2018 International Conference on Unmanned Aircraft Systems (ICUAS)
Dallas, TX, USA, June 12-15, 2018

Comprehensive Risk-based Planning for Small Unmanned Aircraft System Rooftop Landing

Jeremy Castagno[1], Cosme Ochoa[2], and Ella Atkins[3]

Abstract— **The expected proliferation of Unmanned Aircraft Systems (UAS) has prompted many to question their safety and reliability, particularly in urban areas. Failures and anomalies can lead to the need for emergency landing, which in turn requires the UAS operator or autonomy to rapidly identify and evaluate the risks for possible landing sites and trajectories to reach these sites. This paper proposes a method to optimize the overall emergency landing site and flight path risks. Although sensors can scan an immediate area, no safe site might be observable in which case pre-processed data on more distant safe sites is required. For example, in urban regions, out of sight flat rooftops may pose less risk to people and property than landing in streets or sidewalks. This paper proposes the offline construction of a landing site database using a variety of public data sources, uniquely allowing for the assessment of risk associated with a rooftop landing. A real-time *map-based* planner is presented that demonstrates a novel trade-off between landing site risk and path risk and provides a heuristic to improve decision-making efficiency.**

I. INTRODUCTION

Unmanned Aircraft Systems (UAS) are expected to proliferate in urban and rural environments. What was once a technology used primarily by the military is now the core technology of numerous commercial ventures [1]. Operation of UAS in densely-populated areas will require flight near buildings and over people. Safety must be a top priority for system designers and regulators as UAS will pose new risks to an overflown population.

One of the primary safety concerns is ensuring robust emergency landing capabilities [2], [3]. A controlled emergency landing requires landing site selection, trajectory planning to that site, and stable flight control to actually reach that site. When a UAS spots a safe site within sensor range, a safe landing can be immediately executed. In other cases, the UAS must explore beyond sensor range or utilize pre-processed map data to identify a safe site [4]. An onboard database of maps and even landing sites can be incorporated into an efficient autonomous decision making framework. For example, Refs. [5] and [6] utilize airborne flight risk models to build emergency landing plans for fixed-wing and urban flight operations, respectively.

This paper proposes a multi-objective optimization strategy to minimize combined risks posed during flight and touch down at the chosen landing site. Landing site and local area map information is pre-processed and stored onboard. A novel minimum integrated risk heuristic is proposed to guide an onboard emergency landing planner to explore the suite of landing sites likely to minimize overall landing risk. Pareto fronts over landing site and path risk are generated for an urban region in which flat and gently-sloped rooftops are marked as potential landing sites.

The primary contribution of this paper is in its approach to multi-objective optimization over landing site risk and path risk during emergency landing planning. A secondary contribution is our admissible heuristic to efficiently guide the landing planner to low-risk solutions. Risk metrics are quantified from offline construction of a database using public data sources. A risk map is proposed that supports distinction of risk due to the landing site and that offers a cost map used to asses risk for path planning. The proposed emergency planning framework enables a UAS to select a minimal-risk landing site while also considering path risk.

The paper is structured as follows. Section II provides background in emergency landing planning done using local sensing and map-based planners. Section III presents a problem statement including motivation for rooftop landing. Section IV discusses the offline construction of a risk-aware landing site database used for landing site selection as well as cost maps used for path planning. Section V summarizes the search strategy used for emergency landing planning. Section VI presents case studies to generate a landing site database and build emergency flight plans from multiple starting positions, focusing on analysis of the trade-off between landing site risk and path risk. Finally, Sec. VII presents conclusions and future work.

II. BACKGROUND

Emergency landing planning is typically accomplished with either onboard sensor-based planning or map-based planning [7], [8]. The former generates decisions based solely on real-time information provided by onboard sensors, while the latter utilizes map database information in decision-making. Sec. II-A summarizes sensor-based planning while an overview of map-based planning is presented in Sec. II-B. A meta-level framework to unify these methods has been proposed for general (fixed-wind) aviation [9] as well as multicopter UAS [8]. The planning architecture relies on utilizing local sensors to land if the immediate area is safe and to adopt a map-based planner otherwise. Our work focuses on map-based planning.

---

[1] Jeremy Castagno is a Robotics Institute PhD student, University of Michigan `jdcasta@umich.edu`

[2] Cosme Ochoa is a Robotics Institute PhD student, University of Michigan `cosme@umich.edu`

[3] Ella Atkins is a Professor of Aerospace Engineering and Robotics, University of Michigan `ematkins@umich.edu`

978-1-5386-1353-5/18/$31.00 ©2018 IEEE

A. Local Sensor Planning

Emergency landing planning using onboard sensors, e.g., cameras and radar, has demonstrated success in the identification of landing sites and subsequent controlled landings. Warren et al. used downward facing cameras to identify and characterize possible landing sites according to size, shape, slope, and nearby obstacles [7]. Others have demonstrated real-time control using an outer vision-based controller sending navigation commands to a low-level controller responsible for stable flight [10]. Experimental results with UAS were carried out in both studies and have demonstrated robust capabilities of using sensors for this general task. Ref. [11] describes successful experiments in autonomous local landing using video and scanning laser (lidar) data. Ref. [12] presents a risk map for trajectory planning that translates risk to probability of an air vehicle reaching its destination, but this map is vehicle-centric whereas our risk map is environment-centric, e.g., mapping risk posed to an overflown community and population during an emergency.

B. Map-based Planning

Sensor-based planners can only identify landing sites and assess risk with information available from local sensors, vehicle models, and health data; whereas a map-based planner would incorporate information on emergency flight risk beyond sensor line-of-sight. A map-based planner provides a database of possible landing site locations as well as local terrain and buildings for use in both risk mapping and flight planning. A holistic view of risk often takes into account failures or uncertain performance of the vehicle [12] as well as risk a UAS poses to people and property [13], [14], [8]. Pre-processed risk data can be quickly queried and used in real-time settings to safely and autonomously navigate a UAS to a low-risk landing site.

Previous work has utilized offline database planning strategies for both piloted fixed wing [9] and UAS systems [8]. Work by Donato et al. investigated the use of non-conventional data sources to construct risk metrics for emergency landing site selection with fixed wing aircraft [14]. Data such as census records, Open Street Maps (OSM), and mobile phone records were processed to quantify risk to overflown people, the vehicle itself, and property. The proposed planning architecture used an onboard database of landing sites to select a risk-optimal landing site in the event of an emergency. Path planning was performed at a constant altitude, assessing risk to people through fusion of census data and mobile phone activity.

