



Safe2Ditch: Emergency Landing for Small Unmanned Aircraft Systems

Parker C. Lusk*

Brigham Young University, Provo, Utah 84602

Patricia C. Glaab[†] and Louis J. Glaab[‡]

NASA Langley Research Center, Hampton, Virginia 23681

and

Randal W. Beard[§]

Brigham Young University, Provo, Utah 84602

DOI: 10.2514/1.1010706

Emergency landing is a critical safety feature for the expected increase of autonomous vehicles in the airspace. This paper presents Safe2Ditch, an autonomous crash management system for landing small unmanned aerial vehicles in populated environments. Using a prepopulated database, the highest-rated landing site is selected. As the vehicle progresses toward the selected landing site, a camera is used to image the area around the landing site. Using a recently developed visual multiple target tracker, moving ground obstacles are tracked and geolocated using a flat Earth model. These inertial target estimates are used to revise the landing site and allow the vehicle to land quickly in the presence of non-cooperative obstacles. The complete system is demonstrated through 2000 Monte Carlo simulations and 16 hardware flight tests.

I. Introduction

TECHNOLOGY and computing capabilities continue to allow small unmanned aircraft systems (sUAS) to be used in sophisticated applications such as infrastructure monitoring, medical services delivery, search and rescue, natural disaster assessment, package delivery, and atmospheric observation. Small UAS are well suited for these tasks because of their low cost, quick deployment time, and ability to perform hazardous work. However, to fully realize the economic and societal benefits of these civil and commercial applications, a structure must be put in place to integrate sUAS into the National Airspace System (NAS) [1].

Many efforts are currently being made in this regard, in terms of both policy and technology [2–4]. Most notable is the UAS Traffic Management (UTM) effort being led by NASA and the U.S. Federal Aviation Administration (FAA) [5]. UTM is a traffic management ecosystem for low-altitude (below 400 ft, 120 m) sUAS operations in uncontrolled airspace. Its purpose is to connect UAS operators with real-time data from other UAS in the vicinity, weather and terrain data, and an overall NAS regulator (e.g., the FAA). This architecture aims to maximize information flow between users, giving flexibility to operators and their sUAS operations when working in low-risk or isolated areas and providing support and structure when multiple sUAS operate in densely populated areas.

Integration into the NAS as suggested by UTM requires that unmanned systems flying beyond visual line of sight (BVLOS) have much more capable on-board autonomy—not only to carry out specific missions, but also to enable safety-critical components such as sense and avoid, forced landing, operation in GPS-degraded environments, and communication-loss mitigation. These safety concerns must be fully addressed and standardized across sUAS in the NAS. The main contribution of this paper is an emergency landing scheme for sUAS operating at low altitudes in the NAS.

For manned aircraft, emergency landing is a crucial aspect of flight safety. When technical problems such as engine failure or structural damage occur, the pilot must quickly select a suitable landing site that might not be at a designated airstrip. Using communication with air traffic control, a safe landing zone is chosen that is clear of people and other obstacles. In the case of uncontrolled general aviation airspace, such as FAA Class G airspace, it is the pilot's responsibility to visually locate a safe landing site clear of people and other obstacles. Performing forced landing maneuvers at low altitude for unmanned aircraft requires the on-board system to be able to perceive the environment and make decisions on a landing site autonomously. Additionally, the size, weight, and power constraints of sUAS prevent the use of double- or triple-mode redundancy as is common in manned aircraft. This results in the need of a quick reaction time for sUAS emergency landing systems.

A successful model for autonomous interaction with the environment can be found in the sensory-based decision-making process known as *situation awareness* [6]. Originating in the military aircraft pilot community, situation awareness is defined as the human process of perceiving details in the environment, comprehending how those details affect the current goal, and projecting that comprehension to what will happen in the near future [7]. This process of situation awareness allows human operators to make critical decisions in a timely and effective manner. Similarly, adding a sense of situation awareness in autonomous vehicles allows them to operate effectively in dynamic environments [8]. For example, McAree and Chen [9] discuss adding artificial situation awareness to UAS by using automated dependent surveillance-broadcast (ADS-B) to cooperatively sense other agents while landing at prepared sites.

To better discuss the need for increased autonomy and situation awareness, we define the sUAS emergency landing problem through the following steps.

1) *Event detection*. During normal operation of an sUAS mission, an emergency event may be received from various sources. In the context of NAS integration, the UTM system may broadcast a land directive for the purposes of aircraft deconfliction or other safety reasons. Other examples of emergency events include vehicle fault detection such as low power and damaged motors.

Received 19 November 2018; revision received 19 June 2019; accepted for publication 20 June 2019; published online 15 July 2019. Copyright © 2019 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 2327-3097 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Graduate Researcher, Electrical and Computer Engineering; parkerclusk@gmail.com.

[†]Aeronautics Systems Analysis Branch, Systems Analysis and Concepts Directorate.

[‡]Assistant Branch Head, Aeronautics Systems Engineering Branch, Engineering Directorate.

[§]Professor, Electrical and Computer Engineering. Associate Fellow AIAA.

2) *Landing site selection.* During an emergency, the availability of an officially designated landing site is unlikely. In that case, safety systems for sUAS must be able to identify a suitable location for landing or crashing. This could include locations such as building tops, trees, lakes, parks, or roads. This task could be performed offline before the mission so that a database of landing site contingencies exists along the flight plan.

3) *Path planning.* Once a new emergency landing site is selected, an efficient path to the desired location must be designed. Examples include optimizing trajectories for power consumption, time to land, or observability constraints. If a sense-and-avoid system is present, a reactive path planning scheme may be used to safely guide the vehicle through air traffic. Additionally, a 3D obstacle map could be used to remain within UTM-cleared airspace or to plan paths around static obstacles such as buildings or trees.

4) *Fault tolerant control.* Depending on the vehicle and the nature of the emergency event, specialized autopilot routines may need to be used to maintain maneuverability of damaged vehicles. Additionally, it is the goal of the controller to ensure that paths created by the planner are followed.

5) *Tactical maneuvering.* As the sUAS makes its final approach to the impromptu landing site, there are likely to be dynamic ground obstacles such as cars, animals, and people that were not accounted for in the path planning stage. The goal of tactical maneuvering is to react to these non-cooperative ground obstacles using on-board sensing so that the vehicle can land without causing harm to the ground obstacles or itself. Depending on the population density of the chosen landing site, this step of emergency landing can be the most challenging in regard to safety and is known as the “last 50 feet” concern by UTM [5].

Safe2Ditch is a concept for an on-board crash management system that continuously monitors vehicle health and mission objectives. This paper focuses on real-time landing site selection and builds upon [10] to introduce visual situation awareness into the site selection process. While other methods discussed in Sec. II have designed landing site selection schemes, this paper incorporates visual tracking so that the current state of potential ditch sites can be assessed while still at higher altitudes. In particular, this paper contributes an implementation that combines a database of landing sites and real-time sensor information. Simulation and hardware experiments demonstrate the ability of *Safe2Ditch* to quickly select the best landing site using visual tracking.

The paper is organized as follows. Section II reviews current solutions relevant to forced landing of sUAS. Section III discusses each of the emergency landing steps in the context of the *Safe2Ditch* architecture. Section IV discusses the vision processing and target tracking algorithms that were designed and implemented to inform *Safe2Ditch* about obstacle motion on the ground. In Sec. V, the *Safe2Ditch* system is tested in simulation and in hardware. A discussion of the results is also given. We conclude with final thoughts and next steps in Sec. VI.

II. Related Work

A. Emergency Landing

One of the earliest fully integrated emergency landing systems to be flown on-board an unmanned aerial vehicle was by Scherer et al. [11]. In real-time, the system demonstrated the ability to detect, select, and navigate a full-scale unmanned helicopter to an unprepared landing site. Using lidar for perception and on-board computing resources, the system is able to build an online 3D point cloud of the terrain surrounding the full-scale unmanned helicopter. This terrain model is then used to evaluate how well a 3D model of the landing gear fits at different locations. Once the contact of the landing gear to the terrain model is nearly level, the landing site is determined and the helicopter autonomously navigates and lands there.

