Vision based Collaborative Localization for Swarms of Aerial Vehicles

Sai Vemprala
PhD Candidate
Texas A&M University
College Station, TX, USA

Srikanth Saripalli
Associate Professor
Texas A&M University
College Station, TX, USA

ABSTRACT

We present a framework for localizing a swarm of multirotor micro aerial vehicles (MAV) through collaboration using vision based sensing. For MAVs equipped with monocular cameras, this technique, built upon a relative pose estimation strategy between two or more cameras, enables the MAVs to share information of a common map and thus estimate accurate metric poses between each other even through fast motion and changing environments. Synchronized feature detection, matching and robust tracking enable the use of multiple view geometry concepts for performing the estimation. Furthermore, we present the implementation details of this technique followed by a set of results which involves evaluation of the accuracy of the pose estimates through test cases in both simulated and real experiments. Our test cases involve a group of quadrotors in simulation, as well as real world flight tests with two MAVs.

INTRODUCTION

Micro aerial vehicles (MAV) are currently popular choices for many robotic applications. Their multirotor configuration paired with small size, light weight, relative inexpensiveness and prototyping ease make them capable of navigating in challenging environments, thus being applicable for disaster management, reconnaissance, aerial imaging etc. Particularly in recent years, there is growing interest in the idea of swarms of MAVs which behave as centralized or decentralized groups, which can improve factors such as mapping coverage etc., thereby enhancing the efficiency in terms of performing the task at hand. As the areas of application for MAVs and the environments they need to traverse grow more challenging, precise localization becomes essential.

One of the most widely used technologies to determine the position and orientation (pose) of an MAV is through the combination of a GPS receiver and an inertial measurement unit (IMU). Fusing these sensors results in the six degree-of-freedom pose of the MAV and provides a simple and straightforward way of localization. Yet, the GPS/IMU technology has its shortcomings: GPS reception cannot always be guaranteed, it can suffer from problems with obstructions, multi-path (while in cluttered environments and/or low altitude flights), and is inapplicable for indoor flight. Given these problems, research has tried to identify alternative sensing mechanisms, by utilizing other sensors such as sonars, radio beacons, cameras and laser scanners. Out of these, monocular cameras particularly demonstrate high potential as onboard sensors for MAVs as they are small in size and almost ubiquitous in present-day MAV platforms. Using the images from monocular cameras, computer vision algorithms allow for simultaneously localizing vehicles as well as mapping the environment to allow for navigation. Computer vision based approaches are also effective for multiple vehicles: under the constraint that there's at least a minimum amount of overlap between what different cameras are observing, multiple view geometry concepts can be used for pose estimation. Hence, a system with two MAVs equipped with cameras can be considered as a decoupled stereo system. Especially considering the fact that a single camera pose estimation suffers from the issue of scale ambiguity, it is advantageous to use the information from neighboring MAVs to resolve the scale of the environment.

Vision based localization has been studied extensively in the literature. Pose estimation for aerial vehicles using optical flow is performed onboard the commercially available PX4Flow camera (Ref. 1). More advanced estimation algorithm such as monocular SLAM have been investigated onboard multirotor vehicles, which try to remove scale ambiguity either by fusing vision data with an IMU (Ref. 2) or using multiple initial views (Ref. 3) to obtain metric scale information. Similar SLAM strategies have been studied using stereo cameras (Ref. 4) and RGBD sensors such as the Microsoft Kinect (Ref. 5) with intended applications onboard robotic vehicles. (Ref. 6) presents a relative visual odometry technique for a single ground vehicle equipped with a monocular camera which enhances accuracy by using the relative rotation angle from a different sensor such as an IMU as an estimate.

Collaborative/relative localization has been studied in the literature for multiple vehicles as well, although mostly involving ground robots. (Ref. 7) presents a technique for ground robots to perform cooperative localization using infrared LEDs. Similarly, fiducial markers are used for collaboration during localization in (Ref. 8). Another cooperative estimation framework, presented in (Ref. 9) uses RGBD sensors.

Presented at the AHS International 73rd Annual Forum & Technology Display, Fort Worth, Texas, USA, May 9–11, 2017.