Map-based planning for an energy-constrained multicoptor in an urban environment was studied in [8]. A 3D occupancy grid to represent risk was generated as a combination of terrain, population, and obstacle costs. Terrain cost was evaluated using slope as well as terrain type. Census data was used to determine population risk, and property risk was assigned equally to all buildings. Costs were weighted and added to construct a multidimensional grid for flight planning. In the Ref. [8] framework, a landing site is a *terrain* grid cell that has a lower cost than a configurable threshold. Note that all landing sites were treated equally so the primary contribution was in risk assessment for path planning without regard for relative landing site risk metrics.

Our work holistically trades off risk of landing sites and their risk-optimal paths. While others have used publicly-available data for collision avoidance and for selecting open field or road landing sites, our work proposes roof top landings based on roof shape as a risk metric. In addition we propose the fusion of OpenStreetMap (OSM) data with other sources (e.g. catalogued LIDAR data) to augment features sets such as providing building height information when not otherwise available. Refs. [8], [14], and [9] are the primary inspiration of this paper. We combine the strengths of risk-aware landing site selection in [14] with the cost maps for path planning by [8]. Data concerning roof shape is provided by OpenStreetMap (OSM); if missing, this information may be generated utilizing public LIDAR and satellite images [15]. The authors' previous work on roof shape analysis utilizing machine learning is used in this paper to provide building roof shapes [16]. Section IV discusses how to process multiple data sources to construct an offline database of possible landing sites and cost maps to be used in the proposed emergency planning framework.

III. PROBLEM STATEMENT

A. Preliminaries

While map data may be globally geo-referenced, low-altitude urban flights can be conducted with a local Cartesian reference frame. Let the orthogonal bases for this coordinate system be denoted $\hat{\mathbf{e}}_x$, $\hat{\mathbf{e}}_y$, and $\hat{\mathbf{e}}_z$. The position of the UAS body frame with respect to the local Cartesian reference frame can then be defined as:

$$O_{UAS} = x\,\hat{\mathbf{e}}_x + y\,\hat{\mathbf{e}}_y + z\,\hat{\mathbf{e}}_z = [x, y, z]. \tag{1}$$

When initiating an emergency landing planning cycle, a set of candidate landing sites are generated, defined as

$$\mathcal{S}_{ls} = \{l_i, \ldots, l_n\} \tag{2}$$

where each l_i refers to a landing site holding a set of properties defined as

$$l_i = \{\mathbf{c_i},\ r_{l,i},\ r_{p,i}\} \tag{3}$$

$$\mathbf{c_i} \in \mathbb{R}^3 \tag{4}$$

$$r_{l,i}, r_{p,i} \in \mathbb{R} \tag{5}$$

where $\mathbf{c_i}$ is the centroid of the landing site in the Cartesian reference frame, $r_{l,i}$ is the landing site risk in domain $[0, 1]$, and $r_{p,i}$ is the path risk. Landing site risk is calculated offline and represents the risk intrinsic to touching down at that landing site. Path risk must be calculated online and represents the path length and penalties (quantified as meters) associated with a path. The calculation of both metrics is expanded in Sec. IV-B and Sec. IV-D respectively.

B. *Planning Scenario*

During an emergency landing, the first objective is to identify safe landing sites nearby. Conventional emergency landing sites, such as parks or grasslands, are often sparsely distributed and unreachable in urban environments. The satellite image in Fig. 1 illustrates this challenge, and the need for a UAS to find alternative landing sites. Buildings with flat roofs are often unoccupied and can provide a safe landing site for a UAS [17]. Common roof shapes include flat, skillion (flat with slight slope), and gabled (slope on both sides); examples are labeled in Fig. 1. In this paper we consider rooftop landings possible given sufficient shape and size properties.

The proposed flight planner requires the vehicle to be able to execute a stable approach to an emergency landing site. There are a number of specific failure scenarios that can be detected and handled in time to execute such a landing. These scenarios include but are not limited to: UAS low energy, lost communication link, bad weather, non-essential sensor or actuator failure, operator emergency landing directive, and non-cooperative aircraft nearby. The methods and optimization techniques discussed in this paper can be used for any vertical take-off and landing (VTOL) aircraft. The case studies later presented are specific to a multicopter in an urban environment.

Fig. 1. Satellite image of an urban environment with multiple flat roof landing sites.

IV. OFFLINE DATABASE CONSTRUCTION

Shown in Fig. 2, database construction begins first by processing multiple Geographic Information System (GIS) data sources into a cohesive feature set. A feature is a geometric object, e.g., polygon or polyhedron, that represents an area of interest, e.g., a building, park, or grassland. These features also have attributes such as building height, roof shape, ground height, land type, and population density. An overview of constructing this feature set is discussed in Section IV-A. Section IV-B details the proposed domain-specific risk model that maps features and their attributes to a numeric risk value for landing site purposes. A final step, outlined in IV-C, converts this data to an embedded landing-site database used in real-time planning. Section IV-D discusses the construction of cost maps used during path planning.

A. *GIS Data Processing*

In order to construct a feature rich database, data is fused from OSM, LIDAR, census, and terrain height sources. This paper focuses on gathering information relevant to emergency landing for UAS in an urban environment, but the procedure is general for other planning problems. The first stage of database processing is to extract features from OSM as outlined in Sec. IV-A.1. Buildings are further processed to determine height information utilizing catalogued LIDAR data as outlined in Sec. IV-A.2. Finally Sec. IV-A.3 discusses how each landing site is assigned a ground height and average population density by consulting digital elevation models and census records, respectively.

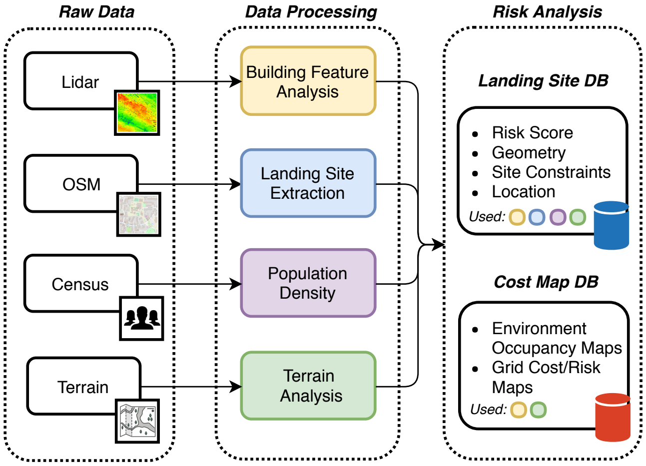


Fig. 2. Data Pre-processing for landing site and cost map database generation.