Mejias and Fitzgerald [12] developed a visual landing site detection system using a method based on image segmentation. Applied to video data from a Cessna 172 flying at approximately 1500 ft (450 m) above ground level, the detection system is able to find large potential landing sites. Warren et al. [13] use a similar system that incorporates digital elevation models (DEM) and inertial measurement unit (IMU) data to perform dense 3D reconstruction using a structure-from-motion technique. Shen et al. [14] also developed a visual landing site detection system for manned fixed-wing flight to assist pilots in choosing an appropriate landing site, but was only tested on simulated data. Note that the aforementioned works do not explicitly handle cases of motion in the potential landing sites.

While these automated systems were successful for larger aircraft, in this paper we target sUAS with limited size, weight, and power constraints. Additionally, the operational envelop that we are targeting is below 400 ft (120 m) and with only 1 min to land once the emergency is triggered. Eendebak et al. [15] demonstrate a vision-based landing site selection on postprocessed video acquired from a handheld camera at 100 ft (30 m). Using a background estimation filter, foreground elements are exposed and a graph-cut-based segmentation method is used to create a 2D obstacle map that allows the safest landing site to be chosen. However, the use of background estimation and subtraction requires slow, smooth flight and the sUAS must hover in place while potential landing sites are being imaged. In the work of Mackay et al. [16], landing site identification for multirotors is proposed using a plane-fitting algorithm on RGB-D point-cloud data.

Existing commercial solutions are based heavily on parachute technology, such as ParaZero [17]. These passive systems are inexpensive and effective at ensuring a safe descent, but leave the sUAS vulnerable to wind and crippled in cooperative sense-and-avoid capability. Other sUAS simply rely on the human operator to steer the drone to a clear landing zone as it slowly descends [18].

Research in other facets of the sUAS emergency landing problem has also been progressing. In emergency landing of manned aviation, Meuleau et al. [19] present a method of assessing risk and planning minimum risk paths around obstacles such as weather, terrain, and urban development, which are determined via an offline database and ground control communication. Di Donato and Atkins [20] proposed enhancing the site-selection process with the addition of nontraditional information such as real-time mobile phone activity to assist in avoiding densely occupied areas. In contrast, our work assumes a database of preselected ditch sites and focuses on refining site selection based on real-time motion. Castagno et al. [21] proposed a risk-based planner for rooftop landing of sUAS using a database that is constructed offline. Instead of focusing on the real-time state of potential landing sites using local sensor input, Castagno et al. [21] use a real-time planner to assess both the landing and path risk of nearby rooftops outside the perceptual range. The work of Coombes et al. [22] provides a landing site reachability analysis of a fixed-wing sUAS that is gliding due to engine failure. Mueller and D'Andrea [23] demonstrate fault-tolerant control of a quadrotor with complete loss of various propellers.

B. Visual Target Tracking

Visual target tracking is an active area of computer vision research [24,25]. Three defining characteristics of visual trackers are 1) their ability to track from a stationary versus moving camera, 2) their ability to track single versus multiple objects, and 3) detection-free tracking, which must be manually initialized, versus tracking-by-detection, which can self-initialize tracks. In this paper, we use an on-board visual multiple object tracker that can self-initialize tracks moving independently from the vehicle's motion.

In a survey of vision-based techniques used on UAS, Kanellakis and Nikolakopoulos [26] discuss various works focusing on aerial surveillance and tracking of ground objects. Rodríguez-Canosa et al. [27] use Parallel Tracking and Mapping (PTAM) for motion estimation, which is then used to create an artificial optical flow field. The difference between Lucas-Kanade optical flow and the artificial flow field exposes dynamically moving objects; however, this technique requires initialization using a marker map and may struggle to detect objects moving with similar velocity to the multirotor. Li et al. [28] successfully track other sUAS for sense-and-avoid applications in postprocessed aerial footage; however, they

assume that targets are nondeformable, which limits the types of objects that can be tracked (e.g., nonbiological objects). Jiang and Cao [29] are able to track multiple objects in postprocessed aerial video using detections based on background modeling, but do not use Bayesian filtering for track management. Teutsch and Krüger [30] also track multiple objects in postprocessed aerial videos and demonstrate traffic surveillance and tracking from a constant altitude. Their approach is most similar to the tracking pipeline used in our work, but assumes constant-altitude stable flight over structured environments, which results in smooth video input. The visual multiple object tracker used in this paper can detect and track an arbitrary number of targets in real-time using a bank of Kalman filters.

III. Safe2Ditch Architecture

Safe2Ditch is an sUAS crash management system. Its goal is to provide emergency landing capability to either rotorcraft or fixed-wing autonomous vehicles. By communicating with the vehicle's sensors and autopilot, the set of core Safe2Ditch algorithms is able to react to emergency situations. To allow widespread access to the safety that this system provides, Safe2Ditch is designed as a flexible framework capable of interfacing with a wide variety of commercial off-the-shelf (COTS) components.

Because of the modular construction of Safe2Ditch, application to fixed-wing vehicles is done by replacing the path planning module. The current implementation specifically addresses multirotors.

To manage each of the emergency landing steps as defined in Sec. I, Safe2Ditch has a number of subsystem components as shown in Fig. 1. This paper addresses and demonstrates the six components shaded with light gray, while health monitoring and tactical maneuvering are left for future work. The function and interdependence of each subsystem is described below in the context of the notional emergency scenario depicted in Fig. 2. As the system manages forced landings and crashes, the vehicle's priorities are to

- 1) avoid people,
- 2) avoid damaging property, and
- 3) avoid damaging itself.

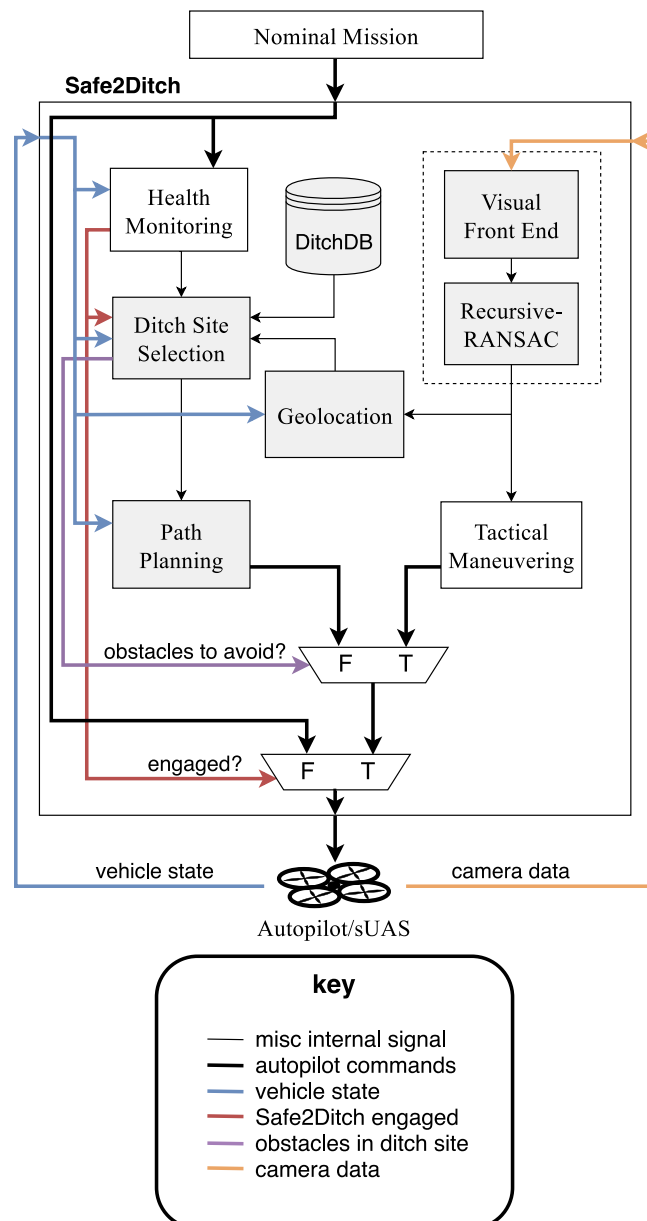


Fig. 1 Notional flight scenario in which Safe2Ditch would be used.

