AIAA SciTech Forum
9 - 13 January 2017, Grapevine, Texas
AIAA Information Systems-AIAA Infotech @ Aerospace

10.2514/6.2017-0222

Fail-Safe Navigation for Autonomous Urban Multicopter Flight

Cosme A. Ochoa*
University of Michigan, Ann Arbor, MI, 48109, USA

Ella M. Atkins†
University of Michigan, Ann Arbor, MI, 48109, USA

Current multicopter fail-safe protocols respond with pre-programmed behaviors that aim to minimize possible damage to the aircraft or avoid risk to the nearby population. Existing fail-safe protocols for multicopters involve either returning to a pre-designated home point or executing an automatic landing protocol. Such strategies are rigid and can even increase risk in cases with complex terrain or high overflown population densities. This paper examines three alternative fail-safe protocols for autonomous urban multicopter flight that trade the simplicity of existing protocols with responsiveness to terrain, multicopter state, and overflown population. We utilize nontraditional data and sensor fusion to make an informed landing site selection with a focus on building rooftops as safe urban landing sites. The first protocol assigns a "Clearance to Land" flag to all buildings to determine the nearest rooftop in the area that can serve as a feasible landing site. The second protocol first determines a possible landing site within the multicopter's reachable footprint using lot types provided by a property database. During traversal to that site the multicopter examines the rooftops of overflown buildings and generates an alternative landing plan if it discovers a viable rooftop not previously mapped. The third protocol utilizes the known building locations and their approximate geometries as specified by the database to generate a spanning tree coverage solution followed by the multicopter to search for a viable landing site. A feature-based map of Manhattan generated from the PLUTO and MapPLUTO tax lot databases is used for simulations of all three protocols.

Keywords: Fail-safe operation, Urban Flight, Map-Based Navigation, Flight Planning, Path Planning, Coverage Planning

I. Introduction

Small Unmanned Aircraft Systems (sUAS) such as multicopters are expected to fly over populated urban centers as well as isolated rural test ranges once they are sufficiently safe. Triple redundancy is likely not feasible, so sensor-data fusion and fail-safe contingency response are critical capabilities to develop, certify, and deploy across the emerging sUAS fleet. Given real-time and situational awareness decision-making constraints, the supervisory role of a remote operator/dispatcher is inadequate for UAS contingency response; instead the UAS must quickly take action to assess a hazardous situation and react, potentially in only a few seconds. Current air traffic control (ATC) procedures do not address UAS failure events, particularly given that sUAS are expected to operate with very little altitude recovery margin. In addition, UAS rely on datalink-based communication for remote pilot commands to even be received. In the event of a lost data signal, neither ATC nor the remote pilot are able to command the distressed aircraft, further motivating the use of onboard fail-safe protocols.[1]

Current fail-safe protocols in hobby-class multicopter systems respond with pre-programmed behaviors to prevent the crash of the sUAS in response to a recognized anomaly or system failure. Existing fail-safe

*PhD Student, Robotics Institute, Student Member
†Professor, Department of Aerospace Engineering, Associate Fellow.

protocols include returning to a designated home point when the radio control (RC) or datalink signal has been lost or when the multicopter battery dips below a fixed threshold. If the multicopter loses a navigation (e.g., GPS) signal or detects a critically low battery level, the multicopter will typically initiate an automatic landing protocol.[2] Electronic geofencing has also been proposed to prevent fly-away,[3,4] but to-date has relied on user entry or a sparse database of restricted sites, e.g., the White House lawn.

This paper proposes a set of map-based fail-safe protocols that enable a multicopter experiencing an anomaly requiring a near-term landing, e.g., low battery energy, to safely terminate its flight in an urban environment where an immediate fail-safe landing poses risk to people on the ground. This work builds upon the work of Ten Harmsel et. al. which used map-based elements to plan an emergency landing path for a UAS that experiences low energy levels when flying above Manhattan.[5] Ten Harmsel et. al. utilized public building, terrain, and population databases to create a cost map explored by an A* search algorithm to generate an emergency flight plan. Using database information of the area, one can augment the search space typically limited by the field of view and range of onboard sensors. This paper proposes to combine onboard sensor observations and database information with a coverage flight planner to allow an autonomous multicopter to make the most informed decision regarding where to land, with focus on [flat] building rooftops as safe urban landing sites.

Safe urban flight requires more than awareness of GPS coordinates, both because of the sUAS's need to recognize and account for local structures and because GPS data may not be available in urban canyon environments.[6] A key to safely navigating GPS-denied areas is to fuse data with localization techniques. For example, previous work by Leutenegger and Siegwart proposed an Extended Kalman Filter (EKF) state estimation framework using airspeed measurements as backup during periods of unavailable GPS data.[7] Mao et al. proposed a similar EKF framework using inter-UAS range measurements between multiple cooperative aircraft to estimate the positions of the aircraft that had lost their GPS connection.[8] Rady et al. presented a vision based approach to construct a feature based map of the area used for UAS localization to complete a mission or fly to an emergency safe point.[9] In this paper, map-based navigation is used to supplement traditional inertial navigation via GPS. While transit flight plans will typically rely on inertial state estimates, fail-safe flight plans are more concerned with the local environment where the transit and ultimate landing will occur.

This paper proposes three alternative fail-safe protocols for autonomous multicopter flight in Manhattan using sensor and nontraditional database fusion. Given access to the PLUTO and MapPLUTO land use and geographic databases of the Manhattan's tax lots,[10] the multicopter can take advantage of this information to build a feature-based map and generate alternative fight plans to land on a building rooftop. While rooftops are not the only emergency landing option, they are perhaps the most unlikely areas to be occupied thus present relatively low risk of harm to people relative to surface areas. The investigated fail-safe protocols prioritize a quick response to system failure such that the multicopter can land safely in time to avoid an uncontrolled descent into an occupied street or sidewalk.