1) Open Street Map: OpenStreetMap (OSM) is an online collaborative mapping project that generates a worldwide database of geographic information [18]. OSM not only provides detailed information on buildings, terrain types, and roads, but it also provides classification of land use (e.g. military, school). The building blocks of OSM are *elements*, i.e., georeferenced geometries such as polygons for buildings. Every element can be provided semantics through the attachment of tags. A tag is composed of two free form text fields, a *key* and a *value*. For example, an element can have the tags "*building=residential*" and "*height=8*" attached to it, indicating that this element represents a private residential building 8 meters tall. Any key/tag combination is possible in OSM, however the community agrees on certain key/value pairs to generate an informal standard.

In order for this text-based data to be usable in a practical way, a domain specific processing/parsing step often combined with a subsequent storage step is needed. Many tools exist to aid this process and help in the construction of a

TABLE I

BUILDING TABLE FROM OSM

Column Name	Key	Values	Description
Height	height	All	Records height of bulding from ground plane to top of roof
Roof Shape	roof:shape	All	Records the roof shape of the building
Geometry	NA	NA	Georefrenced geometry

TABLE II

NON-BUILDING TABLE FROM OSM

Column Name	Key	Values	Description
Amenity	amenity	All	Describes an assortment of community facilities
Land Use	landuse	All	Describes the primary use of land by humans
Natural	natrual	All	Describes geological or landcover features
Leisure	leisure	All	Describes land areas where people go in their spare time
Geometry	NA	NA	Georefrenced geometry

database [19]. Using these tools, one can generate a database with two tabular data structures, one for buildings and one for non-buildings, that hold the defining geometry of each feature and other relevant attributes. Table I and II show the imported table structures of building and non-building features, respectively, and the OSM tags used to extract them. Further processing is discussed below to augment the database with additional attributes using other GIS data sources.

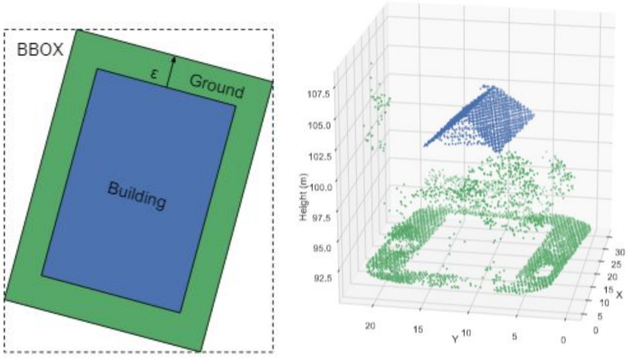
2) Building height estimation: Building height refers to its maximum height, specifically the distance between the top edge of a building (including the roof, but excluding antennas, spires, and other equipment mounted on the roof) and the lowest point at the bottom ground plane. This information is often incomplete from OSM and can be estimated using LIDAR data.

Our technique is shown in Fig. 3(a). The blue polygon (p_b) representing the building outline is expanded a configurable distance, ϵ, to construct the green polygon (p_g) not including the blue area. This green polygon's bounding box (BBOX) is used to quickly filter points in the LIDAR data set not related to the building of interest. This reduced LIDAR point set, L, is then quickly grouped into the ground set G and building set B defined as

$$G = \{ point \in L \mid p_g \text{ contains}\, point \}$$
$$B = \{ point \in L \mid p_b \text{ contains}\, point \}$$

A ray casting algorithm was chosen to determine if a point resides in each polygon [20]. Fig. 3(b) shows a 3D visualization of the LIDAR point separations. These two sets of z-coordinates can then be analyzed to determine the minimum ground height and maximum roof height of the building. To account for possible outliers, the bottom 2% and top 2% of ground and building points can be neglected respectively. The conservative building height estimate is then computed as $\text{height} = \max(B) - \min(G)$.

3) Population density and ground height estimation: Census records have long been used to estimate static population density [21], [14], [8] and will similarly be used in this research. Ground height can be estimated from a digital surface model (DSM), which provides height values of the terrain including vegetation. Census and DSM data sources are often released as a *raster*, or a matrix of cells (pixels) arranged into rows and columns where each grid cell contains a value representing information of interest. These rasters are geo-referenced, meaning each cell refers to a specific location on the surface of the Earth. One can determine the average population density and ground height for each feature in the database by using these data sources.



(a) (b)

Fig. 3. (a) Polygon representations of building outline (blue) and ground plane (green), (b) 3D visualization of LIDAR point cloud separation. Blue points are the building points; green points include the ground plane and nearby foliage.

This process is completed for each feature by rasterizing its vector geometry onto the raster dataset as shown in Fig. 4. The green dashed outline represents the vector feature geometry overlaid on the grid. The multiple lines defined by each connected vertex of the polygon are rasterized according to Bresenhams line algorithm [22]. All cells enclosed by this line rasterization are highlighted in blue, and the population density and ground height can be taken as the average of these cells. These attribute values are then added to the database for later access.