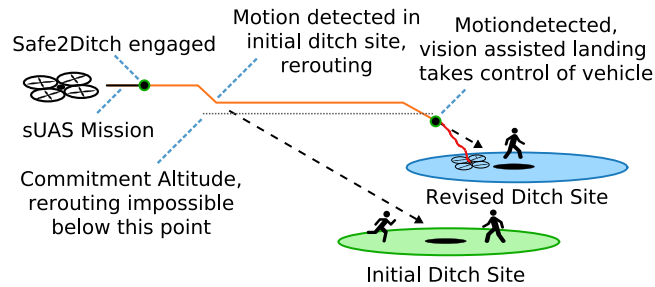


Fig. 2 Safe2Ditch system architecture. This paper addresses and demonstrates the six light gray components. When Safe2Ditch is not engaged, the nominal mission waypoints are passed unchanged.

These priorities are encoded as risks and are allowed to take on different weights in other works [19,21]. We assume that once Safe2Ditch is engaged, the vehicle is only able to descend and move laterally (i.e., it cannot ascend). This assumption simulates the urgency of quickly landing in the event that only 60 s of flight capability is available due to depleted battery.

A. Health Monitoring

The Safe2Ditch system is continuously running for the duration of the sUAS mission. During normal operation, Safe2Ditch is simply monitoring vehicle health and communication channels that may relay an emergency land directive. For example, the battery or fuel required for the mission completion may be insufficient due to unexpected headwinds, a fuel leak, or poor battery health. Alternatively, the vehicle state determined by the autopilot may deviate significantly from the internal dynamic model of the health monitoring systems. If either of these or other performance anomalies occur, the health monitoring system engages Safe2Ditch to manage the emergency.

B. Ditch Site Selection

Once Safe2Ditch is engaged, a ditch site is selected. This step is managed by the Ditch Site Selector subsystem, which is connected to the ditch site database (DitchDB) and the visual tracking subsystem as shown in Fig. 1. A preloaded database provides optional ditch site locations with key descriptors. The current position of the sUAS and the estimated time to land from the health monitoring system inform which ditch sites are within range. If no sites are in range, the vehicle immediately begins its descent and uses all of its remaining energy for tactical maneuvering. If prevetted sites are within range, the Ditch Site Selector selects the optimal site based on predicted vacancy, size of the site, terrain factors, and other weighted factors. Specific values for the weight factors are a subject of ongoing research. The current Safe2Ditch prototype places the highest value on predicted vacancy because the rotorcraft has no runway length requirement and because the test range is outfitted with many ditch site options within the vehicles range. Simulation runs using larger real-world scenarios may indicate more emphasis on range to conserve energy for tactical maneuvering. Once the ditch site is selected, its location is given to the path planner.

The use of a preloaded database has precedence in commercial aviation where airport and airspace information is preloaded for use by the flight management system (FMS) for the planned region for a given flight. This allows the system to have a continually updated set of information and provides opportunity for governance. For example, future databases could provide optional ditch sites as well as avoidance areas.

C. Path Planning

Using the position of the selected ditch site, the path planner generates an efficient path for landing. An efficient path minimizes the landing time and maximizes the amount of time that the ditch site can be imaged by the on-board camera. By increasing the amount of time that the camera can image the ditch site, the visual tracking subsystem can better inform the Ditch Site Selector of any motion in potential ditch sites. Therefore, for a multirotor, an efficient path can be achieved by spending the majority of the landing time descending at an angle equal to the camera mounting angle. As discussed in Sec. V, the vehicle used for flight tests has a camera that is mounted at a 45 deg angle toward the ground.

To maximize the time descending at the optimum imaging angle, an intermediate waypoint is inserted between the vehicle's current position and the ditch site. This waypoint becomes the top-of-descent (TOD) point, as shown in Fig. 3. For the 45 deg camera mount used in this paper, this is conveniently located at a distance equal to the vehicle's current altitude.

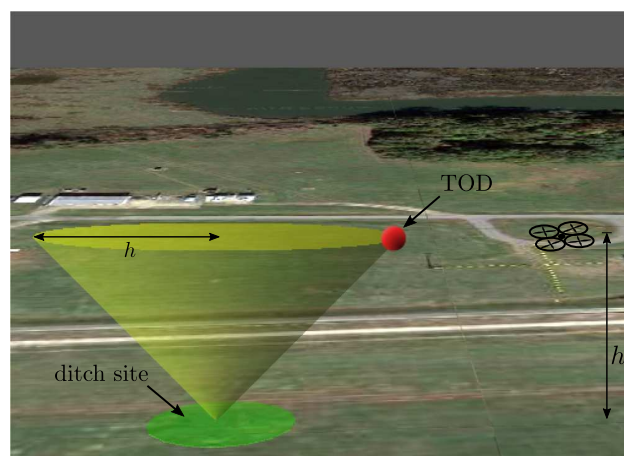


Fig. 3 Annotated ditch site visualization. The Safe2Ditch path planner creates a TOD point so that imaging of the ditch site for obstacles is maximized.

D. Tactical Maneuvering

As the vehicle descends toward the selected ditch site, newly detected obstacles in the landing area may cause Safe2Ditch to consider another ditch site as better suited for landing. However, at some altitude, the sUAS will not have enough battery power to reroute to other ditch sites and must commit to land. The commitment altitude is calculated by comparing the remaining battery power of the sUAS with the distance needed to reach other ditch sites. If the sUAS does not have enough power to reach another ditch site, the better-suited ditch site is disregarded and the sUAS continues its path to the original location. In these cases, Safe2Ditch is forced to choose a suboptimal ditch site that contains moving ground objects. To maintain safety and prevent harm to the moving ground obstacles, Safe2Ditch alters its 45 deg straight-line path and begins to use a control scheme that tactically maneuvers around moving ground obstacles in the ditch site. This vision-based control scheme is not discussed in this paper.

E. Visual Tracking Subsystem

The visual tracking subsystem is made up of the *visual front end* and the *Recursive-RANSAC* blocks shown in Fig. 1. Using only the vehicle's on-board camera, the visual tracking subsystem can estimate the position and higher-order dynamics of moving ground obstacles, as well as provide awareness of nonplanar ground objects such as fences or trees. This image-based tracking information is then geolocated and used to inform the Ditch Site Selector of motion in the ditch site. Additionally, the image-based tracking information could be used for vision-based control techniques to perform tactical maneuvering. The visual tracking and geolocation components are discussed in more detail in Sec. IV.

IV. Visual Multiple Target Tracking

To enable autonomy in dynamic environments, a visual tracking subsystem is integrated into the Safe2Ditch architecture. This visual target tracking component plays two roles: 1) to inform the Ditch Site Selector (see Sec. III.B) as it is scoring ditch sites, and 2) for tactical avoidance during the final landing maneuver. The work presented in this paper builds on our previous work of designing a vision-based multiple target tracking system for use on a descending platform [10].

The visual target tracking subsystem allows tracking of multiple moving targets from the on-board camera of the sUAS. Combining a visual measurement source with an online estimation back end known as Recursive-RANSAC, a robust tracking filter is created that requires no operator intervention to initialize or manage target tracking. Target tracks produced by Recursive-RANSAC are then geolocated and projected onto a flat Earth model to estimate the 3D position of the visually tracked targets.