The multicopter first must determine whether the city database provides sufficient information concerning whether each building is designated as a potential landing site (i.e., a "Clearance to Land" flag is set). If so, the multicopter can generate and follow a plan to the closest available rooftop – diversion to a known rooftop is the simplest protocol examined in this paper. The database may not contain such information, however, in which case the multicopter can fly to a surface landing site (e.g. the nearest open park) and examine rooftops enroute using an onboard camera. This hybrid strategy of inspecting landing sites enroute to a known site is our second fail-safe procotol. If no rooftop or surface safe landing site is identifed, the multicopter can execute a coverage path to explore the area and evaluate each overflown building's rooftop to find a safe landing site, representing the third fail-safe protocol examined in this paper.

The three proposed fail-safe algorithms combine the use of database information, map generation, flight planning, and area coverage exploration to navigate the multicopter to land safely on mapped and discovered building rooftops. The paper will extend the use of coverage algorithms to aid in searching for a safe landing site beyond sensor range when this information is not available in an accessible database. A series of simulation case studies evaluate the proposed protocols on low battery fail-safe scenarios represenative of low-altitude urban multicopter missions.

The remainder of the paper is organized as follows. Sec. II summarizes the fail-safe landing problem and the three alternative protocols proposed in this paper. Sec. III describes the PLUTO and MapPLUTO databases and summarizes processing required to prepare map data for the simulation environment. Sec. IV describes how the simulation environment was constructed from map data while Sec. V provides a more detailed description of the planning algorithms used for the three fail-safe protocols. Sec. VI presents resuls from simulation case studies in which each of the three protocols is applied to a multicopter experiencing a fail-safe trigger while flying over Midtown Manhattan. Finally, conclusions and future work are discussed in Sec. VII.

II. Problem Statement

This paper examines three alternative fail-safe protocols for a multicopter flying over an urban area, specifically Midtown Manhattan. Case studies use the PLUTO and MapPLUTO land use and geographic databases provided by the New York of City Planning to generate simulations and to enable the multicopter to make an informed landing site decision in time to avoid an unsafe descent onto pedestrians. A feature-based map of the city is built using relevant processed lot information including: lot location (latitude and longitude), lot shape, land use, and average building height. For simplicity, each lot is approximated as a single building with the same features as the tax lot it represents.

Prompted by the occurrence of a low battery event triggering a near-term fail-safe landing, the multicopter's flight planner utilizes the generated map to determine a viable landing site and generates a constant altitude flight plan for the multicopter to execute. This paper assumes the multicopter can localize and navigate safely in the urban environment while executing a constant altitude fail-safe-triggered flight plan.

Three alternative fail-safe protocols are examined in this paper. The decision regarding which protocol to execute depends on the the amount of information available to the multicopter at the time the fail-safe protocol is invoked. The following table summarizes which protocol should be executed given the information provided by the database.

Protocol	Clearance to Land	Land Use	Building Location	Building Shape	Building Height
1	✓	✓or -	✓	✓	✓
2	-	✓	✓	✓	✓
3	-	-	✓	✓	✓

Table 1: Denotes which alternative fail-safe protocol should be executed given the available (✓) and unavailable (-) information with respect to the database being used by the multicopter for planning.

One case study is provided for each fail-safe protocol to illustrate its use. In the first case study, the multicopter determines the closest rooftop flagged as available by a "Clearance to Land" category in the database. A rooftop with the "Clearance to Land" flag is assumed free of all obstacles and people; this flag is only set when the roof has a geometry (e.g. slope, shape, and area) to support the safe landing of a sUAS multicopter. Once the rooftop site has been identified A* search is used to generate a landing flight plan with more details provided below.

In the second case study, the multicopter first finds a relatively distant but [barely] reachable landing site using the Land Use categories provided by the database for each tax lot and the multicopter's battery life at the time of failure. Once this landing site has been determined, the multicopter again utilizes A* search to generate a flight plan to this landing site. As the multicopter travels along its trajectory it observes overflown building rooftops geometries using an onboard downward-facing camera. The overflown creates an"observation tube" with size a function of camera field of view. If the multicopter discovers a viable landing rooftop enroute, the multicopter will execute an immediate landing protocol to eliminate the risk posed by continuing the flight to the more distant site.

In the third case study, it is assumed that the multicopter does not have sufficient information to select a landing site beyond sensor field of view. Given the property database but a priori knowledge of landing sites, the multicopter generates a spanning tree-based coverage (STC) flight plan aimed at discovering a viable rooftop for fail-safe landing. If the multicopter's battery level reaches a critical state along its flight, the multicopter will attempt to land on the best rooftop observed so far within a "footprint" radius constrained by measured battery energy level.

III. Data Processing

A. Database Description

The New York City Department of City Planning (DCP), along with other city agencies, maintains the PLUTO and MapPLUTO public databases as part of their Open Data initiative.[10] These databases contain tax lot, land use, and geographic information classified by the five boroughs of New York City: Manhattan, Brooklyn, Queens, The Bronx, and Staten Island. For this paper only the information regarding Manhattan borough was taken into consideration. The PLUTO database contains extensive tax lot information in comma-separated value format (CSV). MapPLUTO merges the all PLUTO tax lot data into an ESRI shapefile for each borough with respect to a GCS_North_American_1983 geographic coordinate reference.