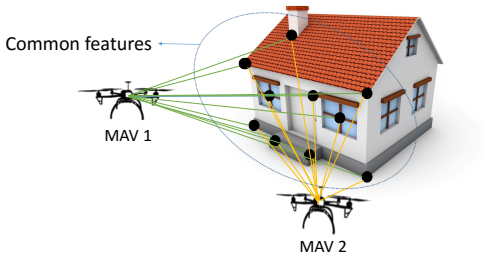

Fig. 1. Collaborative localization between two MAVs

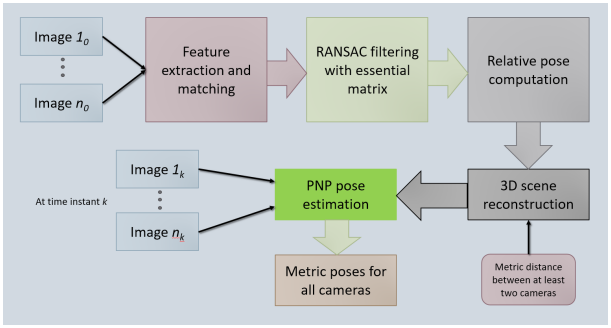

Fig. 2. CL algorithm outline

Relative localization using robot-to-robot distance measurements, which is then optimized using an iterative least squares method has been presented in (Ref. 10). In (Ref. 11), an approach is discussed for collaborative aerial vehicle-ground vehicle communication and localization. More recent work presents an idea of collaborative formation control of UAVs using vision sensors. (Ref. 12)

In this paper, we present the framework and implementation of a collaborative localization method using vision for a group of micro aerial vehicles, as an extension to our previous work presented in (Ref. 13). We equip each MAV with a monocular camera, which is used as the primary sensor. A feature-based method detects salient features in the environment from each vehicle (treated as the primary source of information), which are then used for relative localization between the MAVs through correspondence matching. This collaborative localization technique provides two advantages compared to other localization approaches. Visual odometry and SLAM techniques have been discussed extensively in the literature, but applying separate algorithms independently on all vehicles creates problems such as increasing the computational load on each one; as well as introducing multiple sources of error which do not correlate with each other. Hence, it is desired to perform localization as part of a group that is sharing common information, which ensures that all vehicles are in one frame of reference. Secondly, MAV swarms usually focus on small, lightweight vehicles: which constrains the sensing and computational ability of each vehicle. Hence, it's desirable to perform only low-level local sensing individually rather than performing complex sensor fusion on each platform (Ref. 14).

COLLABORATIVE LOCALIZATION

Figure 1 demonstrates the concept of this collaborative localization approach with two MAVs. In this concept, two MAVs are assumed to be equipped with monocular cameras observing a scene by capturing images. From these images, salient features are detected and common ones matched (encircled in the image). These matched features, coupled with previously known poses of the MAVs can be used to reconstruct the environment as a set of 3D points, which is then stored as a global map. The MAVs continue to take pictures of the scene, and by comparing each image and visible feature points in it with the global map, the new poses are computed for that particular vehicle. Throughout this approach, one MAV can be assumed to act as a reference (the origin) for the coordinate system, and the second vehicle (and possibly more) is localized with respect to the first. As the common features currently visible are tracked over images, once the number of features tracked falls under a certain minimum number, the algorithm clears the existing point cloud and the MAVs collaborate again with their feature matches to create a new one, thus ensuring that localization will not fail due to changes in the scene or fast motion of the MAVs.

To initialize our collaborative localization framework for a group of MAVs, we make two important assumptions.

1. All of the cameras are calibrated, and the camera intrinsic parameters such as focal length, distortion coefficients etc. are known.

2. The initial distance and heading difference between at least two members in the group are accurately known.

For convenience of explanation, we assume there are two MAVs while describing the algorithm.

Feature tracking and matching

To proceed with pose estimation for two MAVs, the first step is to identify important areas in the environment (features) being observed by both cameras simultaneously through a robust and error-free feature identification and matching framework. To identify salient features, we currently use the SIFT (Scale Invariant Feature Transform) technique (Ref. 15). SIFT can robustly identify distinguishable keypoints from an image while being invariant to scaling and orientation. Each keypoint has an associated descriptor, which stores pixel information of the area around the keypoint, which can subsequently be used for tracking or matching the feature.

The SIFT approach is used on both images coming from the MAV's cameras to identify keypoints. As a next step, the algorithm tries to match both sets of keypoints and evaluate the number of corresponding points, i.e., the features visible from both cameras. Given the keypoint location and descriptor data as mentioned above, we utilize approximate nearest neighbor matching to match features and find the common points.