B. Landing Site Risk Model

A domain specific risk model is constructed that maps a landing site risk (r_l) to each feature in the database. Risk is decomposed as terrain cost (C_t), population cost (C_p), and property cost (C_s). Each of these risks are numeric values created from a functional composition of the attributes of each feature (building height, land type, etc.) and are outlined in Sec. IV-B.1 through Sec. IV-B.3. All costs are in the range

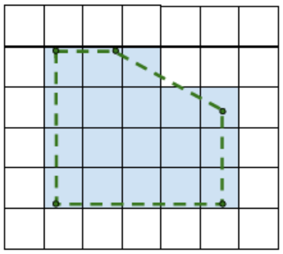


Fig. 4. Example rasterization process of a feature geometry (green dashed line) on a raster data set.

from $[0, 1]$ and are weighted together as follows:

$$r_l = w_t \cdot C_t + w_p \cdot C_p + w_s \cdot C_s \tag{6}$$

1) Terrain Cost: Terrain is any surface on which the UAS may land including building rooftops in this work. Terrain cost expresses the risk *to* the UAS and aims to capture how certain terrain types are more preferable to land on. This value is computed by assigning cost to the landing area terrain or rooftop *class* (C_c) as well as its size (C_a). After assigning costs to each of these factors one can compute a weighted sum for the terrain cost as shown:

$$C_t = w_c \cdot C_c + w_a \cdot C_a \tag{7}$$

The *class* of terrain, e.g., brownfield or skillion roof, is first contingent on whether the feature is a building or not. Building terrain is characterized by its roof shape attribute and is assigned a C_c cost as shown in Table III. Flat roofs are preferred while most other sloped roofs are heavily penalized. Non-buildings have a variety of keywords describing them from the imported OSM attributes. Following the taxonomy from [14], these non-building keywords are aggregated into groups and assigned costs as shown in Table IV. These cost values would be later refined by stakeholders.

TABLE III

BUILDING TYPE COST (C_c)

Group	Roof Shape(s)	Cost (C_c)
Group 1	flat	0
Group 2	skillion	0.5
Group 3	all others	1

Associating a cost to area is dependent upon the needs of the UAS while also considering the average area size of landings sites in the mission region. A function is needed that maps small areas to high risk (1), average areas to medium risk (0.5), and large areas to low risk (0). In addition such a function should allow specifying a minimum area, A_{min}, as well an average area A_{avg}. One possible function that fits this criteria is a piecewise exponentially decaying function. Area cost can then be computed as:

$$C_a(a) = \begin{cases} 1 & a \leq A_{min} \\ 2^{\frac{A_{min} \quad a}{A_{avg} \quad A_{min}}} & a > A_{min} \end{cases} \tag{8}$$

where a is the feature area.

TABLE IV

NON-BUILDING TYPE COST (C_c)

Group	Keywords	Cost (C_c)
Group 4	brownfield, grass, grassland village_green,greenfield	0
Group 5	meadow, cemetary, scrub, heath, farmland	0.25
Group 6	water, riverbank	1
Group 7	recreation_ground, garden, golf_course, track, pitch, playground, common, park	0.5
Group 8	parking	0.6
Group 9	industrial, commercial	1

2) Population Cost: A population cost C_p for each feature can be assigned by utilizing its population density attribute. A function is needed that maps high density to high risk and low density to low risk. Similar to area cost, a function can be generated which specifies an average population density, P_{avg}, mapped to a risk of 0.5:

$$C_p(p) = 1 - 2^{\frac{p}{P_{avg}}} \tag{9}$$

3) Property Cost: To account for the risk of the UAS damaging property, a property cost is assigned to each feature. Buildings are all treated equally, however building type can be later used from OSM to assign higher costs to certain building types. The same groupings mentioned previously in terrain cost are used to provide the mapping found in Table V.

TABLE V

PROPERTY COSTS C_s

Group(s)	Cost(C_s)	Description
Groups 1-3	0.5	Buildings
Groups 4-6	0	Non developed area
Groups 7-8	0.5	Developed area low cost
Group 9	1	Developed area high cost

C. Embedded Landing Site Database

The database discussed so far is a world wide feature database for which risk has been evaluated. A subset of this data composed of strictly feasible landing sites must be generated and stored onboard the UAS. We propose use of SQLite [23] with the GIS extension Spatialite [24] for this purpose. The Spatialite extension adds spatial indexing, meaning every landing site is indexed by its spatial region, allowing for fast queries using R*-tree search [25]. The database has a low memory and storage footprint, and is able to integrate with over 30+ languages including Python and C/C++ used by our team.

D. Cost Mapping and Path Planning

Optimal planning algorithms require a cost function to explore the search space in pursuit of a feasible and even optimal solution. For discrete search planners, the space must first be discretized into a graph $G(V, E)$, where V denotes the nodes (or vertices) with corresponding edge set E depicting any node transitions between configurations. One can then define costs associated with an individual node or edge transition as described below.

An A* path planner was used to validate the proposed emergency landing planning framework. The implemented planning scheme traded off between paths resulting in high risk to the aircraft (i.e., flying to close to nearby buildings) and the distance of travel required. To generate costs first a 3D representation of the operating environment was built from OSM and terrain databases as described above. The OSM databases provided geometric building information including their heights with respect to the ground plane. The terrain database specified the varying elevation of the ground plane as height above Mean Sea Level (MSL).

A set of feature maps were generated to account for risks to the aircraft as it flew in the generated 3D urban environment. First, a simple height-based feature map H_{map} was generated for the neighboring area which depicted the varying height of the landscape for some desired spatial resolution (x_{res}, y_{res}). Second, using the H_{map}, an obstacle occupancy mesh O_{mesh} was generated to define the presence of obstacles, e.g. buildings or the ground, in the environment at the same spatial resolution and for altitude-based resolution z_{res}. Distance to the nearest obstacles was calculated and encoded in an obstacle proximity feature mesh D_{mesh} using breadth-first search in pre-processing only. Finally, using these distances along with a distance threshold d_{thresh} the risk for a given (x_i, y_j, z_k) position in the mesh was calculated as follows:

$$R_{mesh}(i, j, k) = \max\left(1.0 - \frac{D_{mesh}(i, j, k)}{d_{thresh}}, 0\right) \qquad (10)$$

Similar to the repulsion field used in [8], our risk metric generates a linearly decaying risk potential field up to a defined distance threshold. All present obstacles are automatically assigned the maximum risk value, such as a tall building or high point of elevation with respect to the examined altitude. This multidimensional risk mesh is known as the cost map and is used by our A* path planner to generate optimal collision free paths from the current UAS position to a landing site. A* is used with cost $g(n)$ and heuristic $h(n)$ functions calculated as

$$c(n, n') = \text{dist}(n, n') \cdot (1 + \text{risk}(n')) \qquad (11)$$

$$g(n) = g(n') + c(n, n') \qquad (12)$$

$$h(n) = \text{dist}(n, goal) \qquad (13)$$

$$f(n) = g(n) + h(n) \qquad (14)$$

where c is the transition cost between the current node n and the previous node n', g is the cumulative cost, h is the heuristic cost, f is the estimated total cost, $\mathbf{dist}(\cdot)$ calculates the distance between nodes, and $\mathbf{risk}(\cdot)$ returns the risk encoded by the given node data structure. Once a path is found to the goal, the path risk, r_p, is the total accrued cost of the path to reach the goal node.

