Vision-based precision localization of UAVs for sensor payload placement and pickup for field monitoring applications

Zhou, Hao, Lynch, Jerome, Zekkos, Dimitrios

Hao Zhou, Jerome Lynch, Dimitrios Zekkos, "Vision-based precision localization of UAVs for sensor payload placement and pickup for field monitoring applications," Proc. SPIE 10970, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019, 1097007 (27 March 2019); doi: 10.1117/12.2516049

Event: SPIE Smart Structures + Nondestructive Evaluation, 2019, Denver, Colorado, United States

Vision-based Precision Localization of UAVs for Sensor Payload Placement and Pickup for Field Monitoring Applications

Hao Zhou[a], Jerome Lynch[*a,b], Dimitrios Zekkos[a]
[a]Department of Civil and Environmental Engineering
[b]Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109-2125, USA

ABSTRACT

Due to their mobility and autonomy, unmanned aerial vehicles (UAVs) provide an unprecedented opportunity to perform data gathering in a wide array of civil engineering applications such as visual inspection of infrastructure. Given their versatility, the role of UAVs can be expanded by leveraging their autonomous operations to deploy wireless sensing resources. This can be especially valuable in numerous field applications such as shear wave velocity (V_s) assessment of the subsurface. This study explores the feasibility of automating the autonomous placement and pickup of wireless geophone sensors using UAVs for multichannel analysis of surface waves (MASW) for subsurface characterization. Typically, autonomous navigation of UAVs is based on the fusion of inertial sensors and GPS to control the UAV flight trajectory. However, this approach is not sufficiently accurate for missions requiring precision placement and pickup of payloads (such as sensors). Hence, computer vision using fiducial markers can be used to augment traditional inertial sensing to add accuracy to the localization of the UAV relative to payloads. In this study, we use a set of fiducial markers of varying sizes as tracking targets during navigation missions. Pose information extracted from the marker images are integrated into a sensor fusion controller based on the Kalman filter. The work conducts field validation of the proposed computer vision navigation method showing accuracy of the UAV landing on a user defined target within 10 cm; as the UAV descends, smaller fiducial markers are shown to increase the precision of the UAV placement on the ground.

Keywords: Vision-based object tracking, aerial grasping, precision landing, Kalman filter, UAV, drone, computer vision

1. INTRODUCTION

Due to their mobility, UAVs equipped with various onboard sensors have become a popular platform for a wide range of civil engineering applications that traditionally require significant human resources to deploy. For example, use of UAVs for visual construction monitoring and structural condition assessment has resulted in reduction in the cost and effort associated with these otherwise manual processes[1]. Immediate post-disaster infrastructure surface characterization and 3D model construction have been made possible by capturing large collections of high-resolution images using a UAV, particularly in hard-to-reach sites[2]. UAVs carrying environmental sensors can also be used in cooperation with static ground sensor nodes to collect more reliable data for event detection (*e.g.*, fires)[3]. To date, the field has still primarily used UAVs with optical cameras to automate human visual inspection. Also, the majority of UAV-based field monitoring tasks are operated by human pilots which requires expertise in flying, especially when flying UAVs in dangerous environments.

This study explores expanding the role of the UAV in field monitoring applications through the use of autonomous computer vision-based object detection techniques and aerial grasping of sensing payloads. By allowing UAVs to interact directly with their environment, the research expands the reach of UAVs to be used in an entirely new set of field monitoring applications. The focus of the study is on using UAVs to communicate with and autonomously reconfigure an array of geophones to execute shear wave velocity (V_s) measurements using seismic surface waves with the intent to characterize the stiffness of the subsurface. This work is a continuation of the author's prior work on proof-of-concept V_s measurements using a UAV to deploy an impulsive source with the UAV controlling wireless sensor networks (WSNs) to

[*] email: jerlynch@umich.edu; address: 2060 G. G. Brown Building, Ann Arbor, MI 48109-2125, USA; phone: +1 (734) 615-5290; fax: +1 (734) 763-9244.

Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019, edited by
Jerome P. Lynch, Haiying Huang, Hoon Sohn, Kon-Well Wang, Proc. of SPIE Vol. 10970, 1097007
© 2019 SPIE · CCC code: 0277-786X/19/$18 · doi: 10.1117/12.2516049

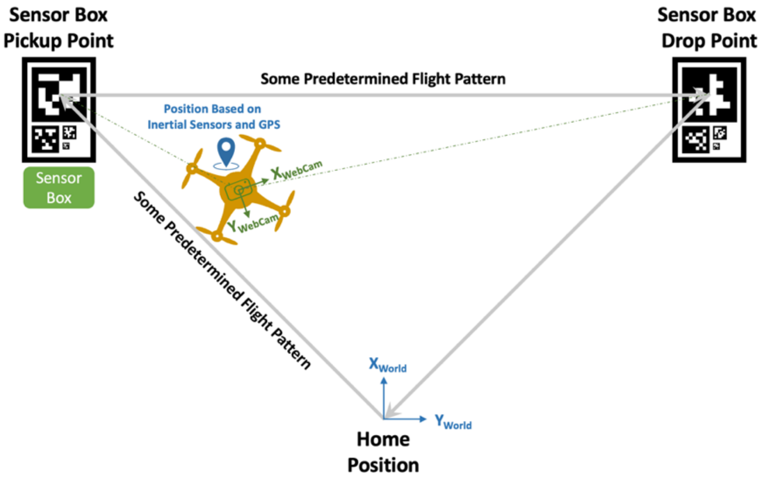

Figure 1. Experiment diagram showcasing the autonomous sensor pickup and replacement using a UAV.

measure ground motions[4,5,6]. While the previous work involved a human pilot flying the UAV, the work was successful in presenting a fully integrated MASW monitoring system based on the use of a UAV.

When automating the UAV pickup/release of sensor resources, two challenges arise: object tracking and aerial grasping. Object tracking requires the relative position of the sensor (*i.e*., object) to the UAV to be known *a priori*. UAVs equipped with IR-LOCK sensors can reliably track and land around an IR beacon, but with a low accuracy of 30 cm[7]. Tracking systems relying on external devices to communicate with UAVs are cumbersome and might get jammed from time to time. Modern computer vision-based tracking methods are a promising alternative. Bähnemann, *et al*.[8] used RGB images and a Kalman filter to detect the 3D position and 2D horizontal velocities of the observed objects. Nguyen, *et al.*[9] proposed a unique landing pattern design that enables robust tracking in both daytime and nighttime. There is a large amount prior work that have utilized marker-based detection algorithms for the autonomous landing and position control of UAVs[10,11,12]. In this study, the UAV determines the relative position of the to-be-collected sensor object through the use of the AprilTag[13] fiducial marker system. Unlike most of the aforementioned background works that largely rely on one fiducial marker, this work adopts a set of AprilTags of different sizes to help the UAV detect the tags from a variety of distances and to enhance the accuracy of placing the UAV over an object for pickup.