Feature refinement and relative pose

Once the feature matches between both cameras are computed, it is then possible to estimate the pose of one camera with respect to the other, and reconstruct the observed environment using both views. Computing this transformation between views, although mathematically straightforward, is susceptible to inaccuracy arising from fast movement, rapidly changing illumination, rotations etc. Hence, the computed feature matches would need to be refined before being used for pose estimation.

A typical way of solving this problem is by using the random sample consensus method (RANSAC). RANSAC is an iterative method that seeks to find outliers from the provided set of data points, which in this case would be the feature matches. Given two images, the transformation between the two sources of the images (cameras) is encoded through the epipolar constraint. Hence, the feature matches obtained (as described in the previous section) can be considered in sets to see whether or not they adhere to this model, and the ones that do not are considered outliers and discarded.

RANSAC typically requires the choice of a parameter known as threshold (T), which determines the confidence. But as in our application, we would require the pose estimation to happen multiple times as the vehicles are in motion, the noise levels of the images/features would not be constant, and presetting the threshold parameter could result in degradation of performance over time. To avoid this problem, we create a more robust approach for relative pose estimation by utilizing a technique known as an a-contrario RANSAC (AC-RANSAC) method, proposed by Moulon et al (Ref. 16). We propagate the feature matches through this a-contrario approach, which adaptively chooses a choice of the parameter T according to the noise in the given data.

This adaptive RANSAC scheme tries to pick feature matches that minimize a distance error metric between the putative point correspondences, i.e., for a given point correspondence pair x_i and x_i', with $\hat{x}_i$ and $\hat{x}_i'$ defined with respect to a pair of matching epipolar lines, the RANSAC algorithm examines the number of correspondences that lie within a threshold of the distance error. The distance error metric e can be expressed as in equation 1. Minimizing e thus is equivalent to minimizing the sum of squared distances (SSD) of the matched points from the epipolar lines, which ensures that the feature matches adhere to the epipolar constraint as closely as possible.

$$e = d(x_i, \hat{x}_i)^2 + d(x_i', \hat{x}_i')^2 \tag{1}$$

Once we obtain a robust model of correspondences between two views, we then compute the relative rotation and translation between the views. In this algorithm, we achieve this step using the essential matrix E, that encodes the transformation between the multiple views. We use the five-point algorithm to compute the essential matrix (Ref. 17). Unlike the algorithms such as the 8-point algorithm that are used to compute the fundamental matrix, 5-point algorithm does not have a degeneracy case if there are coplanar points (Ref. 17), which can be a common occurrence in general scenes. Once the essential matrix is obtained, it can be decomposed to obtain the rotation matrix and the translation vector.

Triangulation and metric pose estimation

Through the process described above, the rotation and translation of all the other views relative to the 'origin' can be obtained. Using these multiple views, the next step is to reconstruct the scene through triangulation, which later acts as the common source of information for the MAVs to perform pose estimation. We use the Hartley-Sturm optimal triangulation procedure (Ref. 18), which uses the locations and orientations of both cameras computed as in the previous section, and the feature match data to create a point cloud of the triangulated features. When the algorithm is initialized for the first time, the metric scale between the cameras (vehicles) is known and provided (in the context of MAVs, assumption 2 can be satisfied by placing them in specific spots before the flights begin), which allows for the construction of an accurate point cloud. This point cloud is then utilized against the images that the MAVs capture, in order to estimate the poses of all cameras, and can be stored as a global map for easy access from all MAVs.

Once the MAVs start moving, they continue to capture images of the scene from their respective new locations. As before, salient features are detected in these new images, and a feature tracking check is performed to see whether any features that are part of the global map are still visible from the new images. As the 3D metric positions of all the mapped features are known, these correspondences between the 3D map and the 2D image features can be used to compute the new pose of the camera and thereby the MAV that has moved. Given these new images, and the existing map from the point cloud, it is possible to compute the metric poses at each location. This process of estimating the pose from 3D-2D correspondences is termed the perspective N point (PnP) method. We apply an iterative Levenberg Marquardt process to solve for the pose of the MAV through minimization of the reprojection error. While dealing with rapidly changing environments, a robust solution is required to solve this problem. Hence, we run the PNP algorithm coupled with the previously described a-contrario RANSAC procedure to ensure the same levels of noise rejection. In this way, the algorithm starts off with a matching and triangulation step, and for a certain amount of time, relies on the MAVs performing localization independently, assuming that they still have access to the global map. This ensures that the MAVs do not have to perform relative localization at every instant, and can depart to further distances from each other.