The relevant fields used to generate a feature-based map of the area include:

Source: Map˙PLUTO

- *ShapeX*: Denotes the x-coordinates of a polygon describing the shape of the tax lot in feet with respect to the shapefile's geographic coordinate reference.
- *ShapeY*: Denotes the y-coordinates of a polygon describing the shape of the tax lot in feet with respect to the shapefile's geographic coordinate reference.

Source: PLUTO

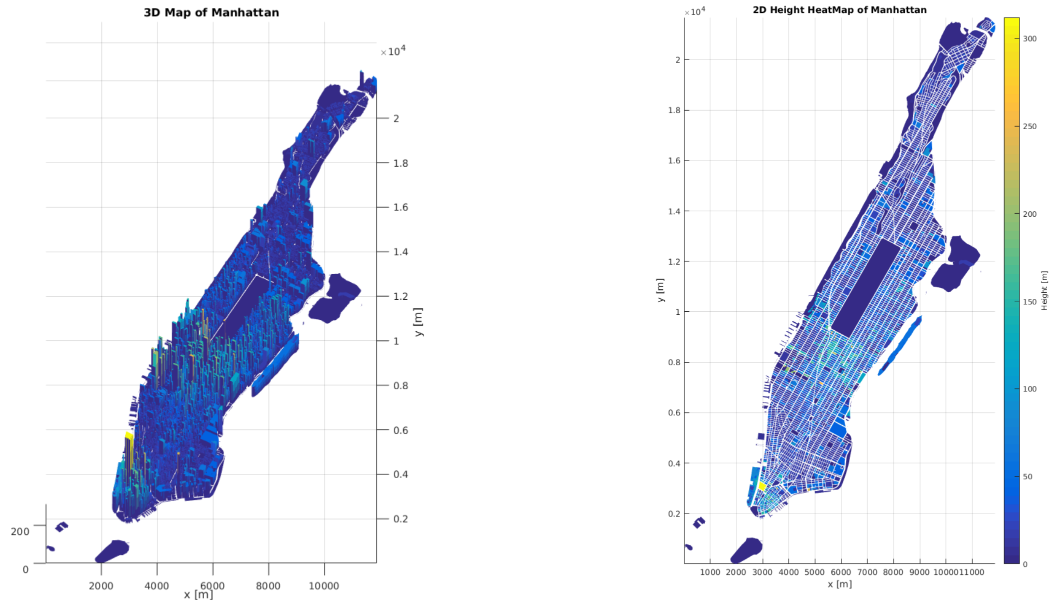
- *Block*: Specifies the tax block in which the tax lot resides. Each tax block is unique within a borough. The block number consists of a 5 digit numeric identifier (99999).
- *Lot*: Specifies the number of the tax lot. Each tax lot is unique within a tax block. The lot number consists of a 4 digit numeric identifier (9999).
- *BldgClass*: Specifies the major use of the structures, facilities, and area withing the tax lot. The building class consists of a mapping using a 2 character alphanumeric identifier as assigned by the New York Department of City Planning.
- *NumFloors*: The number of full and partial stories of the tallest building with the tax lot. The number of floors consists of numeric 5 digit identifier (999.99).

B. Database Processing

Each tax lot was a assigned a unique identifier which consisted of its block number followed by lot number as given by the city database. To take into account the buildings in each tax lot, the best approximation consisted of simulating one building for each tax lot with the building's geometry consistent with that of the tax lot itself, as defined by *ShapeX* and *ShapeY*. In addition, the *NumFloors* category for each tax lot was used to estimate the height of each building by creating an additional category, *BuildingHeight*. For simplicity, each floor was approximated to have 10 ft (3.048 m) floor-to-ceiling height.

IV. Simulation Environment

The generated metropolitan area used in this paper was based on the Manhattan borough of New York city. The urban area was discretized at $2m \times 2m$ resolution to avoid missing small buildings while providing sufficient clearance for the multicopter to fit within a single cell. Note, given that all simulations were performed only inside the borough of Manhattan, all location feature data was translated with respect to the left-most and lower-most points of the tax lots represented by MapPLUTO. Using the shifted coordinates of the polygons for each tax lot $(ShapeX, ShapeY)$ and the approximate height for a building $BuildingHeight$ for each tax lot a 3D map and 2D height-based heatmap of Manhattan were generated as shown in Figure 1.



(a) A 3D approximation of the Manhattan borough of New York City using PLUTO and MapPLUTO.

(b) A 2D heat map of the Manhattan borough of New York City depicting the variation in building heights.

Figure 1: Simulation Environment for Multicopter Flight over Manhattan.

V. Flight Planning

Flight planning prescribes a trajectory or waypoint sequence that enables the sUAS to reach its destination safely. Accurate weather forecasts, battery or fuel level, sUAS location, and information concerning the area where the sUAS is flying are used by the flight planner to calculate an optimal baseline flight plan. In-flight replanning to find an emergency landing destination and a path to that location during a high-risk situation is imperative particularly for a multicopter flying in a densely populated urban environment.

A. Protocol 1: Clearance to Land

The "Clearance to Land" map shown below was generated using the building information provided by the city database. The map highlights all buildings within the area that have given consent for a multicopter to land on their rooftops. For simplicity, each building was randomly assigned a "Clearance to Land" flag prior to simulation. Note, it assumed that if a building has given clearance to land on there are no obstacles on its rooftop and the rooftop geometry guarantees a safe landing. In combination with the city's height map shown in Section IV, the flight planner can use this information to calculate a flight plan for the multicopter.