A. Camera Geometry

To aid in downstream tasks, the measurement processing of the visual tracker is performed using normalized image coordinates as opposed to pixels. Consider the geometry of a pinhole camera model, as shown in Fig. 4. Suppose that the point P exists in 3D space and can be expressed in the camera frame as $P^c = [x^c \ y^c \ z^c]^T$. The perspective projection equations for imaging the point P are given by

$$u = f_x x^{\text{im}} + c_x = f_x \frac{x^c}{z^c} + c_x \quad (1)$$

$$v = f_y y^{\text{im}} + c_y = f_y \frac{y^c}{z^c} + c_y \quad (2)$$

where u and v are, respectively, the x and y pixels; f_x and f_y are the focal lengths in the x and y directions in units of pixels; c_x and c_y are the pixel offsets to the camera's principal point; and x^{im} and y^{im} are the coordinates of P expressed on the normalized image plane. Note that the focal length and the principal point offsets may be found via camera calibration. In homogeneous coordinates, the normalized image coordinates can be written as the 3-vector

$$p^{\text{im}} = [x^{\text{im}} \ y^{\text{im}} \ 1]^T \quad (3)$$

Therefore, each u , v pixel received from the camera sensor can be transformed into the normalized image plane using Eqs. (1) and (2).

The use of normalized image coordinates enables parameter tuning to be generalized to different visual systems and different scenarios, using the camera calibration of each camera to calculate the normalized image coordinates. In particular, this allows altitude-dependent tuning of the visual tracking algorithm as presented in [10]. Additionally, unit bearing vectors can be constructed from the camera data by normalizing p^{im} . These bearing vectors are used in this work for target geolocation, as discussed in Sec. IV.D.

B. Visual Measurement Front End

The purpose of the visual measurement front end is to process incoming video data and generate measurements that can be fed to the Recursive-RANSAC Tracker. Because of the robustness of the Recursive-RANSAC algorithm, we value many low-quality measurements over few high-quality measurements. Motivated by this heuristic, the vision processing is performed with a calibrated camera in a three-step pipeline to 1) find feature correspondences between images, 2) compute a homography, and 3) detect true object motion. Each of these steps is briefly described below. More information can be found in [10].

1) *Feature management*: At each time step k , features from the last image X_{k-1} are propagated forward into the current image as X_{k-1}^+ using optical flow [31]. Feature correspondences (X_{k-1}, X_{k-1}^+) are sent as input to the next step in the pipeline for further processing. A new set of features X_k are then found using the Shi-Tomasi corner detection method for the current image $\mathcal{I}_k$. These features will be propagated in the next iteration. This step is known as Kanade-Lucas-Tomasi (KLT) tracking [32] and is depicted in Fig. 5a.

2) *Homography generation*: Using the feature correspondences (X_{k-1}, X_{k-1}^+) from the KLT tracker, a perspective transformation H known as a homography is estimated. This step is crucial for tracking on-board an sUAS because it allows the set of features X_{k-1} and X_k to be represented in the same coordinate frame through image registration. The quality of a homography estimation between camera views can be visualized via difference imaging, as shown in Fig. 5b. Note that the visual tracking subsystem only makes use of KLT features and that the difference image is only computed when assessing the homography estimation quality.

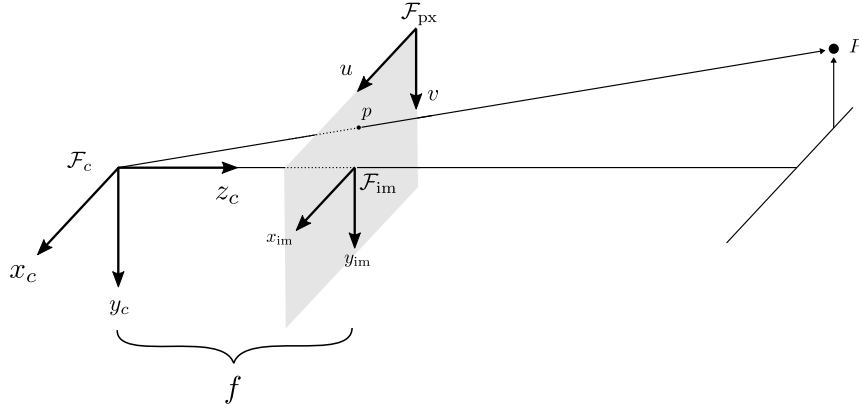


Fig. 4 Example output from each step of the visual multiple target tracker. Images are taken from video processed on-board a descending multirotor.

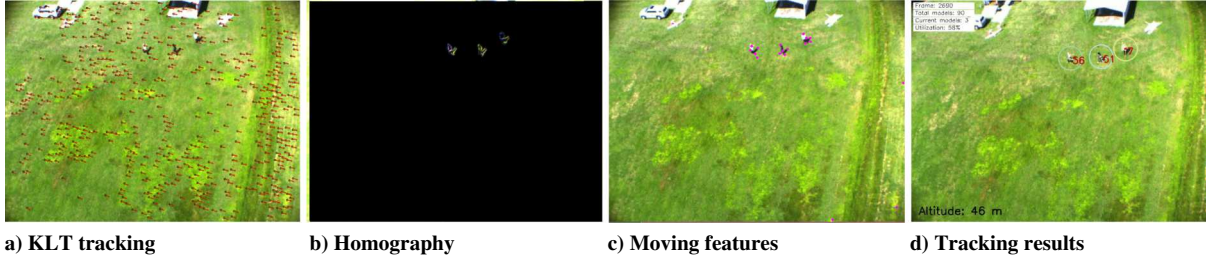


Fig. 5 Geometry of a pinhole camera with both the pixel plane $\mathcal{F}_{px}$ and the normalized image plane $\mathcal{F}_{im}$ shown.

3) *Moving object detection*: Equipped with a homography and a set of feature correspondences, the velocity of each of the feature points can be calculated as

$$V = X_{k-1}^+ - HX_{k-1} \quad (4)$$

If the homography estimate explains the motion of static features in the image plane well, then the velocity of static features will be nearly zero, leaving behind the motion of independent objects only, as shown in Fig. 5c. Measurements ($z_j = [x, y, v_x, v_y]^T \in Z_j$) of independently moving objects are defined as feature points that have a velocity magnitude within predefined thresholds, given by

$$Z_{k-1} = \{(x_i, v_i) \in X_{k-1}^+ \times V: \tau_{v_{\min}} \leq \|v_i\|_2 \leq \tau_{v_{\max}}\} \quad (5)$$

This scan of measurements is then used by Recursive-RANSAC to estimate the position, velocity, acceleration, and jerk of targets.

C. Recursive-RANSAC Tracker

Recursive-RANSAC is an online estimation algorithm capable of tracking an arbitrary number of objects in clutter [33,34]. Measurements are received in the surveillance region $\mathcal{R}$ of the system, where $\mathcal{R} \subset \mathbb{R}^2$ and represents the image plane. Recursive-RANSAC uses the random sample consensus (RANSAC) algorithm to quickly initialize hypothesis models that best fit the current measurements in a maximum-likelihood sense. Once a model is initialized, a bank of Kalman filters is used to propagate each model forward based on nearly constant jerk kinematics and a data association step as new measurements are received. The tracks found by Recursive-RANSAC can be seen in Fig. 5d.

The benefits of using Recursive-RANSAC are found in its ease of implementation, lightweight computational requirements, robustness in rejecting outliers, and its ability to initialize and manage models without an operator [35].