Until the end of a certain window in time (which is determined by whether enough features from the map are still visible), this global map is stored and used to estimate the metric poses using the 3D to 2D correspondences. Once the number of features that are both part of the point cloud and are still visible

from the cameras falls under a certain threshold, the MAVs collaborate to perform feature matching again, resulting in an updated point cloud, and thereby, global map. Thus, the algorithm starts with point cloud construction and switches to individual MAV localization, a process that's repeated at each window in time. By switching to 3D-2D correspondences and only communicating when a map update is needed, there is no requirement of continuous communication between the vehicles, and the only information that needs to be accessed during this period of time would be the set of points forming the global map.

IMPLEMENTATION AND RESULTS

We have implemented the collaborative localization (CL) algorithm described in the previous section on datasets from both simulation and real experiments. The simulation experiments were used for preliminary validation, and were done using the RotorS simulator in ROS/Gazebo. For the real experiments, we used a two-camera setup of Point Grey Chameleon3 USB3 cameras and two MAVs which were a DJI F450 quadrotor and a 3DR X8 platform. The images were captured at 15 Hz, and with an effective resolution of 1280x960 each. Both MAVs contained the 3DRobotics PIXHAWK acting as the main flight controller. They were also equipped with a UBlox GPS receiver, the data from which is fused with the internal IMU by the PIXHAWK and propagated through an extended Kalman filter (EKF) to provide pose estimates for comparison. Odroid U3 single board computers were installed on the MAVs for image logging. The pose estimation was performed offline, using datasets obtained from the flights. Open source implementations of algorithms such as the AC-RANSAC, PNP etc. were used from the packages OpenCV (Ref. 19) and OpenMVG (Ref. 20).

Simulation results

In figure 3, we show the result of the CL initialization procedure for a swarm of MAVs (9 in this instance). The two MAVs that have the longest baseline between them were used for feature matching and reconstruction, and the rest of the MAVs were localized with respect to this map. 3(a) shows a screenshot from the simulator Gazebo, where the MAVs are seen observing a common scene, which in this case is a house. The green points in figure 3(b) denote the MAV positions, whereas the white points are part of the point cloud (features from the house) that represents the global map.

Furthermore, figure 4 shows the result from three MAVs moving in square-shaped patterns in the simulator using the CL algorithm, side by side. The three paths of the MAVs are plotted in red, and the ground truth is plotted in blue. It can be seen that the CL algorithm's pose estimates track the ground truth closely, and we have observed RMS errors of 5-10 cm for the complete path.

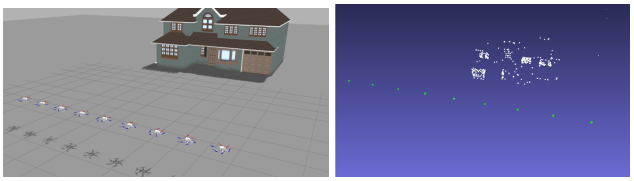

Fig. 3. Relative localization between nine vehicles sharing a common scene. Scene (left) is observed through their onboard cameras, and the feature matches between the vehicles are used to reconstruct the 3D representation (right), as well as the positions of each vehicle. Screenshot obtained from the Gazebo simulator.

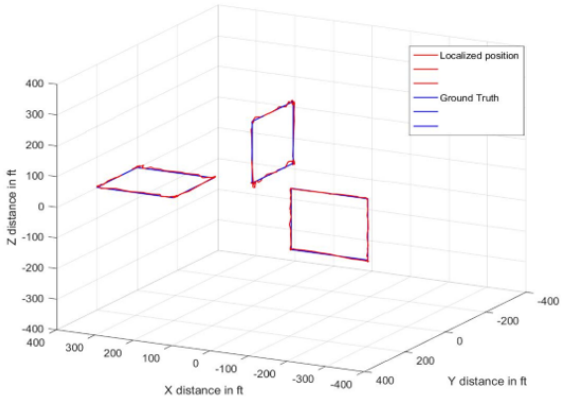

Fig. 4. Positions of three MAVs flying in square trajectories in the Gazebo simulator, as computed by the CL algorithm. While CL positions are plotted in red, the ground truth is plotted in blue, and the CL localization matches the ground truth closely.