For aerial grasping, a UAV mounted gripper needs to be lightweight due to restrictions on flight times associated with heavy payloads. The flight dynamics of the UAV might also be altered with the gripper and payloads the gripper picks up. A feedback control solution is required to successfully execute the precision placement of the UAV to grasp a stationary object. Different approaches to address these challenges exist in the literature. Thomas, *et al.*[14] analyzed the dynamics of a quadrotor equipped with a servomotor-driven gripper claw and designed control laws for the coupled system to achieve avian-like high-speed object retrieval. Gawel, *et al.*[15] proposed a novel gripper design that features an electro-permanent magnet coupled with a passively compliant mechanical structure, which could be used to pick up objects with partly ferrous surface. Several groups have presented the autonomous assembly of small structures using a flock of autonomous flying agents equipped with grippers[16,17]. However, most of the prior research focused on aerial grasping have been performed indoors where an accurate motion capture localization system[18] is used to provide reliable measurement of UAV pose. In an outdoor environment, it is difficult to achieve high positioning accuracy of a UAV due to environmental disturbances such as wind.

In this study, a UAV is used for autonomously picking up and dropping a sensor box as shown in Figure 1. The aerial grasping problem this research presents is simplified in two ways. First, an electro-permanent magnet gripper is integrated in a UAV to pick up sensor enclosures with a steel sheet on its top surface. An advantage of magnetic gripping is that it allows for a slightly higher tolerance on position error due to the magnetic field radiating a small distance away from the UAV. During sensor placement and pickup, the UAV will naturally encounter turbulence as it gets close to the ground, which makes the task difficult. The work addresses the challenge by separating aerial grasping into two steps: first the precision landing on top of the sensor box followed by magnetic gripping while the UAV is landed. The sensor pickup and drop-off points are both defined by a set of four unique AprilTags sitting on the ground. The experiment starts with the UAV executing a predetermined fight trajectory using inertial sensors and GPS. Once the onboard camera detects the

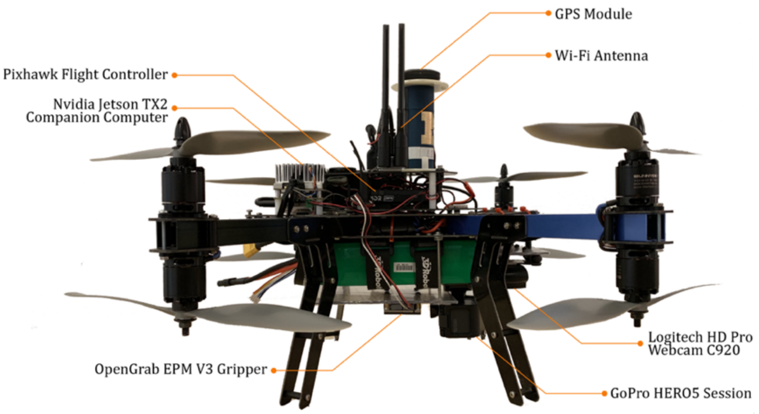

Figure 2. Customized 3DR X8 octo-rotor UAV.

AprilTag pattern on the ground at the pickup point, the UAV conducts a precision landing followed by a gripping attempt of the sensor. Precision landing is based on a feedback control algorithm that uses the images taken from the UAV camera and does not require GPS data. After successfully grasping the sensor, the UAV will take off and follow its predetermined flight trajectory until it finds the drop-off point at which point it does a precision landing to place the sensor. The proposed autonomous operation runs entirely onboard the UAV and features only a marked landing platform. This paper presents the automated sensor pickup and drop-off method for the UAV followed by experimental validation.

2. SYSTEM DESCRIPTION

2.1 Hardware System Description

A commercial off-the-shelf (COTS) 3D Robotics (3DR) X8 octo-rotor UAV is customized to fit the autonomous payload delivery application (Figure 2). With the same "X" shaped frame as a quad-rotor UAV, the 3DR X8 features 2 motors with inverted propellers on each of its 4 arms (and thus the name "X8"). While doubling the thrust of a normal quadcopter, the 8 motors on the X-frame also serve as a safety feature should one motor go out during flight. The frame is made of aluminum offering a sturdy but lightweight airframe. The UAV can carry nearly a 1 kg payload in addition to its self-weight. The UAV is powered by a 16,000mAh 4S LiPo battery that can provide approximately 15 minutes of flight.

Central to the UAV hardware system are two computing units: a Pixhawk flight controller and an Nvidia Jetson TX2 single board computer. The Pixhawk flight controller takes in measurements from an internal IMU (*i.e.*, accelerometers, gyroscopes, and magnetometers), GPS module, and barometer to estimate the UAV's pose (*i.e.*, position and orientation) while controlling the UAV flight for stable navigation. The Nvidia Jetson TX2 computer behaves like the "brain" of the UAV by automating high-level missions that may include complicated flight trajectories; the TX2 interfaces to the Pixhawk to send mission-specific navigation commands to the flight controller. The TX2 is equipped with a 256-core Pascal GPU, a dual-core Nvidia Denver 2.0 CPU, a quad-core ARM Cortex A57 CPU, and 8 GB 128-bit LPDDR4 memory. As a result, the TX2 is a powerful computing unit that is well-suited for expanding the functionality of the UAV. Computationally intensive tasks (*e.g.*, computer vision, path planning, and 3D modeling) are performed by the onboard TX2 thereby adding great intelligence to the behavior of the UAV.

Multiple devices can be connected to the two computing units (*i.e.*, Pixhawk and TX2). An OpenGrab EPM v3 electro-permanent magnet gripper[19] is mounted to the bottom surface of the UAV. Connected to the Pixhawk AUX OUT port, the gripper accepts grabbing/releasing commands (*i.e.*, two different pulse-width modulation (PWM) signals) encoded using the MAVLink[20] protocol. This small cuboid-shaped (about 4 x 4 x 2 cm^3) gripper is capable of securely holding up to 15 kg of cargo. A thin square steel sheet is attached to the top of the cargo which is a plastic sensor box in this study. The gripping and releasing of the sensor box is accomplished by the UAV delivering a short voltage pulse to an electromagnet inside the gripper, which reverses its field. Steady-state power consumption for the gripper is under 50 mW. However, a significant amount of power is required when first gripping or releasing the sensor box but this lasts of less than 1 second so it does not draw down the UAV battery much.