D. Target Geolocation

Once Recursive-RANSAC produces target tracks, the position p_{uas} of the sUAS is used to estimate their inertial position. Given the target track p^{im} expressed in the normalized image plane as in Eq. (3), a normalized line-of-sight (LOS) vector expressed in the camera frame can be constructed as $\hat{\ell}^c = p^{im}/|p^{im}|$. Using the unit LOS vector $\hat{\ell}^c$ and the height above ground level h of the sUAS, the objective is to estimate the inertial position of the target, assuming a level ground plane. The inertial position of the target can be expressed as

$$p_{tgt}^i = p_{uas}^i + L\hat{\ell}^i \quad (6)$$

where L is the range from the sUAS to the ground target. Because cameras are bearing-only sensors, the range must be estimated. The LOS vector in the inertial frame is given by

$$\hat{\ell}^i = R_b^i R_c^b \hat{\ell}^c \quad (7)$$

where $R_b^i \in \text{SO}(3)$ is the rotation from the body frame $\mathcal{F}_b$ to the inertial frame $\mathcal{F}_i$, and $R_c^b \in \text{SO}(3)$ is the rotation from the camera frame $\mathcal{F}_c$ to the body frame. The range L can be calculated using the cosine of γ , the angle between the inertial k^i -axis and the unit LOS vector $\hat{\ell}^i$. Note that the

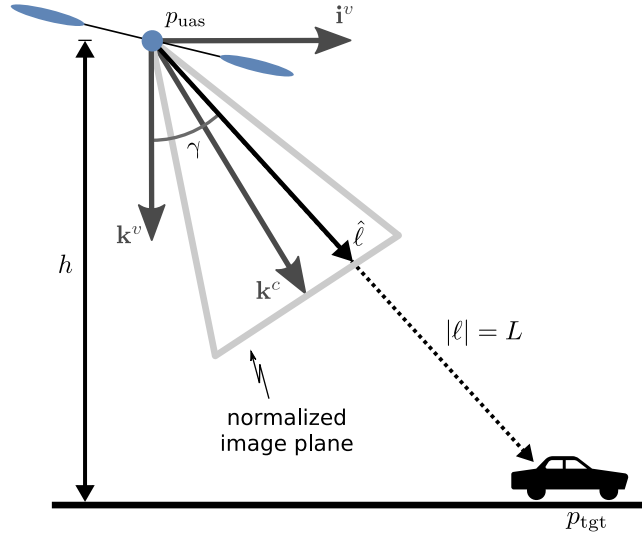


Fig. 6 Target geolocation side view. The objective is to find an estimate of p_{tgt} expressed in the inertial frame.

vehicle frame shown in Fig. 6 is aligned with the inertial frame, but translated to the vehicle's center of gravity (i.e., $k^v = k^i$). Using the following equality

$$\hat{\ell}^i \cdot k^i = \cos \gamma = \frac{h}{L} \quad (8)$$

the range can be written

$$L = \frac{h}{\hat{\ell}^i \cdot k^i} \quad (9)$$

Thus, the inertial position of the target at time t is given by combining Eqs. (6), (7), and (9). The inertial position of moving ground obstacles can then be used to inform the Ditch Site Selector during an emergency landing, as discussed in Sec. III.B.

V. Experimentation

To demonstrate the obstacle-aware Ditch Site Selection component of the Safe2Ditch system, simulation and hardware experiments were performed. The multirotor used for hardware tests is shown in Fig. 7 with details given in Table 1.

A. Implementation

The Safe2Ditch system is implemented in C++ and Python for real-time execution on-board an NVIDIA Jetson TX2 embedded computer. Using the Robot Operating System (ROS) [36] for message passing, each component can be designed, implemented, and tested independently of each other. This allows for rapid experimentation and hardware validation of the Safe2Ditch concept.

In addition to other standard software engineering concepts used in real-time applications, the visual processing stage is implemented using CUDA routines to exploit the parallelism inherent in image processing. The NVIDIA Jetson TX2 contains 256 low-power CUDA cores that enable the use of this type of processing.

The Safe2Ditch system is able to communicate with the Pixhawk autopilot that is used to control the multirotor and to estimate its current pose and velocity. This allows the geolocation algorithm (see Sec. IV.D) and the Ditch Site Selector (see Sec. III.B) to access the needed inertial information about the vehicle.

B. Software-in-the-Loop Simulation

To perform successful hardware flight tests, it is important to simulate the integrated system as best as possible. Using the ROS ecosystem with the Gazebo physics simulator, it is possible to create a software-in-the-loop (SIL) simulation by executing the Pixhawk/APM software on a desktop computer. This allows the same Safe2Ditch software that will run on the flight computer to be executed in simulation. Using satellite imagery, the SIL is additionally able to produce synthetic images of flight test locations with simulated people walking in various ditch sites. Within the SIL environment, end-to-end emergency landing scenarios are tested that exercise the Safe2Ditch components and increase confidence in effective flight demonstrations.

The geographical location used in SIL simulations is the sUAS Test Site at the NASA Langley Research Center. The nominal mission starts with the sUAS taking off at the home position and consists of flying counterclockwise through five waypoints at a height above ground level (AGL) of 60 m, as shown in Fig. 8. Three prioritized ditch sites are loaded into the DitchDB and are also shown in Fig. 8, with ditch site 1 being the first choice. At some random time t_{engage} , Safe2Ditch is engaged and attempts to select ditch site 1, which has the highest priority in the DitchDB. Safe2Ditch must manage landing safely and quickly in the presence of randomly moving targets in the area bounded by dashed lines. If a target is detected in ditch site 1, Safe2Ditch reroutes to the next best ditch site (shown in yellow).

Using this test environment, a series of Monte Carlo simulations were instantiated, parameterized by the number of moving targets, $N_t \in \{1, 2, \dots, 10\}$. The i th target is modeled as a simple unicycle with kinematics



Fig. 7 The multirotor used in hardware experiments, with a Pixhawk autopilot. The camera has a resolution of 800×600 at 30 fps and is mounted at a 45 deg angle.

$$\dot{x}_i = v_i \cos(\theta_i) \quad \dot{y}_i = v_i \sin(\theta_i) \quad \dot{\theta}_i = \omega_i \quad (10)$$

where $v_i(t)$ is linear velocity and $\omega_i(t)$ is angular velocity. A waypoint-based path planner is used to command each target to a sequence of desired positions (x_i^d, y_i^d) using $\omega_i(t)$ as the control and letting $v_i(t) \equiv v_i(0)$.

For each $N_i \in \{1, 2, \dots, 10\}$, $M = 200$ trials were used to test the sensitivity of Safe2Ditch with respect to the number of moving obstacles N_i near the ditch site, the initial position $(x_i(0), y_i(0))$ and initial velocity $v_i(0)$ of a target, and the time that the Safe2Ditch system receives an emergency event, t_{engage} . The initial position of the i th target is sampled from a uniform distribution bounded by the area around the first ditch site as shown in Fig. 8. Subsequent waypoints are sampled from this same distribution, $(x_i^d, y_i^d) \sim \mathcal{U}(-120, -60) \times \mathcal{U}(-30, 30)$. The velocity with which the target moves is chosen as $v_i(0) \sim \mathcal{U}(0.5, 2.5)$ and $t_{\text{engage}} \sim \mathcal{U}(30, 70)$.

C. Monte Carlo Simulation Results

The following three metrics are used to evaluate simulation results:

1) *Time to action*, t_{action} . This metric captures how many seconds it took from when an obstacle entered the currently selected ditch site and is in the camera field of view to when this obstacle caused a reroute to the next best ditch site. Said another way, t_{action} describes the responsiveness of the system. Responsiveness is critical for emergency landing systems because each passing second can cause greater loss in controllability, energy, or altitude.

2) *Rate of failure*. A failure occurs if the multirotor lands in a ditch site with one or more targets present and in the camera field of view. For the purposes of simulation, we consider landing to occur once the multirotor descends to 5 m. A target is considered present in a ditch site if any part of the target has continuously intersected the ditch site area for more than 3 s. Additionally, we assume that agents would not knowingly walk under a landing multirotor and in our results we ignore any simulated target that is in the ditch site but not in the field of view of the camera. In the future, a camera with a wider field of view or complementary sensor modalities could be used to increase the detection region of Safe2Ditch.

3) *Rate of false reroutes*. As the sUAS descends toward the ditch site, it is possible that the visual tracking system outputs false positives that cause the Ditch Site Selector to unnecessarily reroute to another ditch site. Additionally, estimation error in the geolocation algorithm could cause Safe2Ditch to misclassify tracks as being inside the ditch site when the true target position is not.

These metrics are shown in Figs. 9–11 for 2000 Monte Carlo trials. From Fig. 9, we note that the performance of Safe2Ditch is robust to a varying number of obstacles moving in and near the ditch site. This behavior is desirable because as potential landing sites become more cluttered, a fast decision time will allow time for multiple reroutes within the time constraints. In fact, we see from Fig. 10 that Safe2Ditch performs with less than 1% failure rate as the number of moving obstacles increases, which can be explained by the fact that the opportunity of tracking at least one moving obstacle in the ditch site is increased. As expected, the higher the number of moving objects in and near a ditch site, the more likely a reroute will occur (see Fig. 11). Note that in all the simulation trials, no false reroutes occurred. In the simulation environment, there is no parallax for the visual tracker to falsely detect as a moving object, and the terrain is actually flat. Because of this, the tracking quality is high and the geolocation results using Eq. (9) are accurate.