Results from real experiments

For our real experiments, we used images captured from the X8 and F450 platforms, which were processed offboard for obtaining the pose estimates. Figure 5 shows the working of the AC-RANSAC for a pair of sample images, demonstrating how it's useful for removing outliers in our application. Erroneous feature matches from figure 5(a), which don't adhere to the epipolar constraint are removed by AC-RANSAC, thus resulting in accurate matches that can be described by a single transformation, which directly translates to a pose estimate with low uncertainty.

Figure 6 shows the trajectory of the X8 as it was commanded

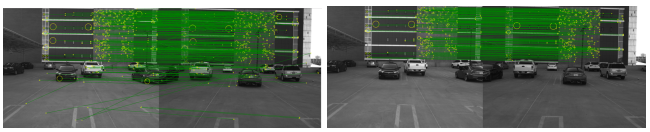

Fig. 5. AC-RANSAC filtering of the feature matches. Left image shows the raw feature matches, which contain some erratic matches that do not fit the epipolar model. On the right, AC-RANSAC filtered matches are shown, after the technique eliminates inaccurate matches.

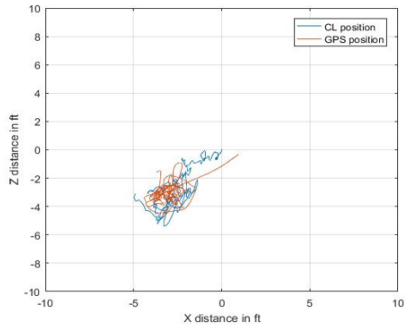

Fig. 6.

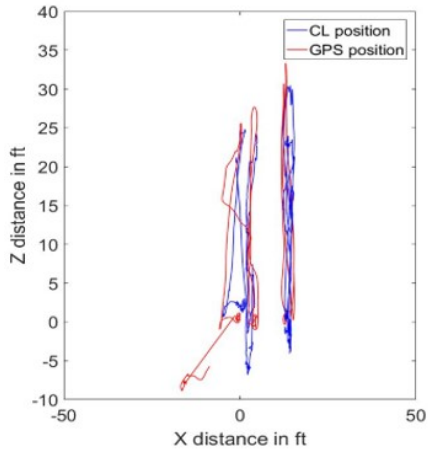

Fig. 7. Comparison between GPS and CL positions for two MAVs. Barring the slight drift observed in the GPS sensor measurements, the two algorithms are mostly in agreement.

to hover along with the F450. For comparison, we plot both the GPS position and the position from the CL algorithm, which are in agreement with each other.

Figure 7 plots the trajectories of the positions obtained from the CL algorithm and the GPS position as the vehicles were randomly flown in a forward-backward motion. Again, it can be seen that the trajectory motions were mostly in agreement: we observe a slight drift in the GPS estimate in the trajectory of the X8 vehicle. In figure 8, we show the full 3D plots from the previous flight for both MAVs as computed by the CL algorithm. Figure 9 shows the 3D plots from two MAVs as they're flown through side-side trajectories.

CONCLUSIONS

Through this paper, we present our approach for collaborative localization for multiple MAVs. This localization method relies on vision based sensing, which we achieve through monocular cameras installed onboard the MAVs. We utilize feature detection and matching as our primary source of information for estimating relative poses between vehicles through multiple view geometry, and subsequently metric poses through 3D-2D correspondences, thus eliminating the need to perform feature matching at every instant. We show

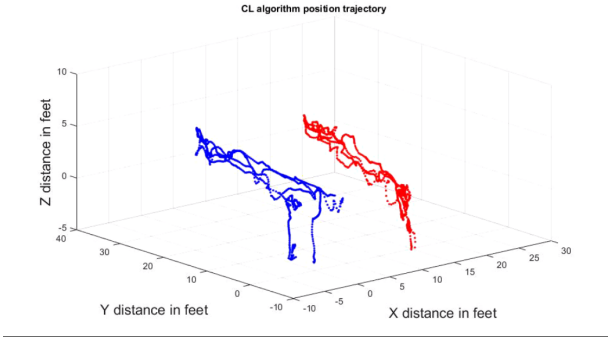

Fig. 8. Pose data of two MAVs flying forward-backward trajectories as computed by the CL algorithm

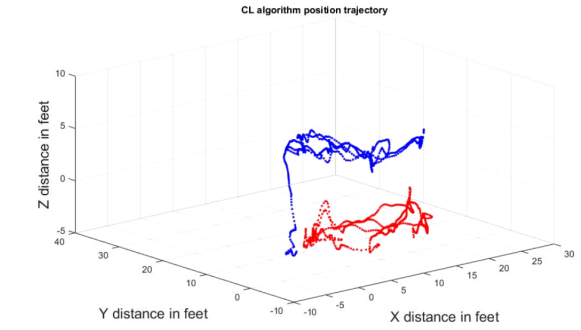

Fig. 9. Pose data of two MAVs flying side-side computed from the CL algorithm

results from simulation as well as real tests, which validate the algorithm and its applicability to groups of MAVs.