The first proposed fail-safe protocol prescribed in this paper assumes that the multicopter has access to the city building database and the "Clearance to Land" map previously described. To determine the best fail-safe landing location, the planner searches for the nearest building that has a set clearance to land flag and that has a height shorter than the multicopter's altitude by at least 2 m when the fail-safe landing system was initially engaged. Note these criteria are not absolute, particularly the altitude constraint, yet the altitude preference does manage energy used for climb. The goal rooftop site is then set as the center of the building's rooftop and a trajectory is generated using map-based A* over the $2m \times 2m$ grid discussed above.

To utilize A* search, a graph is created using the generated city database. Each node is given a unique ID, location value, and height value with respect to the information provided by the database for each of the cells in the discretized environment. If a nodes's location does not lie inside one of the tax lots, as described by $ShapeX$ and $ShapeY$, then it is assigned a height value of zero. This simplification is consistent with the underlying assumption that any grid cell without a height specified by the database corresponds to a street or to a cell lying in one of the surrounding bodies of water. Note, for simplicity, neighboring landmasses

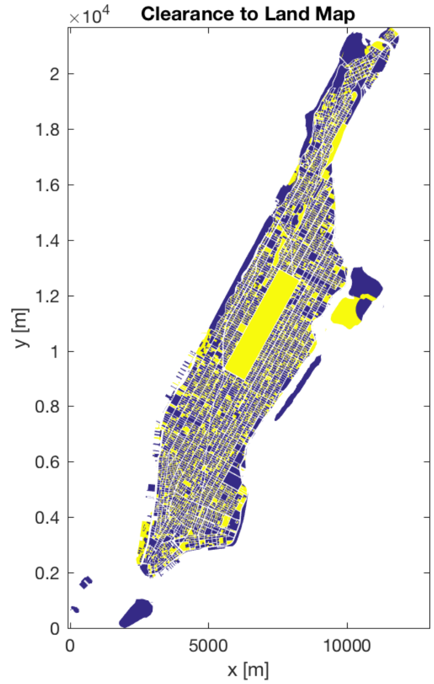
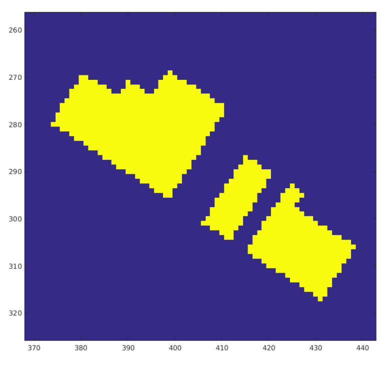


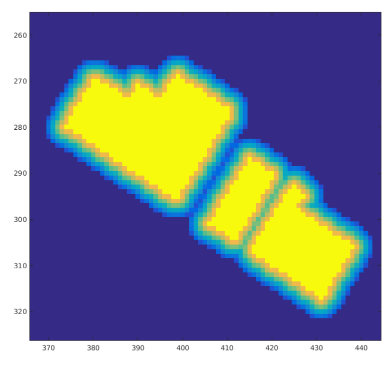
Figure 2: Yellow areas depict buildings or open areas that are likely to offer unobstructed safe landing sites for a sUAS multicopter.

outside Manhattan are not considered in this paper.

Graph edges are created by connecting adjacent nodes given their location values. To assign edge costs, the initial altitude of the multicopter is taken into consideration. If the node corresponds to a building with a maximum height below the multicopter's height by at least two meters then the cost to travel to that node is set to one. If the node corresponded to a building that did not meet the height requirement the cost to travel to that node is set to 100. However, given the optimality of A*, generated trajectories using these edge costs typically lead to solutions that travel along building walls rather than climbing over the buildings. To avoid this problem, a 10 m linearly repulsive field term is injected into A* cost along the building perimeter.



(a) Cost map of the area with obstacles representing buildings taller than the multicopter's flight altitude.



(b) Rejectance term is introduced into the search area cost map to avoid flying too close to buildings with A*.

Figure 3: Rejectance Term

The equation below summarizes the cost function used by A* to find a constant altitude flight plan to the selected building rooftop for the multicopter to safely land.

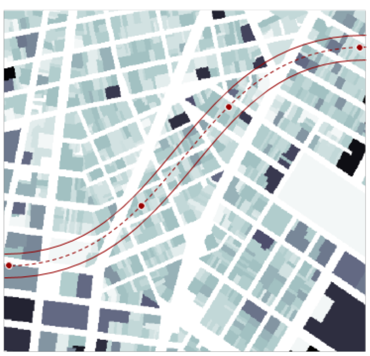
$$f(n) = g(n) + h(n) \tag{1}$$

where $f(n)$ denotes the estimated total cost to the selected rooftop, $g(n)$ represents the cost to travel from the moment the fail-safe protocol is initiated until reaching node n with respected to the cost map of the area, including the building repulsion term, and $h(n)$ represents a Euclidean distance heuristic between node n and the landing goal node.[11]

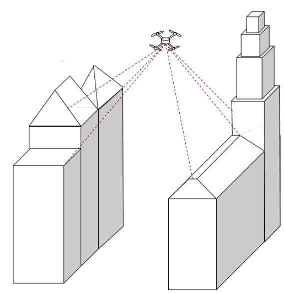
B. Protocol 2: Observation Tube

The second proposed alternative fail-safe flight protocol assumes that the multicopter does not have a "Clearance to Land" map of the area. The multicopter then must reference land use information provided by the city database to determine a tentative landing site, likely an open area within expected reach (i.e., within the multicopter's landing footprint). For this study landing site candidates were restricted to parks and similar areas of predominantly open space. Once the multicopter selects a landing site, it generates a flight plan using the same map-based A* framework discussed earlier and heads towards this destination.