V. MAP-BASED PLANNER

This section describes the proposed real-time map-based planner to be used in emergency landing situations. Section V-A summarizes trade-offs between landing site risk and path risk, while Sec. V-B summarizes our real-time map-based planning method.

A. Trade-off between landing site risk and path risk

The planner must identify the optimal landing site and flight plan pair from the set of candidate landing sites $\mathcal{S}_{ls}$. The risk of each landing site *and* the path to that site presents a multi-objective (MO) optimization problem from which there may not be a single solution simultaneously optimal over both objectives. MO problems therefore require analysis of trade-offs by computing and analyzing a Pareto frontier when possible. The visualization of this frontier helps a system designer choose the relative weighting of objective trade-offs to select a single "best" solution [26].

An example Pareto frontier for our planner is shown in Fig. 5. Each dot represents a specific landing site. The x-axis represents landing site risk and the y-axis represents a normalized path risk to that site. The normalizing constant, R, represents a maximum path length the UAS is allowed to traverse. Note that this graph by itself makes no determination of the weighting or "worth" between the two objectives. The green line connects three points in the Pareto frontier, the set of non-dominated landing sites for which any improvement in one objective results in a negative trade-off in the other. Each of these three points is considered "optimal", and a weighting factor between each objective must be assigned to determine a final choice of weighted minimum total risk solution. Such a weighting scheme is proposed below for each landing site $l_i \in \mathcal{S}_{ls}$:

$$r_{t,i} = w_l \cdot r_{l,i} + w_p \cdot \frac{r_{p,i}}{R} \qquad (15)$$

where $r_{t,i}$ refers to the total risk and w_l and w_p are weights for landing site risk and path risk, respectively. The optimal landing site can then be found by solving the optimization problem shown in Eq. 16.

$$l^* = \underset{l_i \in \mathcal{S}_{ls}}{\arg\min}\ r_{t,i} \qquad (16)$$

B. Real-time Planner

A landing site database stored onboard can be used for landing site risk-assessment and selection, and a 3D regional risk-based cost map can be used for path planning. Large-area map data can be stored in the cloud and updated as needed over time. Before a UAS mission begins, a preflight download is performed which extracts a subset of the data constrained by the geographic bounds of the mission plan. These lightweight embedded databases can be used by the on-board map-based planner in landing site selection and path planning respectively as shown in Fig. 6.

In the event of an emergency landing scenario, the planner first generates a geographic footprint indicating the range of the vehicle. This work does not focus on the construction of

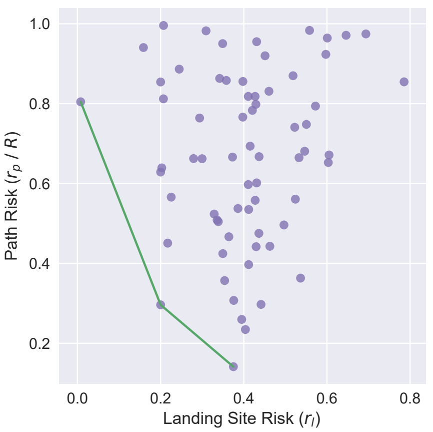


Fig. 5. Example trade-off between landing site risk and path risk. Points in the Pareto frontier are connected by the green line.

such a footprint, however work by [27], [9] and [8] have investigated its generation for fixed-wing and multicopter aircraft, respectively. For purposes of this paper the footprint is a sphere of radius R whose center is $\mathbf{O}_{UAS}$. The footprint is used to efficiently query the spatially-indexed landing site database to provide a list of risk-valued landing sites, $\mathcal{S}_{ls}$. Constraints, such as landing site height or area, are applied to filter undesirable locations and output a set of valid landing sites. If no valid landing sites are found constraints are relaxed first and the footprint expanded as a last resort.

As discussed in Sec. V-A, assessing risk in a global sense requires one to not only quantify the risk of each landing site but also the risk of the *path* to each landing site. However, optimal collision-free path planning inside three dimensional spaces can take a significant amount of time and is impractical to perform in real-time for the hundreds of potential landing sites that may be observed in an urban environment. Algorithm 1 proposes the prioritization of landing sites by estimating their total risk, a weighted sum with the intrinsic landing site risk and their *minimum* path risk. The minimum path risk, $\hat{r}_p$, is estimated as the Euclidean distance between the landing site and $\mathbf{O}_{UAS}$. The resulting set is then sorted by this estimated total risk.

Following this calculation, flight plans are generated in parallel for the top N landing sites to determine their true path risk, r_p. After this more computationally-expensive flight planning procedure, these N landing sites and associated paths are then prioritized as before except using r_p instead of $\hat{r}_p$. The minimum weighted risk landing site location and associated path are provided to the UAS flight controller and executed (when in an autonomous flight mode).

This heuristic does not guarantee global optimality because it is possible that the minimum path risk $\hat{r}_p$ substantially underestimates the true path risk r_p. However, this situation is mitigated by having N fully planned solutions providing actual path risk comparisons over multiple plans. Case studies shown in Sec. VI provide experimental results illustrating the effectiveness of this heuristic and compare it against the globally optimal solutions.

Algorithm 1: Heuristic Landing Site Prioritization

```
Input  : Landing Site Set ($\mathcal{S}_{ls}$),
         UAS Location ($\mathbf{O}_{UAS}$)
         Footprint Radius ($R$),
         Weighting Trade-off ($w_l, w_p$)
Output: Sorted $\mathcal{S}_{ls}$ by estimated total risk, $\hat{r}_{t,i}$
foreach $l_i \in \mathcal{S}_{ls}$ do
    $\hat{r}_{p,i} = ||\mathbf{O}_{UAS} - \mathbf{c_i}||$
    $\hat{r}_{t,i} = w_l \cdot r_{l,i} + w_p \cdot \frac{\hat{r}_{p,i}}{R}$
end
sort $\mathcal{S}_{ls}$ by $\hat{r}_{t,i}$
return $\mathcal{S}_{ls}$
```

VI. SIMULATION RESULTS

All case studies were simulated in a 2.5 km^2 city block of Witten, Germany. Witten was chosen because a large amount of high quality publicly-accessible data sources are available for the area. High resolution raw LIDAR data is available as well a professionally post-processed Digital Terrain Models (DTMs) providing ground elevation [28]. In addition the city of Witten has the desirable characteristic of having the most densely-labeled roof shapes in the OSM database, as well as all buildings being identified. Census raster data from 2011 is available in 100 meter square blocks [29].

