is that the method does not recognize if the drifting is caused by translation or rotation.

To the best of authors' knowledge, this paper appears to be the first one to use epipolar geometry applied to a monocular camera to recover pose from the essential matrix to stabilize the pose of a sUAS. The advantage is that the features can be in different planes, which is useful for a forward facing camera. The disadvantage is that if the features end up being coplanar, the system gets into a degenerate case.

III. Pose Recovery

The first step of the project was to find the rotation and the translation of the Fotokite relative to the initial pose selected for stabilization. The main idea was to remember significant features in the view from the desired pose and then compare each new incoming camera frame to see if the features moved. If yes, the essential matrix was computed using the corresponding features between the two views. From the essential matrix, a projection matrix was recovered. Projection matrix was decomposed to get the rotation and the translation matrices. This section will describe the method in more detail.

Fotokite has an onboard GoPro HERO4 camera that streams low-latency video to the Fotokite's ground control station. The GoPro enables to record 4K video, but for this project it was set to wide-angle WVGA mode with the resolution of 800 × 480 px to reduce computational load. The ground station provides HDMI output with low-latency high-definition (HD) video feed. Fotokite's tether has the maximum length of 20 m, so Fotokite can afford to wirelessly stream HD video. INOGENI 4K HDMI to USB 3.0 Converter was used to input the video feed from the Fotokite ground station's HDMI output to USB 3.0 port of a laptop computer. The camera intrinsic matrix K for the video feed was computed to calibrate the camera. The intrinsic matrix is sensitive to resolution and field of view changes, which is the reason why it was not computed from the camera directly, but rather from the actual video feed after all the conversions made during transmission.

The first incoming video frame from Fotokite was used as the initial frame to set the desired pose with respect to which Fotokite should stabilize. Each incoming frame was converted to grayscale because grayscale images resulted in the higher accuracy of feature tracking.

First, significant features were detected in this initial frame. Three methods for feature detection were tested. The first was SURF algorithm [11], which is the speeded-up version of SIFT [12]. The second was FAST [13], which, unlike SURF, detects only corners. The third was Shi-Tomasi [14], which is also a corner detector. It was shown that it gives better results than Harris Corner Detector [15].

The Shi-Tomasi algorithm was selected because it had superior performance for this particular application. The reason is that in this project, it was necessary to track features across frames and Shi-Tomasi is designed to find features suitable for tracking. The Shi-Tomasi algorithm was parametrized with the maximum number of corners set to 3000, quality level set to 0.01, minimum distance set to 2, and block size set to 3. The initial frame and the corresponding features found by Shi-Tomasi were saved as desired initial frame and desired features position.

After the initialization was done, the actual pose estimation relative to the initial frame was initiated. Each incoming frame was loaded and converted to grayscale. In this new incoming frame, features corresponding to the features in the initial frame were found. Two methods were considered for finding the corresponding features.

The first method was to find features in the current frame independently from the initial frame and then try to match them to the features in the initial frame using FLANN [16] based feature matcher. Most of the time, different features were detected in each frame and the matching was done without any constraints on the distance of the matched features leading to poor performance.

The second method was based on optic flow. Initial features were tracked to the new incoming frame. This was done by taking the position of each feature in the initial frame and trying to find the corresponding feature in the new incoming frame in the imaginary window of fixed small size around the same position. Lukas-Kanade method with pyramids [17] was used for this purpose. The method was parametrized with square tracking window of size 35 px and the maximum number of levels equal to 3. This method explicitly looks for each corresponding feature around the position where the feature originally was in the initial frame leading to superior performance.

The features for which the tracking failed were considered lost and were deleted. The features that went out of the frame were deleted as well. If the number of features dropped bellow certain threshold, the entire stabilization was reset. This caused the Fotokite to set the initial frame to the current frame. This made the system more fault tolerant since if Fotokite got lost and could not stabilize relative to the initial frame, it at least stabilized on the current frame to prevent further drifting.

The next step was to compute Fotokite's current pose relative to the initial frame from the corresponding features between the initial frame and the current frame. Let x_i be features in the initial frame (first view) and x'_i corresponding features in the current frame (second view), both in homogeneous coordinates. There were up to 3000 of those correspondences. The points were in the image coordinates and therefore were subject to camera distortion. Therefore, the points were undistorted using camera distortion coefficients. Then the reverse perspective transformation was applied using the camera intrinsic matrix K. The new intrinsic matrix for the new points was computed.