However, for this case, the multicopter is assumed to carry an onboard overhead camera it can use to observe nearby rooftops enroute. The resulting observation tube" may allow the multicopter to find a viable rooftop in which case it plans an alternative fail-safe path to land on this rooftop. This alternative protocol avoids possible future unsafe scenarios in were to continue executing its original flight plan. The multicopter's real-time data collection and an example observation tube are shown in Figure 4.



(a) Observation tube given the field of view of an camera traveling along a proposed trajectory



(b) Multicopter analyzing nearby rooftop geometries in search of an alternative potential landing site

Figure 4: Depiction of the multicopter observing enroute rooftops for a possible alternative landing site

To determine whether a building is suitable for landing, the multicopter must fly sufficiently near the building for an overhead camera (or LIDAR) to characterize the rooftop's geometry. A non-flat rooftop or a rooftop with a steep angle can prevent the safe landing of the multicopter. For simplicity, to account for the different rooftop types (e.g. shape, slope, area) each building was randomly assigned a "Rooftop Landing Probability" in the simulated city map from a uniform distribution. If the examined rooftop meets a predetermined probability threshold, established by some desired landing confidence level, for safe landing the multicopter will initiate its landing protocol.

C. Protocol 3: STC Coverage

For the final case study, the multicopter utilizes known building locations and their approximate geometries to generate a coverage plan that aid in the search of a possible landing site. Spanning tree- coverage (STC) provides an approximate plan to traverse any gridded area with obstacles.[12] STC can be extended to an online algorithm to plan around detected obstacles, but in this paper we only consider a coverage planning case where the given area height map generated from the database is presumed accurate.

To implement STC, the search area is discretized with grid cells of side length 2ϵ, where the value of ϵ depends on the onboard sensor's field of view. All grid cell centers are treated as nodes of an undirected graph with edges formed between adjacent cells in the area not corresponding to an obstacle, a building taller that the multicopter's height at the time of planning. A minimum spanning tree is constructed such that it reaches all nodes of in the graph connected to the start node, nearest grid center at the time of planning, without any intermediate cycles using Prim's algorithm.[13] Prim's algorithm utilizes a greedy approach to find a minimum spanning tree connecting all nodes in the connected graph containing the start node.

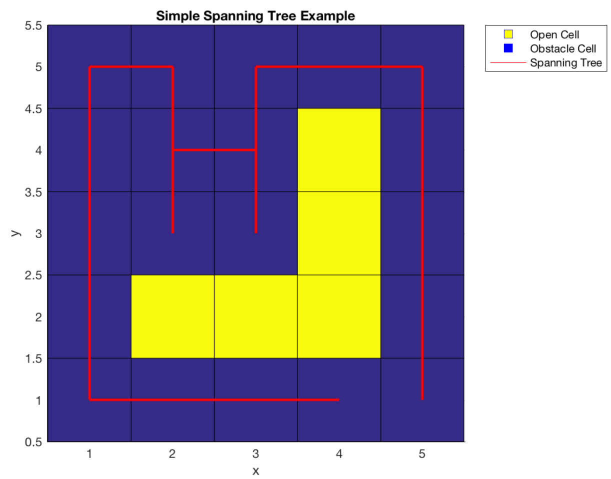


Figure 5: Example of a minimum spanning tree constructed to cover all obstacle free areas.

A viable coverage flight path is determined such that it circumnavigates the spanning tree counterclockwise, returning to the start node if the multicopters failure state (e.g., energy level) allows. This is accomplished by further discretizing each $2\epsilon \times 2\epsilon$ cell into four subgrids. The multicopter then "sweeps" a subgrid path while remaining on one side of the generated spanning tree at all times. A path is then determined such that it circumnavigates the spanning tree, returning to the starting grid if possible. Note that the spanning tree algorithm does not guarantee that the multicopter will identify a safe rooftop landing site given that rooftop landing suitability is unknown prior to executing the coverage plan. This protocol and any other of the protocols in which a safe landing site cannot be reached could be supported by the auto-land fail-safe available on existing multicopters if absolutely necessary to at least allow the multicopter to slowly descend, under autopilot control, into the (potentially populated) urban area before control authority is lost.

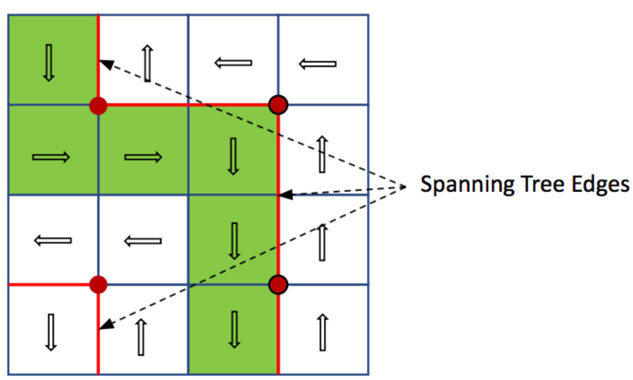


Figure 6: Example of multicopter "sweeping" a subgrid path along the edges of the spanning tree