While more obstacles near potential ditch sites make it clear that a ditch site is not ideal, it could become difficult for the multirotor to decide on ditch site to land in. Future research will investigate selecting an area within a ditch site to land in as long as that subarea is clear. In this way, obstacles on the far end of the ditch site will not rule out the entire area for landing.

Table 1 Hardware details

Component	Description
Platform	3DR Y6
Autopilot	3DR Pixhawk with APM v3.5.4
Sensors	GPS, IMU, barometer, magnetometer
RGB Camera	800×600 @ 30 fps, HFOV $\approx 34^\circ$
Processor	NVIDIA Jetson TX2

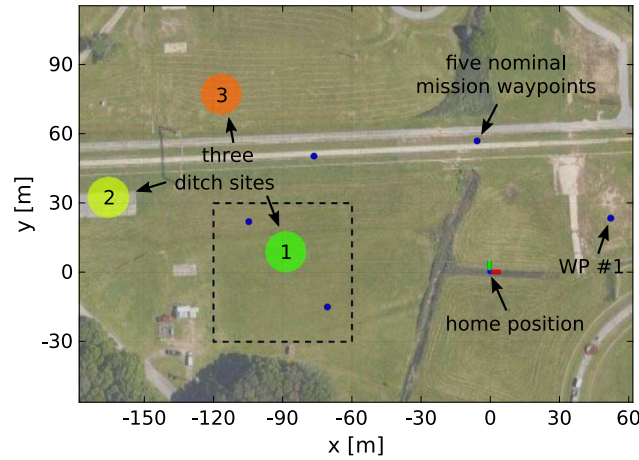


Fig. 8 Simulated environment at the NASA Langley Research Center, expressed in an ENU coordinate frame.

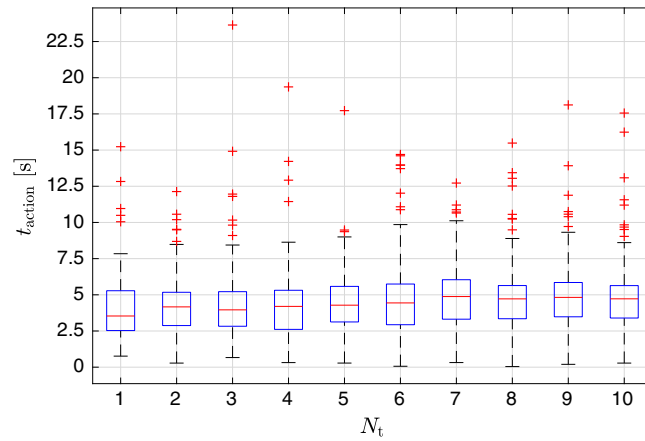


Fig. 9 Distribution of t_{action} . The inclusion of visual data allows Safe2Ditch to quickly respond to non-cooperative obstacles in the ditch site seen by the camera.

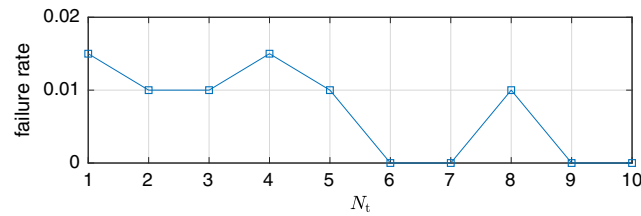


Fig. 10 The rate of failed emergency landings given N_t targets.

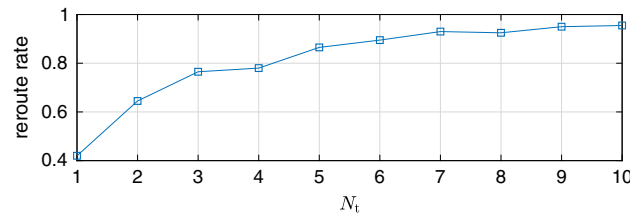


Fig. 11 The rate at which Safe2Ditch rerouted to another ditch site because of visually perceived motion in the original site. No false reroutes occurred in any trial.

D. Hardware Flight Test

Sixteen flight demonstrations were performed to exercise and validate the vision-aided Ditch Site Selection component of the Safe2Ditch emergency landing system. Three of the flights were performed near Provo, UT, with a mission height of 100 ft (30 m) AGL and ditch site radius of 13 ft (4 m). All other flights were performed at the NASA Langley sUAS Test Site, with 6 tests operating at 200 ft (60 m) AGL and 7 tests operating at 400 ft (120 m) AGL. The ditch site radius of the tests performed at NASA Langley was 30 ft (9 m). Of the 16 flights, 8 were performed with motion induced in the ditch site and 1 was performed with motion strictly outside of the ditch site. Motion was created by either an operator-driven RC car or a person wearing personal protective equipment. The other seven flights have no motion induced in or around the ditch site to test for false positives.

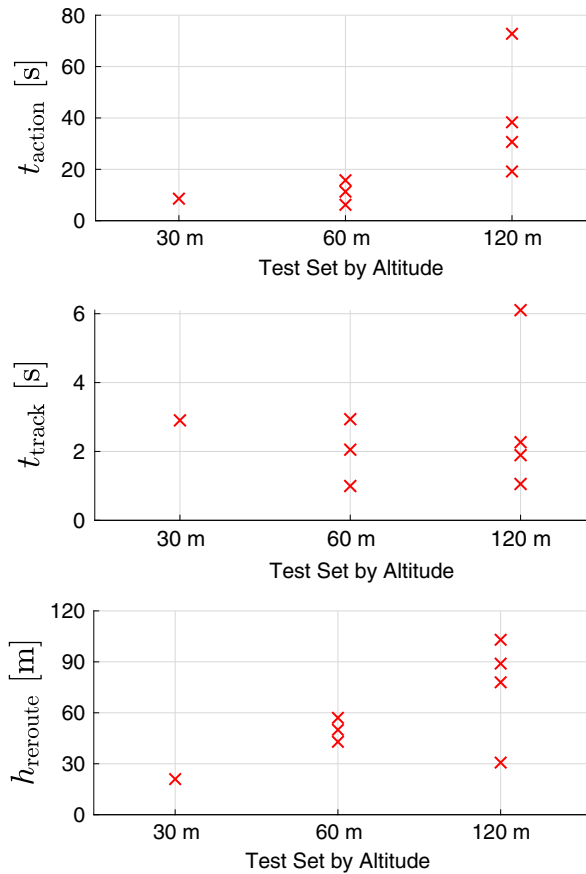


Fig. 12 Results from 16 hardware flight tests. Since only eight tests had motion induced in the ditch site, there are only eight data points.

Each flight scenario starts with a safety pilot controlling the takeoff phase and then transferring control to the autopilot to fly a predetermined mission with a number of waypoints. Each flight test has slight variations in its predetermined mission, with altitudes differing as noted above. At some time t_{engage} during the execution of each mission, an emergency event is generated and the Safe2Ditch system disrupts the mission operation and begins to safely land the vehicle. Throughout the duration of the test flights where motion is induced, targets are commanded to move with arbitrary trajectories and speeds inside of the ditch site.

E. Hardware Results

In addition to the metrics used for simulation results (see Sec. V.C), we analyze the hardware data using the following metrics:

1) *Time to track*, t_{track} . This measurement captures how quickly the visual system detects and tracks moving obstacles. It is measured from the time that the obstacle was both in the field of view and in the currently selected ditch site. The difference between t_{action} and t_{track} measures how long it took the Ditch Site Selector to consider the track as actionable and reroute to another ditch site.

2) *Altitude of reroute*, h_{reroute} . For hardware tests where a reroute is expected, h_{reroute} gives the AGL that the reroute occurred.