Data was processed as outlined in Sec IV-A to generate a landing site database as well cost maps which were 9.6 MB and 88 MB in size, respectively. Table VI shows the weighting parameters chosen to assign risk for all case studies presented. The map-based planner weighting parameters for landing site risk vs path risk, w_l and w_p, were set at 0.6 and 0.4, respectively. The number of sites chosen to plan in parallel, N, was set at four. The footprint radius, R, was set at 200 meters. The 3D cost map resolution was 2 meters in all directions with a d_{thresh} of 10 meters. All simulations were run on a quad-core processor clocked at 3.0 GHz.

TABLE VI

ALL CASE STUDY PARAMETERS FOR RISK METRICS

Risk	Parameter	Value	Description
LS Risk	w_t	0.5	Weight for terrain risk
	w_p	0.25	Weight for population risk
	w_s	0.25	Weight for property risk
Terrain Risk	w_c	0.5	Weight for feature type
	w_a	0.5	Weight for feature area
	A_{avg}	150 m^2	Average area
	A_{min}	10 m^2	Minimum area
Population Risk	P_{avg}	1 $\frac{ppl}{100m^2}$	Average Population

Fig. 7 displays a map of Witten overlaid with highlighted valid landing sites. The colors of each polygon range from light yellow to dark orange representing the total intrinsic risk of each landing site. As expected, large open areas such as Lutherpark are assigned a lower risk, while smaller buildings with undesirable roof shapes are assigned higher risk.

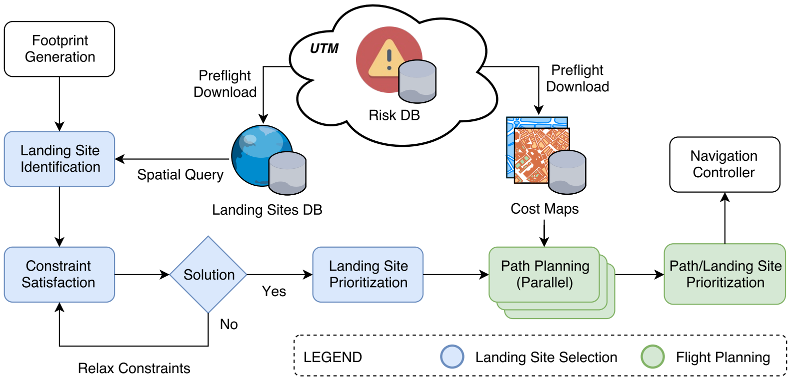


Fig. 6. Proposed map-based planner modified from [9].

A total of 3,492 landings sites are available, where $\sim 70\%$ are buildings and the remainder non-buildings. Summary statistics of the landing site risk distribution are as follows: $mean = 0.57, std = 0.13, min = 0.01, max = 0.85$. These statistics indicate that few low risk landing sites are present in this region, while the majority of landing sites present high risk to people and property.

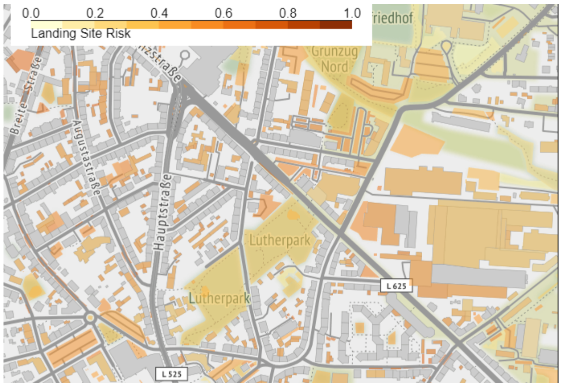


Fig. 7. Map of Witten, Germany. Each colored area is a landing site with the color determining the landing site risk.

A. Case Study Scenarios

Six emergency landing planning scenarios are presented at different initial locations in Witten, Germany. Scenarios 1 and 2 demonstrate emergency situations in which ideal landing sites are far away, yet favorable sites are close by. Scenario 3 and 4 hold the fortunate (yet rare) situation in which many desirable landing sites are nearby while in Scenarios 5 and 6 only high risk landing sites are available. Table VII displays summary information for each scenario. Scenario 1 is given more attention to highlight the components of the map-based planner algorithm.

Fig. 8 displays the first scenario and highlights the trade-off which often must be made between landing site risk and path risk. UAS position is denoted by the blue center marker which is surrounded by available landing sites colored by their risk. The blue numbered circles indicate the landing sites with the lowest intrinsic risk while the red circles indicate the ranked landing sites chosen by the proposed map-based planner. The discrepancy between these two sets is a result of the map-based planner also considering the path risk involved in reaching each landing site. As Sec. V-B details, the map-based planner uses the Euclidean distance between the UAS and the landing site to estimate the minimum path risk, which is illustrated by the straight lines drawn from the first blue and red circle markers to the UAS location. The map-based planner then computes the weighted risk sum to arrive at the rankings displayed by the red numbered circles.

TABLE VII
SCENARIO LOCATIONS AND TIMING STATISTICS

Scenario	Lng/Lat (degrees)	Height (m)	Execution Time (ms)
Scenario 1	51.4400, 7.3400	140	315
Scenario 2	51.4371, 7.3372	120	307
Scenario 3	51.4355, 7.3363	115	303
Scenario 4	51.4383, 7.3392	125	323
Scenario 5	51.4435, 7.3357	130	310
Scenario 6	51.4427, 7.3377	140	273

In order to assess the effectiveness of the map-based planner heuristic, optimal collision-free paths to *all* landing sites for each scenario were generated, providing their actual path risk. Fig. 9 displays a scatter plot for each scenario, with an x-axis of landing site risk and y-axis of path risk. Path risk was normalized by the radial footprint R. Each scenario graph has the same axis limits, allowing one to compare each scenario visually. Each purple dot represents a landing site, while the red dots represent the top four ranked landing sites chosen by the map-based planner with the black arrow indicating the first. The Pareto set for each scenario is depicted by the green line.