Let the projection matrix for initial frame (first view) be:

$$P = [I|0] \tag{1}$$

The goal was to find the projection matrix for the current frame (second view) relative to the initial frame (first view). The epipolar geometry of the two views is given by the following equation:

$$x_i^T K^T E K x_i'^T = 0 \tag{2}$$

Where K is the Fotokite's camera intrinsic matrix and E is essential matrix. Using the undistorted point correspondences x_i and x'_i, essential matrix E was computed from Equation 2 using five-point algorithm [18] and RANSAC [19].

The optimal triangulation method [20] was used to refine the point correspondences x_i and x'_i using the found essential matrix E. The corrected points were selected to minimize the geometric error:

$$d(x_i,\widetilde{x}_i)^2 + d(x'_i,\widetilde{x}'_i)^2 \tag{3}$$

Where $\widetilde{x}_i$ and $\widetilde{x}'_i$ are the new corrected points and d function is a geometric distance subject to epipolar constraint:

$$\widetilde{x}_i'^T F \widetilde{x}_i^T = 0 \tag{4}$$

Where F is the fundamental matrix. The essential matrix E and the corrected points $\widetilde{x}_i$ and $\widetilde{x}'_i$ were used to recover the pose. The essential matrix was decomposed using singular value decomposition to rotation matrix R and translation vector t. Generally, up to four solutions exist. Therefore, the cheirality check [18] was used to select the correct solution. The cheirality check is based on the idea, that triangulated 3D points should be in positive depth. This gave a single solution to R and t.

The last step was to convert R to Euler angles θ.

Let:

$$R = (r_{ij}) \tag{5}$$

Then:

$$\theta_x = \arctan(r_{32}, r_{33}) \tag{6}$$

$$\theta_y = \arctan\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \tag{7}$$

$$\theta_z = \arctan(r_{21}, r_{11}) \tag{8}$$

Where θ_x is rotation around x-axis, θ_y is rotation around y-axis, and θ_z is rotation around z-axis. Before passing θ and t to controls, the values were preprocessed in two steps.

The first step was an outlier rejector. If any component of the Euler angles θ in two consecutive frames changed by more than 5°, the measurement was rejected as an outlier. Only after three consecutive outliers were rejected, the measurement was accepted as the new value. This outlier rejector was based on a heuristic, that it is unlikely that θ would change by more than 5° in two consecutive frames.

The second step was setting thresholds on the estimated values of t and θ. This is especially important for t. The five-point algorithm only recovers t as a unit vector without scale (unless there is an object of known size in the field of view, the scale of t cannot be recovered from feature correspondences). Therefore, t is a unit vector showing only relative translation in the 3-axes. Because t is a unit vector, it is never zero, not even when only very small movement occurred.

To solve this issue, the Euler angles θ were checked first and if the absolute value was more than 5°, then the controller was forced to correct the angular difference close to 0° first. Then, the pixel displacement of features was computed as the mean distance between corresponding features in the current frame and the initial frame in image coordinates. If the features were displaced by more than the certain threshold while θ was close to 0, the displacement of features in the image frame was due to translation. In this case, the values of t were passed to the controller to correct the translation. If the displacement of features in the image frame was bellow certain threshold, no action was required by the controller.

IV. Control

This section is divided into two subsections. The first subsection details orientation control. The second subsection details position control.

A. Orientation Control

The only orientation degree of freedom that was controlled was yaw (θ_y). The desired value of θ_y was 0, which represent 0° difference from the initial position. Yaw, unlike xyz-translation, can be controlled directly. The current θ_y estimated by the vision system was used in a PID controller with proportional gain to control Fotokite's yaw.

Neither tilt (θ_x) nor roll (θ_z) were controlled. Fotokite has gimbal electrically stabilized relative to gravity in tilt and roll axes, so further stabilization was not necessary in these axes.