We present the data in Fig. 12 from the 16 flight tests with motion in the ditch site organized into 3 sets, broken up by the nominal mission altitude: 1) 3 flights at 30 m AGL, 2) 6 flights at 60 m AGL, and 3) 7 flights at 120 m AGL.

As shown by t_{track} in Fig. 12, the visual multiple target tracker can detect and track targets at 120 m and below within approximately 3 s. However, we can see from t_{action} that the Safe2Ditch system took more time to react to moving ground obstacles for the 120 m set. The greatest contributor to this latency in response time is the track geolocation algorithm (see Sec. IV.D), which often produced estimates with an error between 5 and 10 m. Because of this error, ground obstacles near the edge of the ditch site would often be misclassified as being inside or outside the desired landing region. This misclassification caused a single false positive during a flight from the 30 m test set (not shown in Fig. 12), where the moving object was just outside of the ditch site. The seven flights with no motion successfully landed in the first selected ditch site, with no failures.

We show the trajectories of two of the 120 m mission altitude tests flown at NASA Langley in Figs. 13 and 14. Both tests start on the ground at the origin of a local ENU inertial coordinate frame. The trajectory of the multirotor is shown in black, as it begins to travel counterclockwise about the nominal mission waypoints. When Safe2Ditch is engaged, the color of the plotted trajectory matches one of the two ditch sites available to be selected. In Fig. 13, there is no motion in the primary ditch site and the vehicle successfully lands. In Fig. 14, an agent moves around the primary ditch site, which is detected and plotted as red. Once the track of the moving agent becomes actionable, Safe2Ditch reroutes to the secondary ditch site, and the color of the plotted trajectory changes.[†]

The red plotted points in both Figs. 13 and 14 represent all tracks created by the visual tracker, throughout all time. Note that, in both figures, the visual tracker detects and tracks false positives at various locations of the mission. These false positives are a result of the parallax caused by nonplanar objects in the scene (such as trees, poles, and buildings) and not the Recursive-RANSAC algorithm. Parallax is an effect where positions of objects in a 3D scene differ when viewed from two different angles. Because monocular RGB cameras do not measure depth, the planar homography transformation that is discussed in Sec. IV.B cannot compensate for nonplanar motion. While this shows a weakness in the

[†]Video available at <https://youtu.be/r7GNcGiA5uw>.

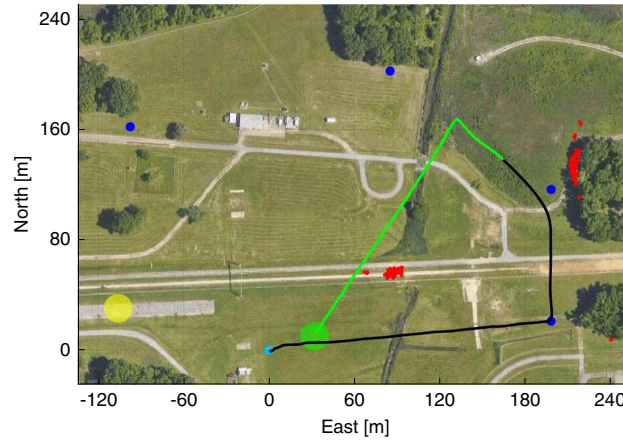


Fig. 13 Trajectory of flight #12. No obstacles are present in the primary (green) ditch site. Once Safe2Ditch is engaged, the vehicle successfully lands in the ditch site.

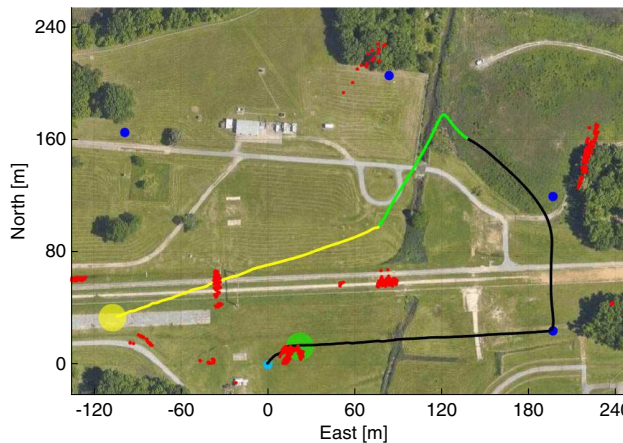


Fig. 14 Trajectory of flight #13. After Safe2Ditch is engaged, motion is detected in the primary ditch site, and the vehicle successfully lands in the secondary ditch site.

ability of the visual tracker to differentiate between moving objects and 3D structure, it does not hinder the objective of Safe2Ditch, which is to avoid obstacles as the vehicle lands. Obstacles in the ditch site that are detected based on motion or parallax will cause the vehicle to reroute to a safer area.

VI. Conclusions

This paper presented a crash management system for small unmanned aircraft systems (sUAS). This system was verified in 2000 Monte Carlo simulations with a constant nominal mission above ground level (AGL) and a varying number of targets. The simulation results were then supplemented with 16 hardware flight tests with either zero or one target and varying the nominal mission AGL.

Using a precompiled database of potential ditch sites along the route of the nominal mission gives the sUAS potential landing options in the event of an emergency. The contribution of this paper is the augmentation of a precompiled database with on-board vision capabilities, resulting in the Safe2Ditch system. The use of on-board visual tracking in conjunction with a precompiled database for initial landing site selection reduces the processing requirements of the vision system, but allows the sUAS to react to non-cooperative obstacles in populated environments. It is an important design criterion that the computational resources needed for managing emergency landing be low so that the normal mission operation is not hindered.

By incorporating a risk description into the problem, objectives can be better balanced. Future research directions include balancing the risk of landing in a ditch site with moving obstacles with the risk of additional flight time. If there is enough remaining flight time, the vehicle could even be commanded to perform a fly over to better survey potential ditch sites. Future research will also develop the tactical maneuvering component to allow landing in a ditch site where moving obstacles are present.

Future work will address the accuracy of the geolocation algorithm needed to determine if obstacles are located within the selected ditch site. To increase the accuracy of the flat Earth model geolocation estimate, the attitude uncertainty should be projected onto the ground plane as well. Then, instead of checking if a single point is within a ditch site, the intersection of the uncertainty and the ditch site would indicate how likely a target is in the ditch site. Alternatively, instead of tracking targets in the image plane, a visual-inertial odometry or visual SLAM approach could be taken to resolve the missing depth information from camera data and track targets directly in the inertial frame. This type of approach would additionally address the effects of parallax when flying low over nonplanar environments.

In addition to the position data from visual tracking, future work will leverage the estimated velocity of each kinematic model corresponding to tracked obstacles in R-RANSAC. Using velocity information can help inform the Ditch Site Selector about obstacles that are not currently in the ditch site, but are headed toward the ditch site. Similarly, this information can positively impact the score of a ditch site if an obstacle is expected to be quickly leaving the area.

Acknowledgments

This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF award Numbers IIP-1161036 and CNS-1650547, along with significant contributions from C-UAS industry members. The authors thank the NASA Langley Systems Analysis & Concepts Directorate (SACD) and the NASA UAS Traffic Management (UTM) project for funding this work and providing flight test support at the NASA Langley sUAS testing site.