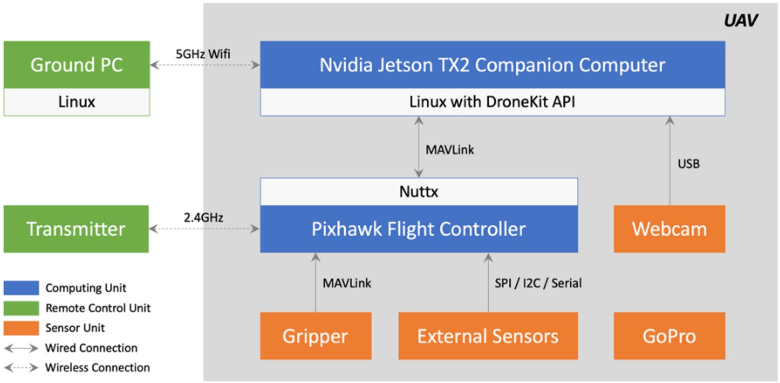

Figure 3. Hardware architecture for the UAV system.

Sitting on the front of the UAV is a downward facing Logitech HD Pro C920 web camera. This lightweight (approximately 50 g) camera features a full HD glass lens and is flanked by two microphones to capture stereo sound. Despite the camera's ability to record 1080p video clips, this study captures images with a lower resolution (640 x 480) due to the need to process in real-time on the limited computational resources of the TX2. The camera's 78-degree field of view is big enough to cover a wide view of the ground when the UAV is flying. The camera is soft mounted beneath the UAV using rubber bulb-type dampers to reduce vibration. The use of a gimbal is intentionally avoided to ensure the camera is offering an unfiltered view of the ground based on the UAV frame pose. The camera is attached to the Nvidia Jetson TX2 USB port. A second camera (GoPro HERO5 Session) is mounted to the UAV for video logging of the UAV flights.

The DroneKit-Python API[21] is installed on the Nvidia Jetson TX2 to allow the TX2 to send commands to the Pixhawk using the MAVLink protocol. A ground-based personal computer is used to communicate with Nvidia Jetson TX2 through a 5 GHz Wi-Fi interface; communication between the UAV and ground-based computer is for collecting data during flight. Pixhawk takes commands from Nvidia Jetson TX2 in the form of MAVLink messages. Although the flight is autonomous based on an automated state machine implemented on the TX2, a human pilot can retake control of the Pixhawk with a radio transmitter if needed. The complete system hardware architecture is shown in Figure 3.

2.2 Software System Description

The UAV software system is composed of two concurrently executing parts: code running on the Pixhawk flight controller and code running on the Nvidia Jetson TX2 companion computer. The flight stack running on the Pixhawk is the ArduCopter firmware from the ArduPilot autopilot project[22]. ArduPilot is an open source autopilot software suite with firmware supporting various unmanned vehicles including ArduCopter (for multirotor UAVs), ArduPlane (for fixed-wing aircraft), APMrover2 (for ground rovers), and ArduSub (for submarines). The ArduCopter firmware is most relevant to this work. The core of the ArduCopter firmware package is an extended Kalman filter (EKF)-based state estimation system with a cascade vehicle position and attitude control system. The EKF uses a series of sensor measurements observed over time, containing statistical noise and other inaccuracies, and produces an estimate of the UAV state (with 24 state variables including vehicle attitude, velocity, and position). There are many advantages of using an EKF estimator. For example, the EKF can offer very accurate estimates even in the face of bad sensor data (*e.g.*, large bias changes in sensor readings); it also can be used to estimate external variables such as wind gusts. An array of autonomous flight modes is supported by ArduCopter such as guided, loiter, and return-to-launch (RTL) operations. The control strategy for trajectory tracking is based on a cascade control architecture with a proportional-integral-derivative (PID) controller controlling vehicle attitude in an inner loop and a second PID controller to stabilize vehicle position in an outer loop to track a desired trajectory. Outputs from the cascade controller (*i.e.*, roll, pitch, yaw and throttle values) are converted to absolute motor outputs (*i.e.*, PWM values) and sent to electronic speed control (ESC) units that command the rotors.

The other significant part of the software system is implemented on the Nvidia Jetson TX2 companion computer. The main software is fundamentally a finite-state machine (FSM) designed for the payload delivery mission. A second background thread is written to constantly grab images from the web camera and feed them into the main thread. An AprilTag detection algorithm and a Kalman filter is run to compute and estimate the position of an AprilTag relative to the UAV. During precision landing, the TX2 constantly sends the desired landing position (*i.e.*, estimated position of a designated AprilTag) to the Pixhawk flight controller using the MAVLink protocol. This desired landing position serves

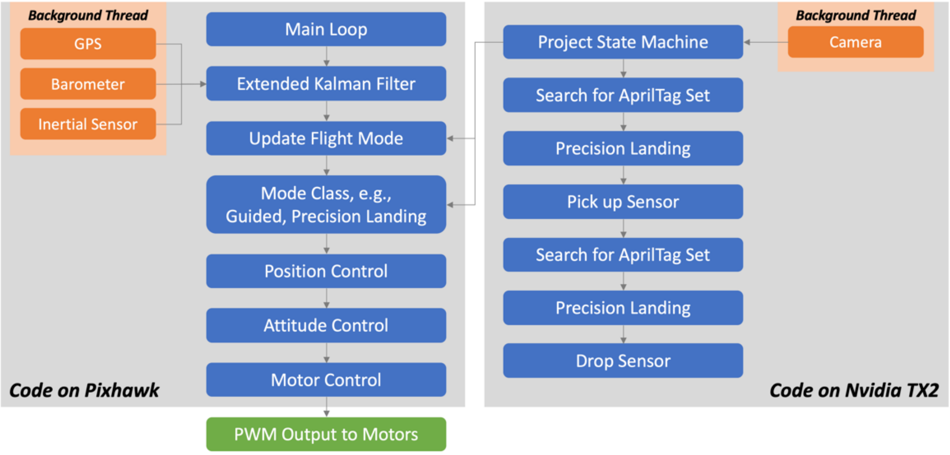

Figure 4. Software architecture for the UAV system: code on the Pixhawk and Nvidia TX2 computing platforms.

as the next navigational waypoint for the UAV and is directly used as the input to the Pixhawk controller. Figure 4 illustrates the entire software system architecture implemented for the UAV. It should be emphasized, that the contributions of this work lay mainly in the design and implementation of the finite-state machine implemented on the TX2 including the use of estimated UAV pose based on computer vision methods. In contrast, the Pixhawk was used as coded with some minor tuning of control parameters for precision control of the UAV.

3. METHODS

3.1 System Finite-State Machine

A finite-state machine (FSM) is an approach to partitioning a complicated system into a set of well-defined states with interconnection between them allowing for system actions to be programmed. An FSM approach to writing embedded firmware for autonomous systems is a scalable approach to building complex functionality with deterministic operations. In this study, mission management is implemented using a finite-state machine as designed in Figure 5. The complex operation of payload pickup and placement is divided into manageable pieces such as flying a preplanned flight trajectory, precision landing, and payload pickup, among others. Transitions between different sub-missions are clearly defined. A UAV starting from the "start" state would eventually reach the "end" state by either accomplishing or failing the mission.