B. Position Control

Fotokite has unconventional position kinematic model. The movement along xyz-axes cannot be controlled nor measured directly. The only position parameters that can be controlled and measured directly are:

1) Tether length l
2) Tether elevation angle e
3) Tether azimuth a

Fotokite stabilizes its position by keeping those values constant. Therefore, if no control signals are sent, l, e, and a remain the same. The important fact is that those values represent the relative position of Fotokite's body to the tether and not to the world frame. Therefore, if the ground station or the tether is moved, Fotokite moves as well. Let l_m be the measured tether length, e_m the measured elevation angle, and a_m the measured azimuth angle.

Then measured x_m, y_m, and z_m position coordinates of Fotokite can be computed:

$$x_m = l_m \cos e_m \cos a_m \tag{9}$$

$$y_m = l_m \cos e_m \sin a_m \tag{10}$$

$$z_m = l_m \sin e_m \tag{11}$$

The measured values for current Fotokite's orientation were obtained as quaternion from onboard IMU and converted to a rotation matrix.

All the measured values for current position and orientation were composed into a homogeneous transformation matrix. This matrix gave the current pose of Fotokite relative to the ground station. Another homogeneous transformation matrix was composed from t. This matrix gave the desired position relative to Fotokite's body. Those two homogeneous matrices were multiplied to get homogeneous transformation matrix giving the desired position relative to the ground station. This matrix was converted to the desired values of tether length, azimuth, and elevation angle. A PID controller with a proportional gain was used to control the tether length, azimuth, and elevation angle towards the desired values.

The entire process from the pose recovery to control is illustrated in Figure 1. The overall computational complexity is limited by the time complexity of the used methods cited in Section III.

V. Experiments

This section presents the experiments performed to show the difference between normal and visually stabilized flight and to quantitatively evaluate the quality of the proposed visual stabilization.

In order to test the system, the fact that Fotokite normally keeps constant pose relative to the ground station was used. If no visual stabilization is used and the ground station is moved, the Fotokite shadows the motion of the ground station and, therefore, moves relative to the world. With the visual pose stabilization on, however, Fotokite should hold its pose relative to the world, even if the ground station is moved.

The experiments were performed in an indoor laboratory. Fotokite was always started at the same pose relative to the world for each trial set. The coordinate system of the world and the experimental setup are illustrated in Figure 2. Since a motion capture system was not available to get the ground truth, the floor was marked in x and z axes with markers placed at every 0.5 m so that the ground truth position of Fotokite can be measured.

The ground station was placed on a movable cart so that it could be easily moved. There were 3 sets of trials with 4 trials in each set. In the first set of trials, the ground station started under Fotokite's position at the origin and was moved along x axis in one direction for 2 m, then back to the origin, then 2 m in the opposite direction and then back to the origin. This contained the total of two changes in direction of movement of the ground station and the total of 8 m traveled by the ground station. The movement along y and z axes was fixed. First 3 trials in the set were performed with visual stabilization on, and the last one with visual stabilization off. Exactly the same was done for z-axis, except that the movement of the ground station was along z-axis and x and y axes were fixed.

For y axis, the setting was slightly different. The ground station can never be above Fotokite because the tether would be cut by the propellers. For this reason, Fotokite was placed near the ceiling above the origin. The ground station started in the middle, went all the way up, all the way down and back to the center. The x and z axes were fixed. This was done three times with visual stabilization on and once with visual stabilization off.

This experimental design enabled to measure the quality of the visual stabilization along each of the axes separately.

Two cameras were used to record the experiments. One was placed parallel to xy-plane and one parallel to yz-plane. The video together with the markers on the ground enabled to approximately measure the ground truth x, y, and z coordinates of Fotokite relative to the world. The xy-plane camera was used to measure Fotokite's position in x and y coordinates. The yz-plane camera was used to measure Fotokite's position in z coordinates. Since Fotokite only moved in those planes, it was always in the same plane as the floor markers.