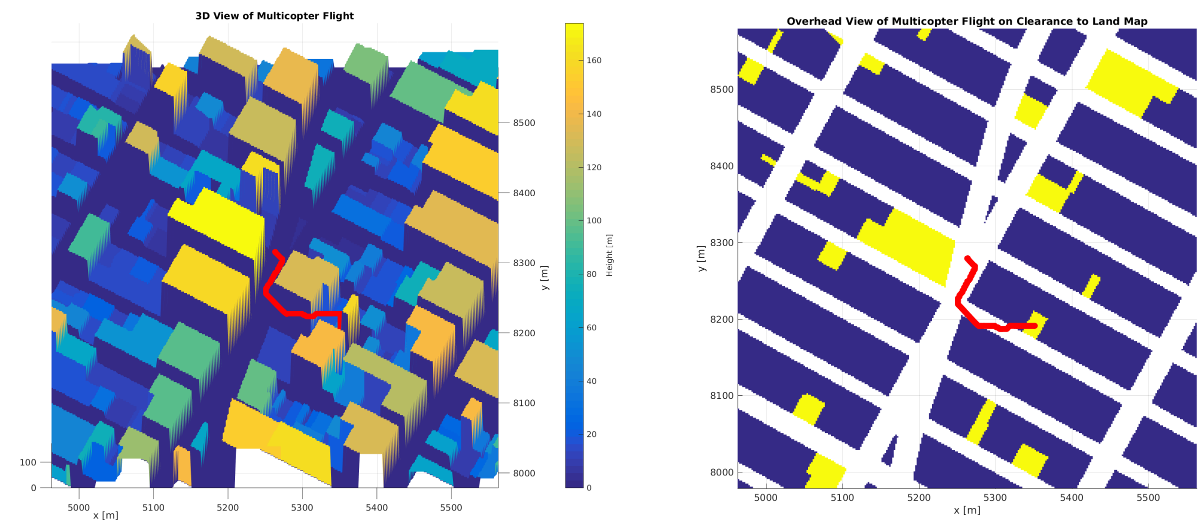
VI. Results

The following case studies consider an autonomous news drone flying over Times Square to provide aerial coverage of the annual New Year's Eve celebration. The simulation begins when the multicopter detects abnormal battery behavior (e.g., unexpectedly low voltage) triggering a fail-safe protocol. The multicopter does not have sufficient battery energy to return to its designated home point and given the densely populated area below it must select an alternative landing site beyond sensor field of view. The multicopter prioritizes a quick response such that the multicopter can land safely in time to avoid an uncontrolled descent into the populated square below. Equipped with the three protocols described above, the multicopter must determine which of the proposed protocols is applicable given available onboard database information.

A. Case Study 1: Clearance to Land

The multicopter is assumed to be flying at a height of 100m directly above above Times Square, corresponding to the center of the local area map. For this case study, "Clearance to Land" building rooftop flags are available in the onboard database. Using this information the multicopter was able to find the closest viable rooftop that could host a safe landing. In this particular simulation, the selected target location corresponds to the rooftop of a nearby retail store.

Using the multicotper's starting location and the target rooftop, a flight plan was generated using map-based A* using the local area cost map with and added rejectance term. Using a PID controller in combination with a waypoint based guidance function, the multicopter was capable of safely navigating through the urban environment to land on the selected rooftop. The resulting traveled trajectory is shown by Figure 7.



(a) 3D flight of multicopter onto nearest viable rooftop as suggested by the database

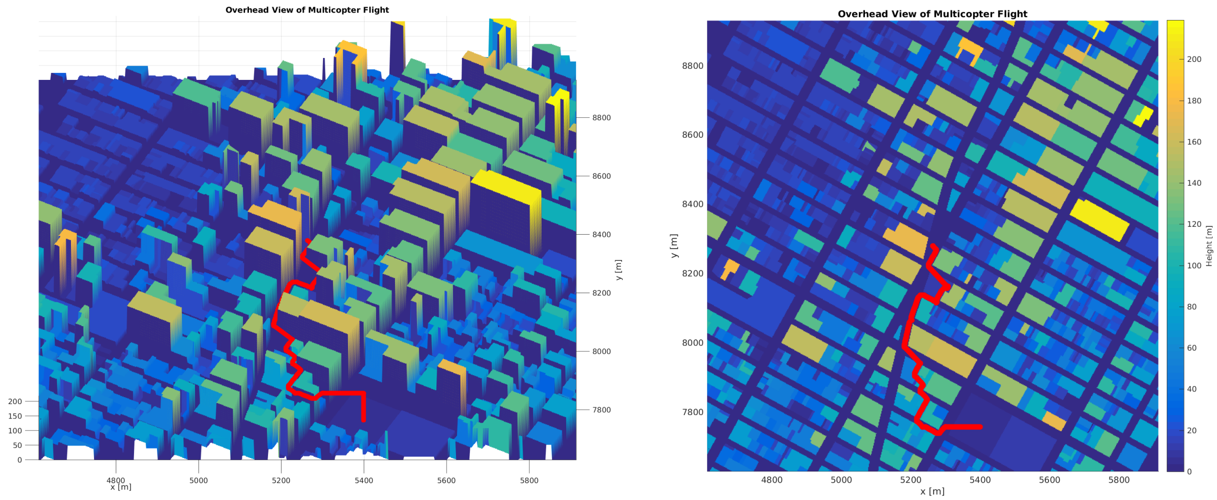
(b) Overhead view of multicopter flying to its destination depicted on the local Clearance to Land Map

Figure 7: Multicopter traveling to the nearest rooftop with a "Clearance to Land" flag.

B. Case Study 2: Observation Tube

As in Case 1, the multicopter is assumed to be flying at a height of 100m directly above above Times Square when it experiences an unexpected low battery fail-safe trigger. However, for this case the multicopter does not have access to "Clearance to Land" information in an onboard database. The multicopter therefore set a tentative land site that could serve as a last resort landing site from land use information provided by the database. The multicopter thus selected the nearest park in the area and generated a flight plan using map-based A*. In this particular simulation, the map-based landing site initially identified corresponds to Bryant Park in Midtown Manhattan.