The graphs in Scenarios 1 and 2 demonstrate how a small negative trade-off in landing site risk (moving right in the graph) can lead to significant reductions in path risk (going down in the graph). One can see the map-based planner taking advantage of this scenario by selecting landing sites with higher landing site risk but significantly reduced path risk. In scenarios 3 and 4 the map based planner does not need to make any significant trade-off, ideal landing sites are

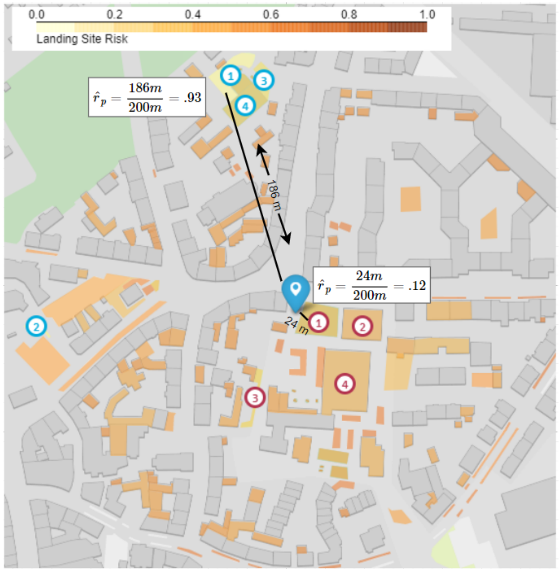


Fig. 8. Landing site selection from Case Study 1. The center blue marker indicates UAS position. Red numbered circles indicate the ranked choices from the map-based planner using *estimated* total risk, the weighted combination of estimated path risk and landing site risk. Blue numbered circles indicate the best landing sites *only* considering landing site risk.

immediately available and chosen. Scenario 5 and 6 show a situation where the Pareto frontier is shifted far to the right, indicating the lack of favorable landing sties. An important observation to note is that the final chosen landing site for all scenarios lies on the Pareto frontier and strike an appropriate balance between landing site risk and path risk.

Landing site total risk is calculated using Eq. 16. The map-based planner for each scenario is detailed in Table VIII and displays landing site type, ranking, estimated total risk, and total risk over the actual flight plan and landing site. A ranking of 1 indicates the optimal solution was chosen, while higher rankings indicate that the path heuristic underestimated the true path risk leading to non-optimal solutions. The map-based planner did not significantly underestimate the path risk leading to choosing the optimal solution in all cases . The heuristic performed well in this study because the UAS position at the point of failure was above buildings and near-straight (or near-direct) paths to the landing site were possible for examined scenarios.

TABLE VIII

MAP-BASED PLANNER CHOICE

Scenario	Type	Ranking	$\hat{r}_t$	r_t
Scenario 1	park	1 / 163	0.287	0.297
Scenario 2	garden	1 / 170	0.381	0.401
Scenario 3	grass	1 / 102	0.155	0.165
Scenario 4	grass	1 / 184	0.200	0.210
Scenario 5	flat roof	1 / 125	0.312	0.320
Scenario 6	flat roof	1 / 122	0.380	0.410

B. Random Tests

To more fully assess the robustness of the real-time map-based planner, 100 random cases were generated. Initial state planar coordinates were randomly generated but the height of each UAS start position was chosen to be just above the terrain surface, anticipating that the Euclidean distance heuristic would underestimate the path risk by forcing the path to maneuver around buildings. As before, optimal paths for all landings sites were created providing the true path risk and the same risk weights for landing site risk and path risk from the previous case studies were used. Equation 16 was used to determine total risk for each random test case and rank the map-based planner's top choice in the solution space. The map-based planner chose the optimal solution for all 100 random test cases. For each choice the heuristic underestimated the total risk by an average of 6% with a maximum underestimation value of 20%.

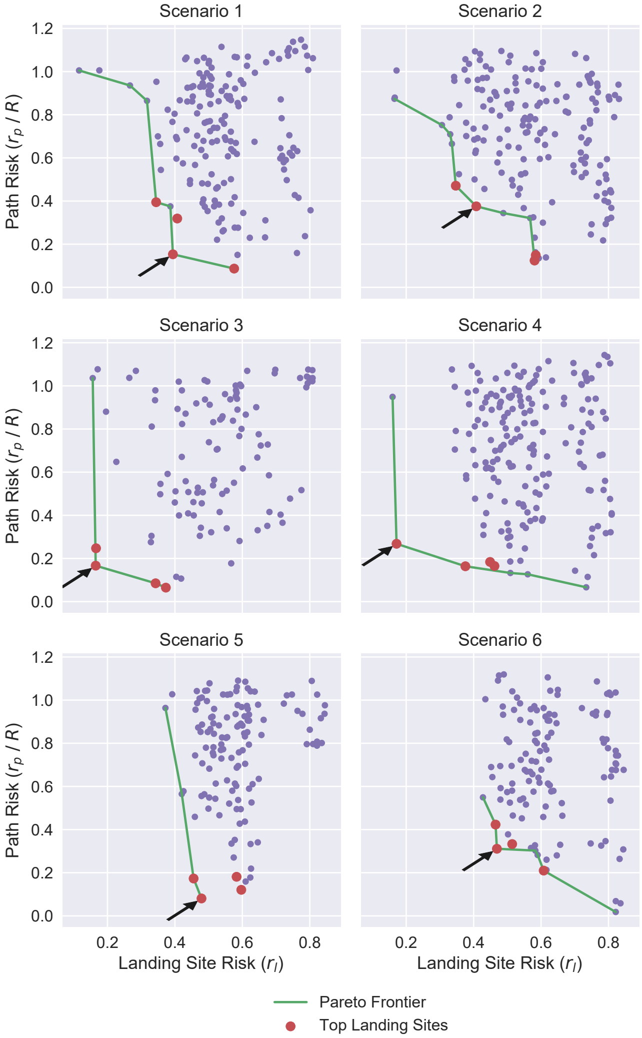


Fig. 9. Experimental results for scenarios 1-6. The x and y axes are landing site risk and path risk respectively. Each purple dots represents a landing site and its associated path, while red dots are top selections by the map-based planner. The black arrow indicates the planners final selection.