3.2 Image Processing

3.2.1 Fiducial Marker Detection Algorithm

Visual fiducials are artificial landmarks designed to help robots comprehend their surroundings. In this study, fiducial markers are placed around the sensor box to help the UAV detect and navigate to the pickup/drop-off point. The fiducial detection system employed in the experiments is called AprilTag[13]. The AprilTag markers are square, high-contrast, QR-code-like 2D tags that have been designed to perform robustly in challenging detection scenarios including in poor light conditions and when occlusions occur. AprilTags are highly distinguishable from one another with reliable detection possible in low-resolution images. When coupled with the calibrated camera information (*i.e.*, camera intrinsic matrix and distortion vector), AprilTag can provide full six degrees-of-freedom (DOF) pose estimation for the camera relative to the tag. The tag detection algorithm begins by computing the gradient at each pixel in the image. Pixels with similar gradient directions and magnitude are clustered together and then filtered out to produce line segments. Candidate marker detections are created from these line segments and verified by the data payload decoded from them. Once pixel coordinates of the marker's center and its four corners are extracted, together with their true position in the world coordinate system (*i.e.*, the actual size of the tag is known), estimating the pose of the camera becomes a standard Perspective-n-Points (PnP) problem. Herein, the PnP problem is solved using solvePnP[23] implemented in Open Source Computer Vision Library (OpenCV)[24]. According to OpenCV documentation, solvePnP finds the pose that minimizes reprojection error based on a Levenberg-Marquardt optimization[25,26]. The strength of this optimization method is that it is fast and accurate. However, a weakness is that it offers one feasible solution each time but the uniqueness of the solution cannot be guaranteed[27]. More specifically,

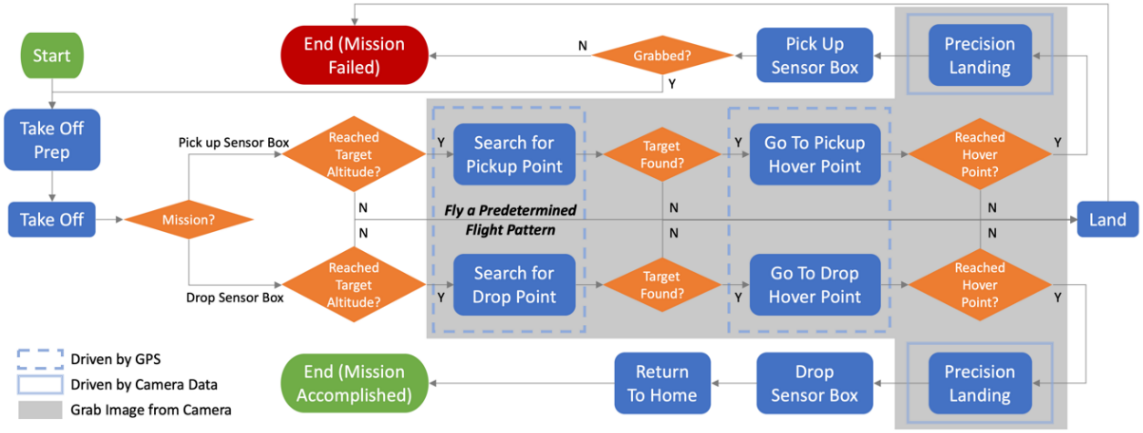

Figure 5. Finite-state machine for autonomous operation of a UAV engaged in sensor delivery.

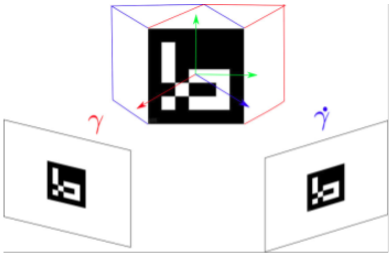

Figure 6. The ambiguity problem where the same marker projection could come from two poses based on the two cubes shown in red and blue.[28]

as shown in Figure 6, a marker could project the same pixel set in a camera for two different camera locations. In general, this ambiguity can be solved if the camera is near to the marker. However, when the marker is small in the field of view of the camera, the errors in the corner estimation grow and ambiguity arises as a problem.[28] To handle this ambiguity, a robust method to estimate the relative pose of the camera with respect to the AprilTag is proposed by using the UAV's attitude (quaternions) information.

3.2.2 Visual Servoing

Figure 7 presents the method for positioning the UAV using the AprilTag system. First, both vision and IMU provide estimates of the three DOF attitude of the camera. However, it is preferable to use attitude estimates from the IMU because of the sophisticated extended Kalman filter implemented in ArduPilot, which gives consistent and accurate results. In the meantime, the three DOF relative position of the camera produced by the solvePnP solver can be ambiguous as previously described. Thus, they cannot be used as measurement updates directly. Instead, the three DOF position is converted to a one DOF distance by calculating the Euclidean distance from the camera to the AprilTag (*i.e.*, $\|\overrightarrow{OM}\|$ in Figure 7). This one-dimensional Euclidean distance is robust because it is immune to pose ambiguity. Also, the verification experiments (described in Section 4.1) will show that the relative error of the Euclidean distance in well within 8%. To be clear, this Euclidean distance from the camera to the AprilTag serves as the only reliable measurement from the solvePnP algorithm.

Three main coordinate frames are required. The north-east-down (NED) frame, denoted "Ground" in Figure 7, is the frame of reference for the GPS measurements and absolute positioning. A camera frame, denoted by "Camera", is defined at the optical center of the camera with its z-axis pointing orthogonally towards the camera's focal plane. The UAV body frame, denoted as UAV, is defined as a coordinate frame fixed to the center of gravity of the UAV with its x-axis pointing to the head of the UAV, y-axis pointing to the UAV's right side, and z-axis pointing down to the UAV's back. The three coordinate systems and the transformations between them are illustrated in Figure 7. In order to get the relative position of the AprilTag from the camera, a representation of $\overrightarrow{OM}$ in the Ground frame of reference is needed. More specifically,