Using an onboard camera, the multicopter can observe the rooftops below within the field of its camera as discussed in Section V. The discovery of a viable rooftop to be used as a landing site depends on the field of view of camera itself as well as the initial route of flight. In addition, the selection of a candidate rooftop depends on a landing site confidence level relative to the original site selected from the map. Both of these metrics can cause the multicopter to pass a potential rooftop landing site along its trajectory. Using a stringent desired landing probability measure of 0.95, Figure 8 demonstrates the traveled trajectory of the multicopter eventually reaching Bryant Park.

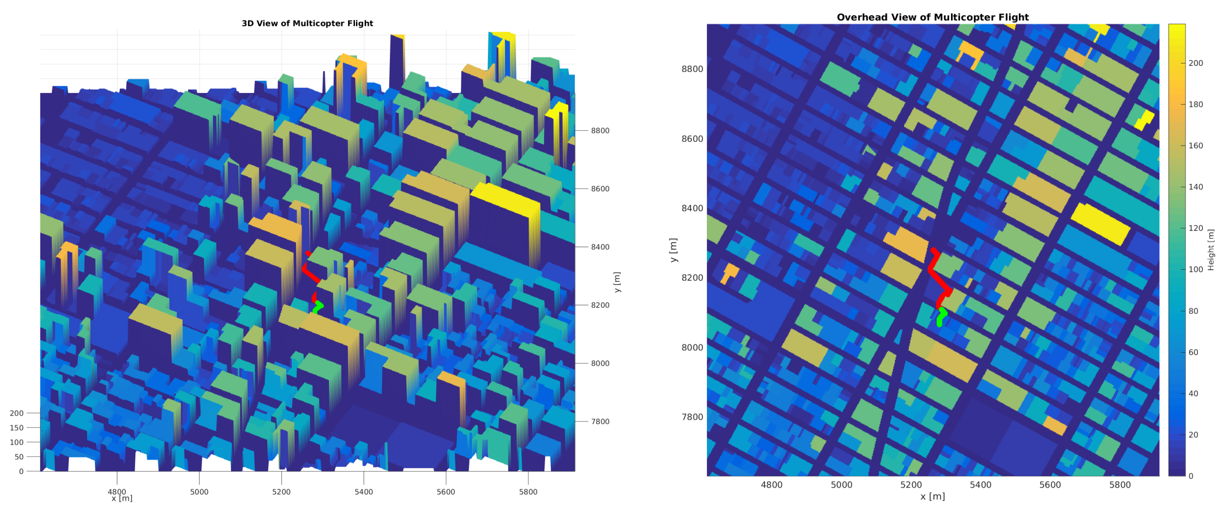


(a) 3D view of multicopter traveling to Bryant Park as a tentative landing site

(b) Overhead view of multicopter traveling to Bryant Park as a tentative landing site

Figure 8: Flight of multicopter to tentative destination when no alternative rooftop landing site is discovered.

In contrast, Figure 9 demonstrates finding an alternative landing site as the multicopter heads towards Byant Park. Using a less stringent desired landing probability measure of 0.85, the multicopter discovered a viable landing rooftop enroute. In this particular simulation the selected landing location corresponded to a nearby parking structure. This demonstrates how the fusion of database information and sensor observations can be used to make a more informed landing site selection decision in the event of an emergency.



(a) 3D view of multicopter flying to to an alternative rooftop landing site upon discovery.

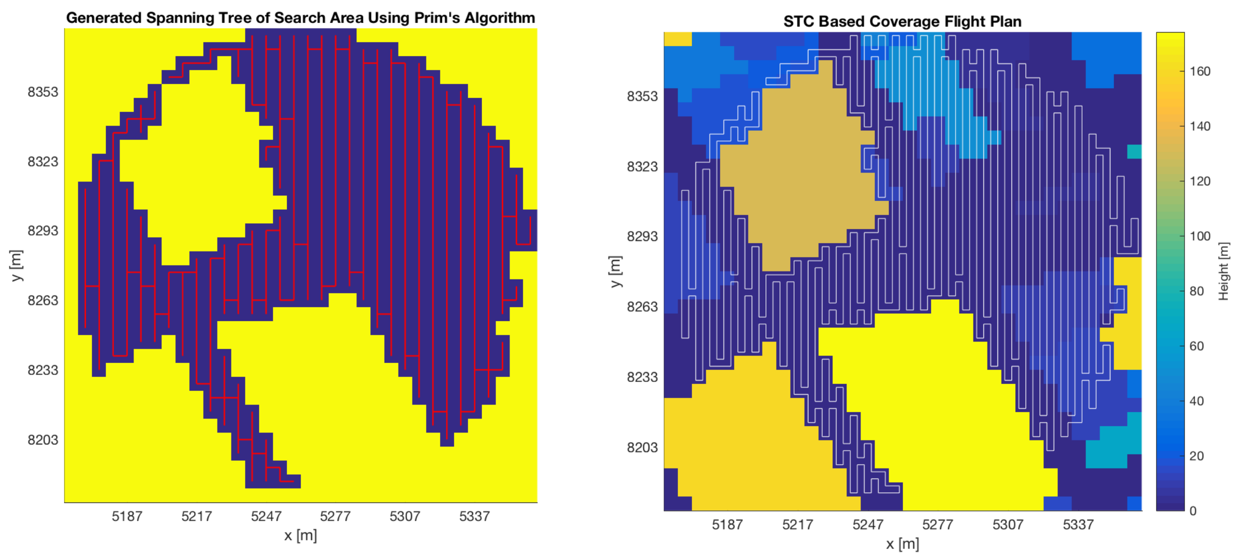
(b) Overhead view of multicopter flying to an alternative rooftop landing site upon discovery.