For our future work, we aim to investigate the possibility of implementing this method in real-time. We are currently working on implementing GPU acceleration to speed up computationally intensive processes such as feature detection, as well as a filtering framework that is capable of fusing individual and relative measurements that allows for a higher degree of collaboration between the MAVs. Furthermore, we aim to perform more experiments with an even higher number of vehicles in high fidelity simulators such as Microsoft AirSim (Ref. 21), which provides photorealistic environments, as well as through real experiments.

REFERENCES

[1]Honegger, D., Meier, L., Tanskanen, P. and Pollefeys, M., "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," pp. 1736-1741,2013 IEEE International Conference on Robotics and Automation (ICRA), 2013.

[2]Jones, E.S. and Soatto, S., "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach,", *The International Journal of Robotics Research,* Vol. 30, (4), 2011, pp. 407-430

[3]Klein, G. and Murray, D., "Parallel Tracking and Mapping for Small AR Workspaces," pp. 225-234, 6th IEEE and ACM

International Symposium on Mixed and Augmented Reality (ISMAR), 2007.

[4]Pire, T., Fischer, T., Civera, J., De Cristoforis, P. and Berlles, J.J., "Stereo parallel tracking and mapping for robot localization," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2015.

[5]Zhang, J., Kaess, M. and Singh, S., "Real-time depth enhanced monocular odometry," pp. 4973-4980, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2014.

[6]Li, B., Heng, L., Lee, G.H. and Pollefeys, M., "A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle.," pp. 1595-1601, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2013.

[7]Faessler, M., Mueggler, E., Schwabe, K. and Scaramuzza, D., "A monocular pose estimation system based on infrared LEDs," pp. 907-913,2014 IEEE International Conference on Robotics and Automation (ICRA), 2014.

[8]Dhiman, V., Ryde, J. and Corso, J.J., "Mutual localization: Two camera relative 6-dof pose estimation from reciprocal fiducial observation," pp. 1347-1354 , 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2013.

[9]Wang, X., Sekercioglu, Y.A. and Drummond, T., "Vision based cooperative pose estimation for localization in multi-robot systems equipped with rgb-d cameras,", *Robotics,* Vol. 4, (1), 2014, pp. 1-22.

[10]Zhou, X.S. and Roumeliotis, S., "Robot-to-robot relative pose estimation from range measurements.,", *IEEE Transactions on Robotics,* Vol. 24, (6), 2008, pp. 1379-1393.

[11]Butzke, J., Gochev, K., Holden, B., Jung, E.J., and Likhachev, M., "Planning for a ground-air robotic system with collaborative localization," pp. 284-291, 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016.

[12]Piasco, N., Marzat, J., Sanfourche, M., "Collaborative localization and formation flying using distributed stereo-vision," 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016.

[13]Vemprala, S. and Saripalli S., "Vision based collaborative localization for multirotor vehicles," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.

[14]Yining, J., Yanxuan, W. and Ningjun, F., "Research on distributed cooperative control of swarm uavs for persistent coverage," pp. 1162-1167, 33rd Chinese Control Conference (CCC), China, July 2014.

[15]Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, Vol. 60, (2), 2004, pp. 91-110.

[16]Moisan, L., Moulon, P. and Monasse, P., "Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers," *Image Processing On Line*, Vol. 2, 2012, pp. 5673.

[17]Nister, D., "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, (6), 2004, pp. 756-770.

[18]Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.

[19]Bradski, G., "OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2236121, 2000.

[20]Moulon, P., Monasse, P., Marlet, R. and Others, "OpenMVG: An Open Multiple View Geometry Library," https://github.com/openMVG/openMVG.

[21]Shah, S., Dey, D., Lovett. C. and Kapoor, A., "Aerial Informatics and Robotics Platform", Microsoft Research MSR-TR-2017-9.