Fotokite was selected in each video frame during post-processing to get its ground truth pixel coordinates. The

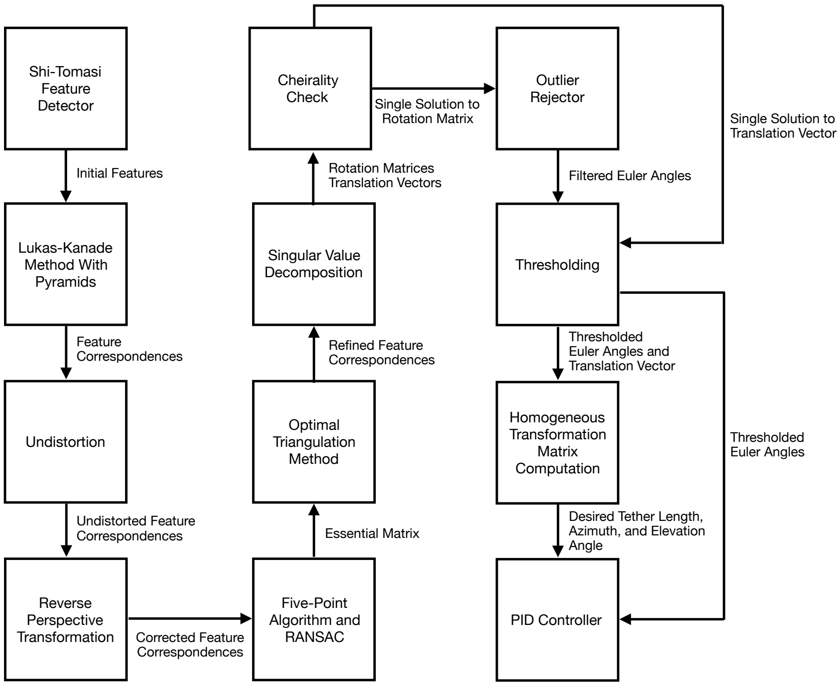
Shi-Tomasi Feature Detector
Initial Features
Lukas-Kanade Method With Pyramids
Feature Correspondences
Undistortion
Undistorted Feature Correspondences
Reverse Perspective Transformation
Corrected Feature Correspondences
Five-Point Algorithm and RANSAC
Essential Matrix
Optimal Triangulation Method
Refined Feature Correspondences
Singular Value Decomposition
Rotation Matrices Translation Vectors
Cheirality Check
Single Solution to Rotation Matrix
Outlier Rejector
Single Solution to Translation Vector
Filtered Euler Angles
Thresholding
Thresholded Euler Angles and Translation Vector
Thresholded Euler Angles
Homogeneous Transformation Matrix Computation
Desired Tether Length, Azimuth, and Elevation Angle
PID Controller

Fig. 1. Flowchart for the method.

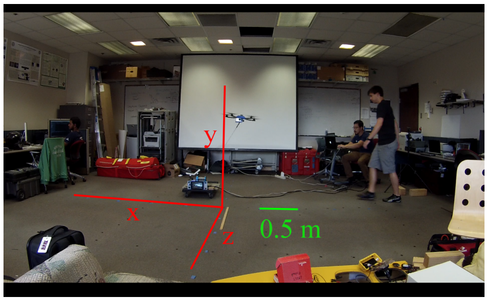

Fig. 2. The experimental setup and the coordinate system of the world for the purpose of the experiment.

Fotokite selection was done semi-automatically using MATLAB with Kanade-Lucas-Tomasi tracking algorithm. The floor markers of known position were used to convert the pixel coordinates to the ground truth metric coordinates by linear interpolation. The Fotokite's ground truth position in each frame was subtracted from the initial ground truth position to get the metric displacement in that frame. For example, the metric displacement of 0.5 m in x-axis means that Fotokite is 0.5 m away from its initial position (the position it was supposed to hold) in the direction of x-axis. The desired displacement is 0 in all the axes.

The results of the experiments are presented in three graphs. The first graph in Figure 3 shows the displacement heatmap. Higher heat (yellow color) indicates more time Fotokite spent in the position with that particular displacement value. Each row represents the trial set with ground station movement along x, y, or z axes respectively. In each row on the left are consolidated the three trials with visual stabilization on and on the right is the trial with visual stabilization off. As you can see, for all the axes the highest heat is around $[0,0]$ for visual stabilization on, meaning that most of the time the displacement was around 0 in all axes. On the other hand, with visual stabilization off, you can see that Fotokite moved freely with the ground station.