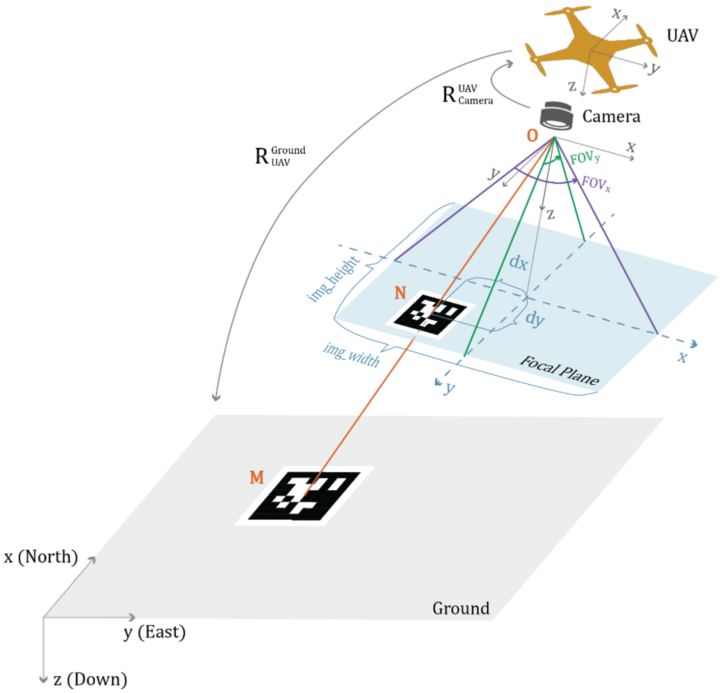

Figure 7. Diagram illustrating how to compute the relative position of the AprilTag from the UAV.

only the representation of $\overrightarrow{OM}$'s direction (*i.e.*, $\frac{\overrightarrow{OM}}{\|\overrightarrow{OM}\|}$) in the Ground frame of reference is needed, since its length $\|\overrightarrow{OM}\|$ is already known. The representation of $\frac{\overrightarrow{OM}}{\|\overrightarrow{OM}\|}$ in the Ground frame can be found as follows. First, let R_A^B represent the rotation matrix that transforms the coordinates of a point from frame A to frame B where the representation of a given vector $\vec{p}$ in frame A, p^A, and its representation in frame B, p^B, have the following relationship: $p^B = R_A^B p^A$. The representation of $\frac{\overrightarrow{OM}}{\|\overrightarrow{OM}\|}$ in the Camera frame of reference can be calculated based on the geometry shown in Figure 7:

$$x\_offset = \frac{FOV_x \cdot dx}{img\_width} \tag{1}$$

$$y\_offset = \frac{FOV_y \cdot dy}{img\_height} \tag{2}$$

$$\frac{\overrightarrow{OM}^{Camera}}{\|\overrightarrow{OM}^{Camera}\|} = \frac{\overrightarrow{ON}^{Camera}}{\|\overrightarrow{ON}^{Camera}\|} = Normalize([\tan(x\_offset) \quad \tan(y\_offset) \quad 1]^T) \tag{3}$$

where FOV_x and FOV_y are the camera's field of view in the x-axis and y-axis directions in radians, respectively. Also, dx and dy denote the distance from the center of the image to the center of AprilTag in terms of pixels. x_offset and y_offset represent the distance from the center of the image to the center of AprilTag in radians. The rotation matrix can then be used to find the representation of $\frac{\overrightarrow{OM}}{\|\overrightarrow{OM}\|}$ in the Ground frame of reference:

$$\frac{\overrightarrow{OM}^{Ground}}{\|\overrightarrow{OM}^{Ground}\|} = R_{UAV}^{Ground} R_{Camera}^{UAV} \frac{\overrightarrow{OM}^{Camera}}{\|\overrightarrow{OM}^{Camera}\|} \tag{4}$$

The transformation matrix R_{UAV}^{Ground} can be simply computed from the UAV's current attitude (quaternions). R_{Camera}^{UAV} relies on how the camera is mounted to the UAV which establishes the geometric relationship between the camera and the UAV.

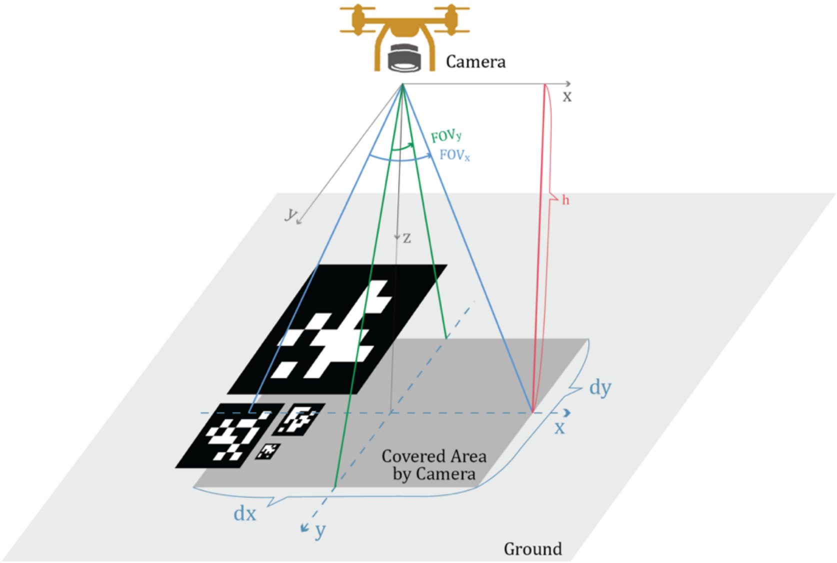

Figure 8. Diagram illustrating limited FOVs of camera (in this case, only the 2 smallest AptilTags can be detected).

3.2.3 Vision-IMU Synchronization

Due to the vision-based position measurement depending on the UAV's current attitude, vision-IMU synchronization is required. A time synchronization procedure between the Pixhawk and Nvidia Jetson TX2 is carried out first. MAVLink messages with timestamps are sent and received back and forth between the flight controller and the companion computer to compute the time differences between them. While continuously acquiring images from the web camera, the background image capture thread on the Nvidia Jetson TX2 also records the image's shutter timestamp and converts it to match Pixhawk's system clock. AprilTag detection algorithm then kicks in to extract useful information from the image (*i.e.*, $\|\overrightarrow{OM}\|$, x_offset, and y_offset). This visual data together with IMU data with the closest timestamp are used together to produce relative position estimations.

3.2.4 Multi-AprilTag Landing Pattern

During a precision landing mission, the area on the ground covered by the downward facing camera becomes smaller as the UAV descends. The maximum lateral and longitudinal distance on the ground covered by the camera, dx and dy, can be calculated based on the height of the UAV h and the camera's field of view, FOV_x and FOV_y (Figure 8):

$$dx = 2h\tan\frac{FOV_x}{2} \tag{5}$$

$$dy = 2h\tan\frac{FOV_y}{2} \tag{6}$$

As the UAV gets closer to the ground, an AprilTag must be small enough to stay in the camera's FOV. However, too small an AprilTag would be undetectable to a distant camera due to limited image resolution. To resolve this problem, a set of AprilTags of different sizes is used to help constantly navigate the UAV during its landing. A pattern containing four AprilTags as shown in Figure 8 is designed. Several AprilTags might be detected at the same time from a single image. We simply set the AprilTag with the smallest size as the active one for flight control and calculate the relative position of the camera with respect to the active AprilTag following the procedure stated above.