Figure 9: Flight of multicopter to alternative rooftop landing site is discovered.

C. Case Study 3: STC Coverage

For this last case study, we examined a situation in which the multicopter has no relevant landing site selection information stored onboard (e.g. no “Clearance to Land” or land use data). Given the densely populated streets below and lack of information to select a potential landing site beyond the sensors’ field of view, the multicopter generated a coverage plan of the area within a constrained search radius. For this particular simulation a search radius of 100m was selected. Typically this search radius might be larger to span multiple city blocks.

Figure 10 depicts a generated minimum spanning tree of the exploration area using Prim’s algorithm. A uniformly weighted graph of the search area is generated with respect to the area cost map and the multicopter’s start location and altitude. The start location was used to designate the start node of the graph while the multicopter’s altitude was utilized to determine the existence of edges within the discretized constrained search area. Using this minimum spanning tree a constant altitude coverage flight plan was created, illustrated below, that can be used by the multicopter to finding a viable rooftop for landing enroute. While no ideal rooftop might be detected, the acceptability threshold can be adjusted over time to enable selection of either the first fully viable rooftop or the best rooftop observed within some energy margin constraint (or deadline).



(a) Minimum spanning tree generated in local search area using Prim’s algorithm

(b) STC based coverage flight plan to be executed in search of viable landing site

Figure 10: STC based constant altitude flight plan within a constrained 100m radial search area around Times Square

VII. Conclusions and Future Work

Three alternative fail-safe protocols for autonomous urban multicopter flight were proposed and tested in simulation for a multicopter flying over the borough Manhattan in New York City. Utilizing nontraditional database and sensor fusion to make an informed landing site selection, the multicopter was able to autonomously determine the best flight plan to execute given the data available.

Given access to “Clearance to Land” flag data for each building in the city, the multicopter is able to safely navigate through the city and land on the closest viable building rooftop. If such information was unavailable but the database provided information regarding land use, the multicopter is able to generate a tentative flight plan to follow while searching for a viable rooftop along its trajectory based on real-time sensor feedback. In the worst case, if only a map of the area was known with no specific rooftop or land use information, the multicopter is able to generate an STC-based coverage algorithm to explore the area in search of a viable rooftop (or clear area).

In the future, this work can be extended to account for localization uncertainty given the inaccuracies of GPS in an urban canyon by using state estimation techniques. In addition, simultaneous localization and mapping (SLAM) techniques and online STC based coverage flight planning could be used when flying in unfamiliar urban areas, prompting online versions of the proposed protocols. Finally, given access to real-time data sources such as cell-phone usage, such methods could be extended to include viable ground landing sites in areas with low population densities or data-based occupancy estimates.

VIII. Acknowledgement

This work was supported in part under NASA Cooperative Agreement NNX16AH81A.

References

[1]Fern, L., Rorie, R. C., and Shively, R., "UAS Contingency Management: The Effect of Different Procedures on ATC in Civil Airspace Operations," *14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014, doi: 10.2514/6.2014-2414.

[2]"3DR - Drone & UAV Technology - Fail-Safe Overview," https://3dr.com/kb/failsafe-overview/ Accessed May 30, 2016.

[3]Stevens, M. N. and Atkins, E. M., "Multi-Mode Guidance for an Independent Multicopter Geofencing System," *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3150.

[4]Luxhoj, J. T., "System Safety Modeling of Alternative Geofencing Configurations for small UAS," *International Journal of Aviation, Aeronautics, and Aerospace*, Vol. 3, No. 1, 2016, pp. 2.

[5]Ten Harmsel, A. J., Olson, I. J., and Atkins, E. M., "Emergency Flight Planning for an Energy-Constrained Multicopter," *Journal of Intelligent & Robotic Systems*, 2016, pp. 1–21, doi: 10.1007/s10846-016-0370-z.

[6]Rufa, Justin R.Atkins, E. M., "Unmanned Aircraft System Navigation in the Urban Environment: A Systems Analysis," *Journal of Aerospace Information Systems*, Vol. 13, No. 4, 2016, pp. 143–160.

[7]Leutenegger, S. and Siegwart, R. Y., "A low-cost and fail-safe Inertial Navigation System for airplanes," May 2012, pp. 612–618, doi: 10.1109/ICRA.2012.6225061.

[8]Mao, G., Drake, S., and Anderson, B. D. O., "Design of an Extended Kalman Filter for UAV Localization," Feb 2007, pp. 224–229, doi: 10.1109/IDC.2007.374554.

[9]Rady, S., Kandil, A. A., and Badreddin, E., "A hybrid localization approach for UAV in GPS denied areas," Dec 2011, pp. 1269–1274, doi: 10.1109/SII.2011.6147631.

[10]"Open Data - NYC.gov," 2016, http://www1.nyc.gov/site/planning/data-maps/open-data.page Accessed Dec 1, 2016.

[11]Russell, Stuart JNorvig, P., *Artificial intelligence*, Prentice Hall, 1st ed., 2009.

[12]Gabriely, Y.Rimon, E., "Spanning-tree based coverage of continuous areas by a mobile robot," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*.

[13]Prim, R. C., "Shortest Connection Networks And Some Generalizations," *Bell System Technical Journal*, Vol. 36, No. 6, 1957, pp. 1389–1401.