The second graph in Figure 4 shows the displacement histogram. The height of each bar represents the number of frames Fotokite had the displacement in the corresponding range given on the graph's horizontal axis. Each row represents the trial set with ground station movement along x, y, and z axes respectively. In each row, histograms for visual stabilization on and visual stabilization off are overlaid. As you can see, for the visual navigation on, the displacement is distributed around 0. On the other hand, for visual stabilization off, the displacement is evenly distributed across the range, which is consistent with the fact that Fotokite moved freely with the ground station. You can note that the displacement for y is shifted approximately 20 cm. This is probably because the initial position with 0 displacement was near the ceiling negatively influencing aerodynamics.

The third graph in Figure 5 shows the displacement in time for each individual trial. Each row represents the trial set with ground station movement along x, y, and z axes respectively. All the trials with visual stabilization on are in dotted blue line and with visual stabilization off are in solid red line. Each subgraph contains three trials with visual stabilization on and one with visual stabilization off. You can see that for visual stabilization on, the displacement is always

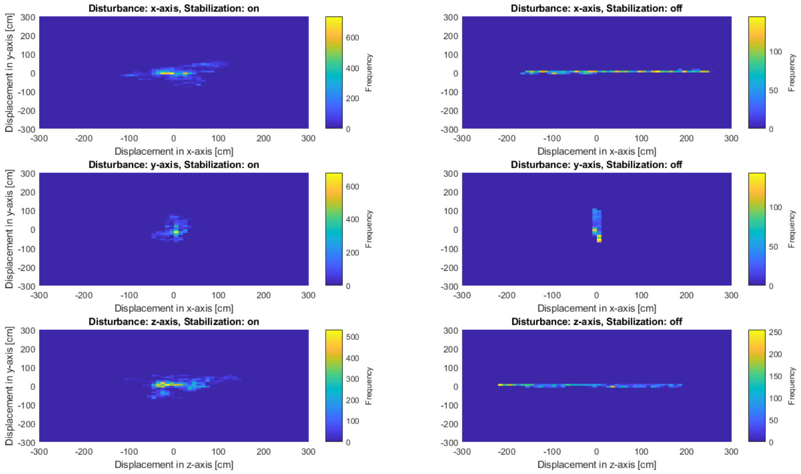

Fig. 3. Displacement heatmap.

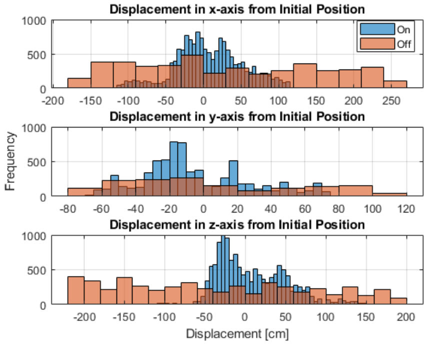

Fig. 4. Displacement histogram.

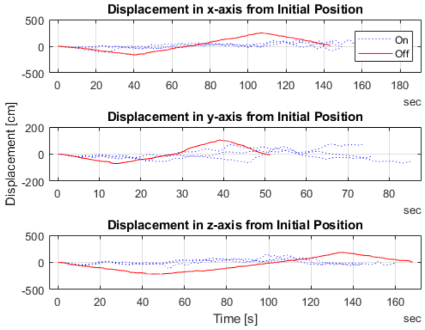

Fig. 5. Displacement in time.

within some range of 0. However, for visual stabilization off, the displacement reflects the fact that Fotokite followed the ground station. This is consistent with the experiment since the ground station went in one direction (2 m for x and z), then back, then the other direction (2 m for x and z) and then back.

Table I lists mean and standard deviation of displacement for each axes for visual stabilization on and visual stabilization off. You can see that visual stabilization on has the lower mean and standard deviation displacement overall and also for each individual axis.

TABLE I
MEAN AND STANDARD DEVIATION OF DISPLACEMENT

Stabilization		***x*-axis**	***y*-axis**	***z*-axis**	**all axes**
On	**Mean**	36 cm	26 cm	35 cm	34 cm
	Standard Deviation	45 cm	31 cm	43 cm	42 cm
Off	**Mean**	106 cm	43 cm	108 cm	98 cm
	Standard Deviation	122 cm	52 cm	119 cm	118 cm

It should be noted, that our experimental space was limited so the maximum possible displacement was 2 m. Since Fotokite followed ground station when the visual stabilization was off, the displacement for visual stabilization off can be made arbitrarily high if enough space is available to move the ground station. In that case, the difference between visual stabilization on and visual stabilization off would be even more significant.