3.3 Kalman-Filter-Based Landing Position Estimation

In order to consistently fuse AprilTag measurements, a Kalman filter is used. Relative position estimation is executed entirely in the Ground (NED) frame of reference. Two independent Kalman filters are used: one for estimations in the North (x) direction and the other for those in the East (y) direction. Without loss of generality, how the Kalman filter in the North (x) direction is formed will be presented. The estimated states are the relative position, x^{rel}, and velocity, v^{rel},

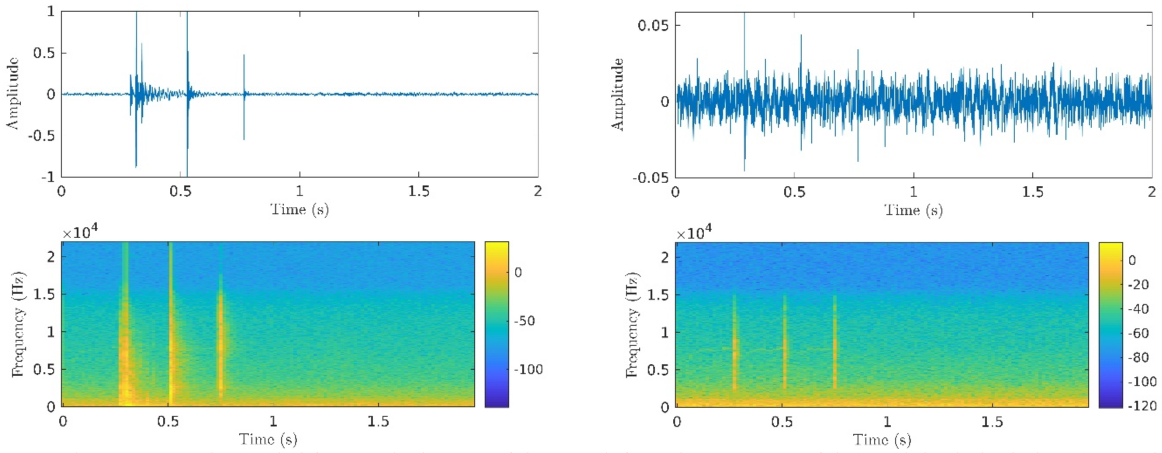

Figure 9. Sound recorded for a typical successful grasp (left) and an unsuccessful one (right) in both time (up) and frequency (down) domain.

of the AprilTag of interest with respect to the UAV in the x direction ($x = [x^{rel}\ v^{rel}]^T$), the controlled input, u, is the negative of the UAV's velocity change in the x direction in δt ($u = [0\ \delta v^{rel}]^T$), the measurement is the relative distance of the AprilTag from the vehicle in the x direction ($z = {x_{meas}}^{rel}$), and the discrete-time state space model is:

$$\begin{cases} \begin{bmatrix} x_{k+1}^{rel} \\ v_{k+1}^{rel} \end{bmatrix} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^{rel} \\ v_k^{rel} \end{bmatrix} + \begin{bmatrix} 0 \\ \delta v_k^{rel} \end{bmatrix} + \begin{bmatrix} 0 \\ \delta {v_{noise}}_k^{rel} \end{bmatrix} \\ {x_{meas}}_k^{rel} = [1 \quad 0] \begin{bmatrix} x_k^{rel} \\ v_k^{rel} \end{bmatrix} + \delta {z_{noise}}_k^{rel} \end{cases} \tag{7}$$

The process noise $\delta {v_{noise}}_k^{rel}$ is set to be the UAV's accelerometer noise times δt, and the measurement noise $\delta {z_{noise}}_k^{rel}$ is assigned to 2% of the UAV's relative distance to AprilTag. A standard two-step process (*i.e.*, prediction and update) is applied to solve the Kalman filter. The Kalman filter in general produces more accurate estimations and brings robustness to the system. For example, during the UAV's landing process, if the camera fails to detect the AprilTag for any reason (*e.g.*, a blurry image) and thus no measurement update is available, the filter will only predict for this iteration and compensates in the next update step for the then longer time interval. This allows the system to estimate the landing position for several seconds without new vision measurements being required.

3.4 Gripper Feedback

Out of the box, the EPM gripper does not provide any feedback information about whether a grab attempt is successful. However, confirmation of pickup is essential for autonomous missions to ensure the efficiency of field operations. Teams attending the Mohamed Bin Zayed International Robotics Challenge (MBZIRC 2017) provided a solution by installing Hall effect sensors to measure the magnetic field change when a ferrous object is attached[29,30]. In this study, a novel method that takes advantage of sound differences between a successful gripping and a failure is adopted. In the case of a successful gripping action, the magnetic force generated by the gripper keeps drawing the ferrous object closer until a firm contact between them is created. A loud sound is generated when this tight contact is established. However, for an unsuccessful gripping attempt, there is no contact and thus only low-level background noise is recorded. When a pickup is attempted, the UAV is programmed to record the sound using the microphones inside the web camera for 2 second after a gripping command is sent to the gripper. Figure 9 shows the soundwaves in both time and frequency domain for a typical successful grasp and an unsuccessful one. The three peaks in the plots arise from capacitors inside the electro-permanent magnet charging three times to create a strong magnetic field.

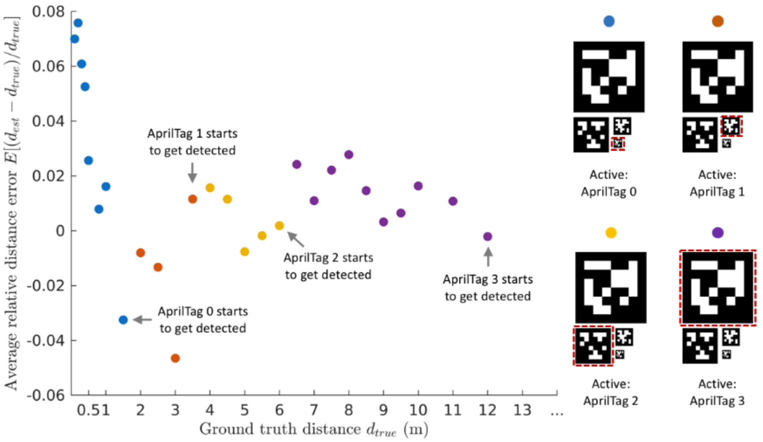

Figure 10. Relative distance error in different scenarios.

4. EXPERIMENTS AND RESULTS

The goal of the study tests is to evaluate the feasibility of using a UAV to autonomously pick up and drop a sensor box in an outdoor environment. First, the accuracy of the AprilTag detection algorithm is assessed using the tag pattern designed. Outdoor experiments are performed in a RC model aircraft club in Scio, Michigan.