References

- [1] Michał, M., Wiśniewski, A., and McMillan, J., "Clarity from Above," *PwC Drone Powered Solutions*, 2016, p. 38, <https://www.pwc.pl/pdf/clarity-from-above-pwc.pdf>.
- [2] Amazon.com Inc., "Revising the Airspace Model for the Safe Integration of Small Unmanned Aircraft Systems," NASA UTM: The Next Era of Aviation, No. 1, 2015, pp. 2–5.
- [3] Narayan, P., Wu, P., Campbell, D., and Walker, R., "An Intelligent Control Architecture for Unmanned Aerial Systems (UAS) in the National Airspace System (NAS)," *2nd Australasian Unmanned Air Vehicle Systems Conference*, Melbourne, Australia, 2007, pp. 20–31.
- [4] Kopardekar, P., and Bradford, S., "UAS Traffic Management (UTM): Research Transition Team (RTT) Plan," 2017, https://www.faa.gov/uas/research/utm/media/FAA_NASA_UAS_Traffic_Management_Research_Plan.pdf [accessed 2017 Aug. 29].
- [5] Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., and Robinson, J. E., III, "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations," *AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2016-3292, 2016. doi:10.2514/6.2016-3292
- [6] Endsley, M. R., "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 37, No. 1, 1995, pp. 32–64. doi:10.1518/001872095779049543
- [7] Endsley, M. R., "Automation and Situation Awareness," *Human Factors in Transportation: Automation and Human Performance: Theory and Applications*, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1996, pp. 163–181. doi:10.1177/154193120104500235
- [8] Adams, J. A., "Unmanned Vehicle Situation Awareness: A Path Forward," *Proceedings of the 2007 Human Systems Integration Symposium*, No. 615, 2007.
- [9] McAree, O., and Chen, W. H., "Artificial Situation Awareness for Increased Autonomy of Unmanned Aerial Systems in the Terminal Area," *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 70, Nos. 1–4, 2013, pp. 545–555. doi:10.1007/s10846-012-9738-x
- [10] Lusk, P. C., and Beard, R. W., "Visual Multiple Target Tracking from a Descending Aerial Platform," *American Control Conference (ACC)*, IEEE, New York, 2018, pp. 5088–5093. doi:10.23919/ACC.2018.8431915
- [11] Scherer, S., Chamberlain, L., and Singh, S., "Autonomous Landing at Unprepared Sites by a Full-Scale Helicopter," *Robotics and Autonomous Systems*, Vol. 60, No. 12, 2012, pp. 1545–1562. doi:10.1016/j.robot.2012.09.004
- [12] Mejias, L., and Fitzgerald, D., "A Multi-Layered Approach for Site Detection in UAS Emergency Landing Scenarios Using Geometry-Based Image Segmentation," *International Conference on Unmanned Aircraft Systems*, IEEE, New York, 2013, pp. 366–372.
- [13] Warren, M., Mejias, L., Yang, X., Arain, B., Gonzalez, F., and Upcroft, B., "Enabling Aircraft Emergency Landings Using Active Visual Site Detection," *Field and Service Robotics. Springer Tracts in Advanced Robotics*, edited by L. Mejias, P. Corke, and J. Roberts, Vol. 105, Springer, Cham, 2015, pp. 167–181. doi:10.1007/978-3-319-07488-7
- [14] Shen, Y. F., Rahman, Z. U., Krusienski, D., and Li, J., "A Vision-Based Automatic Safe Landing-Site Detection System," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 1, 2013, pp. 294–311. doi:10.1109/TAES.2013.6404104
- [15] Eendebak, P., Van Eekeren, A., and Den Hollander, R., "Landing Spot Selection for UAV Emergency Landing," *Proceedings of SPIE: The International Society for Optical Engineering*, Vol. 8741, No. 5, 2013. doi:10.1117/12.2017910
- [16] Mackay, J., Ellingson, G., and McLain, T. W., "Landing Zone Determination for Autonomous Rotorcraft in Surveillance Applications," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2016-1137, 2016. doi:10.2514/6.2016-1137
- [17] ParaZero, "ParaZero—Drone Safety Systems," 2017, <https://parazero.com> [accessed 17 July 2018].
- [18] DJI, *Mavic Pro Disclaimer and Safety Guidelines v1.0*, May 2017, <https://dl.djicdn.com/downloads/mavic/20170809/Mavic+Pro+Disclaimer+and+Safety+Guidelines-EN.pdf>.
- [19] Meuleau, N., Plaunt, C., Smith, D., and Smith, T., "An Emergency Landing Planner for Damaged Aircraft," *AAAI Twenty-First IAAI Conference*, 2009, pp. 114–121.
- [20] Di Donato, P. F. A., and Atkins, E. M., "Evaluating Risk to People and Property for Aircraft Emergency Landing Planning," *Journal of Aerospace Information Systems*, Vol. 14, No. 5, 2016, pp. 4–8. doi:10.2514/1.1010513
- [21] Castagno, J., Ochoa, C., and Atkins, E., "Comprehensive Risk-Based Planning for Small Unmanned Aircraft System Rooftop Landing," *International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, New York, 2018, pp. 1031–1040. doi:10.1109/ICUAS.2018.8453483
- [22] Coombes, M., Chen, W. H., and Render, P., "Reachability Analysis of Landing Sites for Forced Landing of a UAS," *Journal of Intelligent & Robotic Systems*, Vol. 73, Nos. 1–4, 2014, pp. 635–653. doi:10.1007/s10846-013-9920-9
- [23] Mueller, M. W., and D'Andrea, R., "Stability and Control of a Quadcopter Despite the Complete Loss of One, Two, or Three Propellers," *Proceedings: IEEE International Conference on Robotics and Automation*, IEEE, New York, 2014, pp. 45–52. doi:10.1109/ICRA.2014.6906588
- [24] Pan, Z., Liu, S., and Fu, W., "A Review of Visual Moving Target Tracking," *Multimedia Tools and Applications*, Vol. 76, No. 16, 2017, pp. 16989–17018. doi:10.1007/s11042-016-3647-0
- [25] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M., "Visual Tracking: An Experimental Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 7, 2014, pp. 1442–1468. doi:10.1109/TPAMI.2013.230
- [26] Kanellakis, C., and Nikolakopoulos, G., "Survey on Computer Vision for UAVs: Current Developments and Trends," *Journal of Intelligent and Robotic Systems*, Vol. 87, No. 1, 2017, pp. 141–168. doi:10.1007/s10846-017-0483-z
- [27] Rodríguez-Canosa, G. R., Thomas, S., del Cerro, J., Barrientos, A., and MacDonald, B., "A Real-Time Method to Detect and Track Moving Objects (DATMO) from Unmanned Aerial Vehicles (UAVs) Using a Single Camera," *Remote Sensing*, Vol. 4, No. 4, 2012, pp. 1090–1111. doi:10.3390/rs4041090

- [28] Li, J., Ye, D. H., Chung, T., Kolsch, M., Wachs, J., and Bouman, C., "Multi-Target Detection and Tracking from a Single Camera in Unmanned Aerial Vehicles (UAVs)," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, New York, 2016, pp. 4992–4997.
doi:10.1109/IROS.2016.7759733
- [29] Jiang, X., and Cao, X., "Surveillance from Above: A Detection-and-Prediction Based Multiple Target Tracking Method on Aerial Videos," *Integrated Communications Navigation and Surveillance (ICNS)*, IEEE, New York, 2016, pp. 1–0.
doi:10.1109/ICNSURV.2016.7486348
- [30] Teutsch, M., and Krüger, W., "Detection, Segmentation, and Tracking of Moving Objects in UAV Videos," *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, IEEE, New York, 2012, pp. 313–318.
doi:10.1109/AVSS.2012.36
- [31] Lucas, B. D., and Kanade, T., "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proceedings of the 7th International Joint Conference on Artificial Intelligence—Volume 2*, Morgan Kaufmann, San Francisco, CA, 1981, pp. 674–679.
- [32] Shi, J., and Tomasi, C., "Good Features to Track," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, New York, 1994, pp. 593–600.
- [33] Niedfeldt, P. C., and Beard, R. W., "Multiple Target Tracking Using Recursive RANSAC," *American Control Conference*, IEEE, New York, 2014, pp. 3393–3398.
doi:10.1109/ACC.2014.6859273
- [34] Niedfeldt, P. C., Ingersoll, K., and Beard, R. W., "Comparison and Analysis of Recursive-RANSAC for Multiple Target Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 53, No. 1, 2017, pp. 461–476.
doi:10.1109/TAES.2017.2650818
- [35] Ingersoll, K., Niedfeldt, P. C., and Beard, R. W., "Multiple Target Tracking and Stationary Object Detection in Video with Recursive-RANSAC and Tracker-Sensor Feedback," *International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, New York, 2015, pp. 1320–1329.
doi:10.1109/ICUAS.2015.7152426
- [36] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A., "ROS: An Open-Source Robot Operating System," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, IEEE, New York, 2009.

E. Atkins
Associate Editor