On a 2.5 GHz Intel Core i7 laptop computer with algorithms parameters set to the values mentioned in this paper, the average time for one cycle (from loading a video frame to issuing control commands) was 206 ms corresponding to the frequency of 4.85 Hz. The experiments showed that the algorithm is able to control Fotokite in real-time.

VI. Conclusion

This paper proposed, implemented, and tested a visual pose stabilization technique for Fotokite, a tethered sUAS. The method used epipolar geometry to recover pose information from the essential matrix using multiple views from a single forward-facing monocular camera. This method has not been used before to stabilize a sUAS. Unlike related work, this technique works even if the detected features are not in the same plane, which is the case for forward-facing cameras.

According to the experimental results, the system provides a feasible solution for Fotokite pose stabilization relative to the global frame of reference. This will enable Fotokite to stabilize relative to the world in windy conditions or if the ground station is moving.

This project is part of a bigger project in which Fotokite visually servoes a USV-based flotation device to reach drowning victims only based on image-frame visual feedback. Some of the algorithms used are motion sensitive, so it is crucial for Fotokite to remain stable relative to the world frame. The visual pose stabilization presented in this paper will enable the system to be robust in windy conditions typical for coastal areas and to enable Fotokite being stable even if the ground station is on a moving platform, such as a boat.

In terms of future work, more testing in outdoor conditions will be performed. Especially important will be to test the system under varying wind conditions. In the experiments in this paper, only position stabilization was explicitly analyzed. Orientation stabilization was active, but not analyzed in this paper due to the difficulty to artificially disturb Fotokite's orientation in indoor conditions. Testing outside will enable the wind to disturb the orientation and will make it easier to analyze the performance. Lastly, integration testing will be performed together with USV visual servoing to verify that the implemented stabilization system makes the visual servoing perform better.

References

[1] J. Dufek and R. Murphy, "Visual pose estimation of USV from UAV to assist drowning victims recovery," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2016, pp. 147–153.

[2] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "UAV assisted USV visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017.

[3] R. Murphy, J. Dufek, T. Sarmiento, G. Wilde, X. Xiao, J. Braun, L. Mullen, R. Smith, S. Allred, J. Adams, A. Wright, and J. Gingrich, "Two case studies and gaps analysis of flood assessment for emergency management with small unmanned aerial systems," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2016, pp. 54–61.

[4] X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M. J. Travers, and H. Choset, "Locomotive reduction for snake robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3735–3740.

[5] N. Metni and T. Hamel, "A uav for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3 – 10, 2007.

[6] A. Ollero, J. Ferruz, F. Caballero, S. Hurtado, and L. Merino, "Motion compensation and object detection for autonomous helicopter visual navigation in the comets system," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 1, April 2004, pp. 19–24 Vol.1.

[7] L. Merino, F. Caballero, J. Martnez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 165–184, 2006.

[8] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, April 2008.

[9] S. G. Fowers, D. J. Lee, B. J. Tippetts, K. D. Lillywhite, A. W. Dennis, and J. K. Archibald, "Vision aided stabilization and the development of a quad-rotor micro uav," in *2007 International Symposium on Computational Intelligence in Robotics and Automation*, June 2007, pp. 143–148.

[10] P.-J. Bristeau, F. Callou, D. Vissire, and N. Petit, "The navigation and control technology inside the ar.drone micro uav," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1477 – 1484, 2011, 18th IFAC World Congress.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded Up Robust Features*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

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

[13] E. Rosten and T. Drummond, *Machine Learning for High-Speed Corner Detection*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[14] J. Shi and C. Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994, pp. 593–600.

[15] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Manchester, UK, 1988, pp. 10–5244.

[16] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," vol. 1, Setubal, Portugal, 2009, pp. 331–40.

[17] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.

[18] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[20] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146 – 157, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314297905476