4.1 Accuracy of Multi-AprilTag Detection Algorithm

In order to study the accuracy of the multi-AprilTag detection algorithm, the web camera is placed in front of the multi-AprilTag pattern in different relative positions and orientations. Average relative distance error is measured with the results shown in Figure 10. The relative distance error is defined as the estimated Euclidean distance minus the ground truth Euclidean distance normalized by the ground truth Euclidean distance. The most important observation from Figure 10 is that the relative distance error is well within 8% in all test cases and within 2% when the UAV is 4 m or more from the marker. The biggest AprilTag (tag 3) in the pattern can be detected roughly within 12 m while the smallest AprilTag (tag 0) can be detected from a distance no shorter than 0.1 m and no longer than 2 m.

4.2 Outdoor Experiments Results

Figure 11 shows the flight of the UAV in the field. Figure 12 shows three of the UAV flight trajectories during the precision landing part of the flight. The UAV oscillates in the North direction while it remains relatively stable in the East direction. This is because of the switch between different active AprilTags. There is a relatively large change of AprilTag center positions in the North direction, while the change in the East direction is small. Figure 13 shows a sequence of images captured by the UAV web camera during precision landing. Information extracted from the images (*i.e.*, x_offset, y_offset, and $\|\overrightarrow{OM}\|$) is shown in green in the top left corner of each frame. The UAV is able to land on top of the smallest AprilTag with an accuracy less than 10 cm. During the tests, the Nvidia Jetson TX2 companion computer is able to handle the computationally intensive AprilTag detection algorithm at a maximum frequency of 15 Hz. State variables estimated by the implemented Kalman filter are shown in Figure 14. Figure 15 compares these estimated relative distances from the UAV to the AprilTag using the Kalman filter to those computed directly from the solvePnP algorithm using GoPro video images. They follow the same trend, while the Kalman filter provides more conservative estimates.

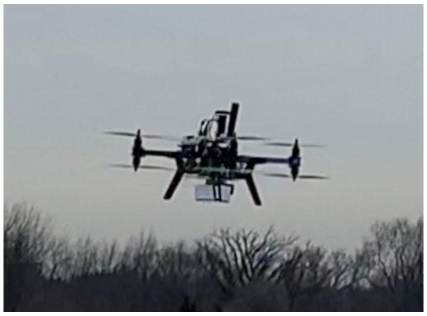

Figure 11. 3DR X8 carrying the sensor box to the drop location.

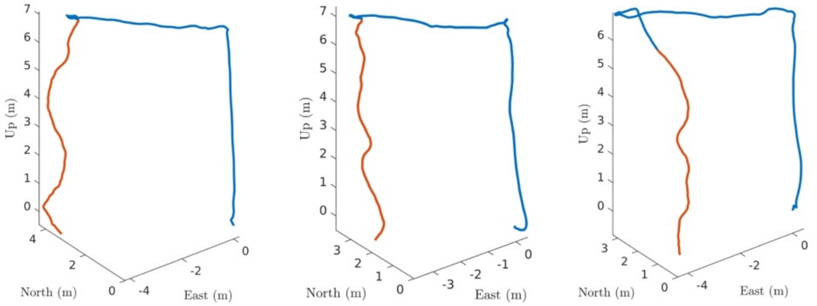

Figure 12. Different trajectories showcasing precision landing. (Blue: searching for pickup location; red: precision landing)

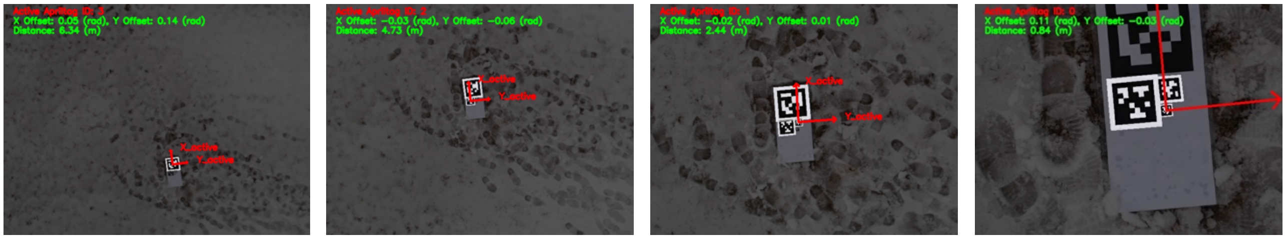

Figure 13. A sequence of images captured by the web camera during precision landing.

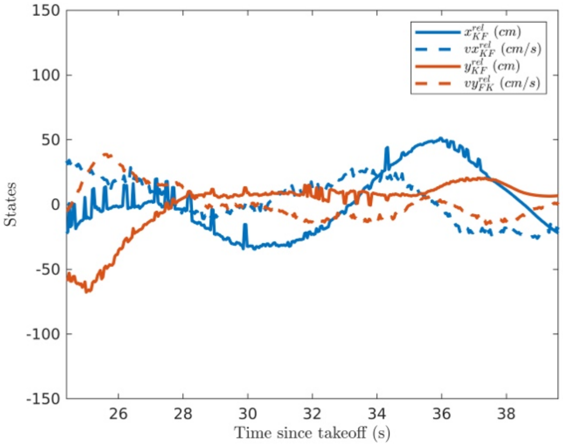

Figure 14. State estimation results using Kalman filter.

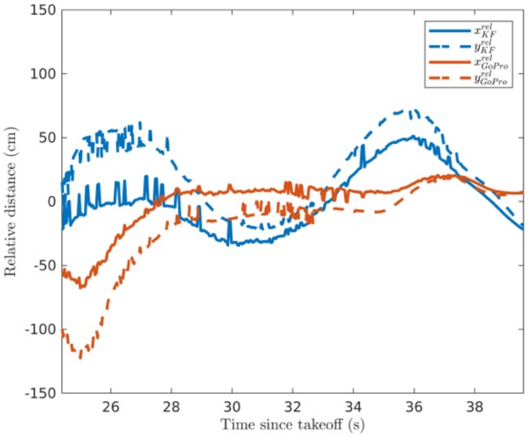

Figure 15. Comparison between Kalman filter state estimations and direct results from solvePnP.

5. CONCLUSIONS

The study presented herein explores the development of UAVs as an intelligent agent capable of picking up and deploying payloads autonomously for field monitoring applications. With the aid of visual fiducial markers, the proposed UAV system can autonomously track, perform landing maneuvers, and pickup/drop desired payloads using only onboard sensors and computing units. Transitions between different missions like searching for a pickup location and grabbing payloads is accomplished using a reliable finite-state machine. Precision positioning of a UAV in the outdoor environment is made possible by the integration of a customized fiducial marker pattern, a robust vision-IMU coupled estimation method and a Kalman filter. The study reveals excellent accuracy less than 10 cm in the placement of the UAV on the ground for sensor placement. Picking up and releasing payloads is successfully implemented using an electro-permanent magnet gripper. A novel approach to identify a successful gripping is also presented. The study proves the feasibility of using a UAV as a sensor resources delivery platform and is geared towards a fully autonomous UAV system that could intelligently reconfigure sensor resources based on processed results of data collected from the sensors in real-time.

6. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation (NSF) under Grant Numbers CMMI-1362975, CCF-1442773, and ECCS- 1446521. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Ham, Y., Han, K. K., Lin, J. J. and Golparvar-Fard, M., “Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (UAVs): a review of related works,” Vis. Eng. **4**(1), 1 (2016).

[2] Zekkos, D., Greenwood, W., Lynch, J., Manousakis, J., Athanasopoulos-Zekkos, A., Clark, M., Cook, K. L. and Saroglou, C., “Lessons learned from the application of UAV-enabled structure-from-motion photogrammetry in geotechnical engineering,” ISSMGE Int. J. Geoengin. Case Hist. **4**(4), 254–274 (2018).

[3] Erman, A., Hoesel, L., Havinga, P. and Wu, J., “Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management,” IEEE Wirel. Commun. **15**(6), 38–46 (2008).

[4] Zekkos, D., Lynch, J. D., Sahadewa, A., Hirose, M. and Ellis, D., “Proof-of-concept shear wave velocity measurements using an unmanned autonomous aerial vehicle,” Geo-Congress 2014 Tech. Pap., 953–962, American Society of Civil Engineers, Reston, VA (2014).

[5] Greenwood, W., Zhou, H., Zekkos, D. and Lynch, J. P., “Experiments using a UAV-deployed impulsive source for multichannel analysis of surface waves testing,” Geotech. Earthq. Eng. Soil Dyn. V, 443–451, Austin (2018).

[6] Zhou, H., Hirose, M., Greenwood, W., Xiao, Y., Lynch, J., Zekkos, D. and Kamat, V., “Demonstration of UAV deployment and control of mobile wireless sensing networks for modal analysis of structures,” 20 April 2016, 98031X, International Society for Optics and Photonics.

[7] “Precision landing and loiter with IR-LOCK.”, <http://ardupilot.org/copter/docs/precision-landing-with-irlock.html> (7 March 2019).

[8] Bahnemann, R., Schindler, D., Kamel, M., Siegwart, R. and Nieto, J., “A decentralized multi-agent unmanned aerial system to search, pick up, and relocate objects,” 2017 IEEE Int. Symp. Safety, Secur. Rescue Robot., 123–128, IEEE (2017).

[9] Nguyen, P. H., Kim, K. W., Lee, Y. W. and Park, K. R., “Remote marker-based tracking for UAV landing using visible-light camera sensor,” Sensors (Basel). **17**(9) (2017).

[10] Chaves, S. M., Wolcott, R. W. and Eustice, R. M., “NEEC research: Toward GPS-denied landing of unmanned aerial vehicles on ships at sea,” Nav. Eng. J. (2015).

[11] Ling, K., “Precision landing of a quadrotor UAV on a moving target using low-cost sensors” (2014).

[12] Borowczyk, A., Nguyen, D.-T., Phu-Van Nguyen, A., Nguyen, D. Q., Saussié, D. and Ny, J. Le., “Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle,” IFAC-PapersOnLine **50**(1), 10488–10494 (2017).

[13] Olson, E., “AprilTag: A robust and flexible visual fiducial system,” Proc. - IEEE Int. Conf. Robot. Autom. (2011).

[14] Thomas, J., Polin, J., Sreenath, K. and Kumar, V., “Avian-inspired grasping for quadrotor micro UAVs,” Vol. 6A 37th Mech. Robot. Conf., V06AT07A014, ASME (2013).

[15] Gawel, A., Kamel, M., Novkovic, T., Widauer, J., Schindler, D., von Altishofen, B. P., Siegwart, R. and Nieto, J., "Aerial picking and delivery of magnetic objects with MAVs," 2017 IEEE Int. Conf. Robot. Autom., 5746–5752, IEEE (2017).

[16] Lindsey, Q., Mellinger, D. and Kumar, V., "Construction of cubic structures with quadrotor teams," Robot. Sci. Syst. **VII** (2011).

[17] Augugliaro, F., Lupashin, S., Hamer, M., Male, C., Hehn, M., Mueller, M. W., Willmann, J. S., Gramazio, F., Kohler, M. and D'Andrea, R., "The flight assembled architecture installation: cooperative construction with flying machines," IEEE Control Syst. **34**(4), 46–64 (2014).

[18] "VICON Motion Capture Systems.", <https://www.vicon.com/> (7 March 2019).

[19] "OpenGrab EPM v3, an electropermanent magnet.", <https://github.com/Zubax/opengrab_epm_v3>.

[20] "MAVLink, micro air vehicle message marshalling library, the standard communication protocol for drones.", <https://github.com/mavlink/mavlink>.

[21] "DroneKit-Python, library for communicating with drones via MAVLink.", <https://github.com/dronekit/dronekit-python>.

[22] "ArduPilot, open source autopilot.", <https://github.com/ArduPilot/ardupilot>.

[23] "Camera calibration and 3D reconstruction, OpenCV 2.4.13.7 documentation.", <https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=solvepnp#bool%2520solvePnP(InputArray%2520objectPoints,%2520InputArray%2520imagePoints,%2520InputArray%2520cameraMatrix,%2520InputArray%2520distCoeffs,> (7 March 2019).

[24] Bradski, G., "The OpenCV library," Dr. Dobb's J. Softw. Tools (2000).

[25] Levenberg, K., "A method for the solution of certain non-linear problems in least squares," Q. Appl. Math. **2**(2), 164–168 (1944).

[26] Marquardt, D. W., "An algorithm for least-squares estimation of nonlinear parameters," J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1963).

[27] López-Cerón, A. and Cañas, J. M., "Accuracy analysis of marker-based 3D visual localization," XXXVII Jornadas Autom. Work. (2016).

[28] Rafael Muñoz Salinas., "ArUco: An efficient library for detection of planar markers and camera pose estimation," <https://docs.google.com/document/d/1QU9KoBtjSM2kF6ITOjQ76xqL7H0TEtXriJX5kwi9Kgc/edit> (7 March 2019).

[29] Loianno, G., Spurny, V., Thomas, J., Baca, T., Thakur, D., Hert, D., Penicka, R., Krajnik, T., Zhou, A., Cho, A., Saska, M. and Kumar, V., "Localization, Grasping, and Transportation of Magnetic Objects by a Team of MAVs in Challenging Desert-Like Environments," IEEE Robot. Autom. Lett. **3**(3), 1576–1583 (2018).

[30] Bähnemann, R., Pantic, M., Popović, M., Schindler, D., Tranzatto, M., Kamel, M., Grimm, M., Widauer, J., Siegwart, R. and Nieto, J., "The ETH-MAV Team in the MBZ International Robotics Challenge," J. F. Robot. **36**(1), 78–103 (2019).