VII. CONCLUSION

This paper proposes a map-based emergency landing planner that explicitly considers landing site risk and path risk to determine a solution. While many other researchers have looked at one metric or the other, this paper quantifies both metrics and selects a solution optimizing the trade-off between landing site and path risk metrics. This paper also includes rooftop landing sites and estimates risk based on the building's roof shape and size. These rooftops provide numerous additional viable landing sites to VTOL aircraft in emergency situations that previously were restricted to surface-based landings. This work relies on data preprocessing that assimilates multiple public data sources to generate landing sites and their risks as well as risk-based cost maps for path planning.

Experimental results in Sec. VI indicate that the proposed map-based planner can quickly and robustly identify landings sites and generate paths during emergency scenarios. The identification of landing sites on average took less than 30 milliseconds by taking advantage of the spatially-indexed landing-site database, while path-planning on average took 300 milliseconds. Note these timings are specific for the environment and the 200 meter radial footprint chosen. Urban centers with higher roof tops, such as Manhattan, may provide significantly different results as paths would tend to follow the street and avenue grid in the urban canyon. The heuristic proposed to rank landing sites by a weighted sum between minimum path risk and landing site risk proved to be quite effective in presented case studies, arriving at the globally optimal solution for all tests performed. While this heuristic performed well in Witten, Germany, further work should be done to investigate its efficacy in more dense urban canyon settings. Future work should also be done to handle situations where no landing sites are found within the search footprint, mandating reliance upon local sensors for an immediate landing.

ACKNOWLEDGEMENT

This work was supported in part by NASA Grant NNX16AH81A and DoD STTR 17-479-02 under a subcontract with Bihrle Applied Research, Inc.

REFERENCES

[1] D. Rodrigues, R. de Melo Pires, E. A. Marconato, C. Areias, J. C. Cunha, K. R. L. J. Castelo Branco, and M. Vieira. Service-oriented architectures for a flexible and safe use of unmanned aerial vehicles. *IEEE Intel. Trans. Sys. Magazine*, 9(1):97–109, Spring 2017.

[2] J. A. Winnefeld and F. Kendall. Unmanned systems integrated roadmap FY 2011-2036. *Office of the Secretary of Defense. US*, 2011.

[3] M. T. DeGarmo. Issues concerning integration of unmanned aerial vehicles in civil airspace. *Center for Advanced Aviation System Development*, 2004.

[4] C. A. Ochoa and E. M. Atkins. Fail-safe navigation for autonomous urban multicopter flight. In *AIAA Information Systems-AIAA Infotech@ Aerospace*, page 0222. 2017.

[5] N. Meuleau, C. Plaunt, D. E. Smith, and T. Smith. An emergency landing planner for damaged aircraft. *Proc. of the 21st Innovative Applications of Artificial Intelligence Conf.*, pages 71–80, 2009.

[6] Shankar Sankararaman. Towards a computational framework for autonomous decision-making in unmanned aerial vehicles. In *AIAA Information Systems-AIAA Infotech @ Aerospace*. American Institute of Aeronautics and Astronautics, jan 2017.

[7] M. Warren, L. Mejias, X. Yang, B. Arain, F. Gonzalez, and B. Upcroft. Enabling aircraft emergency landings using active visual site detection. In *Field and Service Robotics*, pages 167–181. Springer, 2015.

[8] A. J. T. Harmsel, I. J. Olson, and E. M. Atkins. Emergency flight planning for an energy-constrained multicopter. *Journal of Intelligent & Robotic Systems*, 85(1):145–165, Jan 2017.

[9] E. M. Atkins, I. A. Portillo, and M. J. Strube. Emergency flight planning applied to total loss of thrust. *Jnl. of Aircraft*, 43(4):1205–1216, 2006.

[10] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Visually guided landing of an unmanned aerial vehicle. *IEEE transactions on robotics and automation*, 19(3):371–380, 2003.

[11] C. Theodore, D. Rowley, A. Ansar, L. Matthies, S. Goldberg, D. Hubbard, and M. Whalley. Flight trials of a rotorcraft unmanned aerial vehicle landing autonomously at unprepared sites. In *Annual Forum Proc. American Helicopter Society*, volume 62, page 1250, 2006.

[12] C. Goerzen and M. Whalley. Minimal risk motion planning: a new planner for autonomous uavs in uncertain environment. In *AHS Intl. Specialists Meeting on Unmmaned Rotorcraft, Arizona*, 2011.

[13] T. Patterson, S. McClean, P. Morrow, G. Parr, and C. Luo. Timely autonomous identification of uav safe landing zones. *Image and Vision Computing*, 32(9):568–578, 2014.

[14] P. F. A. Di Donato and E. M. Atkins. Evaluating risk to people and property for aircraft emergency landing planning. *Journal of Aerospace Information Systems*, 2017.

[15] F. Alidoost and H. Arefi. Knowledge based 3d building model recognition using convolutional neural networks from lidar and aerial imageries. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016.

[16] J. D. Castagno and E. M. Atkins. Automatic Classification of Roof Shapes for Multicopter Emergency Landing Site Selection. *ArXiv e-prints*, February 2018. Paper Accepted to 2018 AIAA Aviation Forum.

[17] Vishnu R. Desaraju, Nathan Michael, Martin Humenberger, Roland Brockers, Stephan Weiss, Jeremy Nash, and Larry Matthies. Vision-based landing site evaluation and informed optimal trajectory generation toward autonomous rooftop landing. *Autonomous Robots*, 39(3):445–463, jul 2015.

[18] OpenStreetMap. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2018.

[19] Omniscale. Imposm3. `https://github.com/omniscale/imposm3`, 2017. Visited on 2017-01-14.

[20] Joseph Thomas Samosky. *SectionView–A system for interactively specifying and visualizing sections through three-dimensional medical image data*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1993.

[21] R. A. Clothier, R. A. W., N. Fulton, and D. A. Campbell. A casualty risk analysis for unmanned aerial system (uas) operations over inhabited areas. 2007.

[22] J. E. Bresenham. Ambiguities in incremental line rastering. *IEEE Computer Graphics and Applications*, 7(5):31–43, May 1987.

[23] D. Chen, X. Han, and W. Wang. Use of sqlite on embedded system. In *Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on*, pages 210–213. IEEE, 2010.

[24] A. Furieri. Spatialite. `https://www.gaia-gis.it/fossil/libspatialite/index`, 2017. Visited on 2017-01-14.

[25] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In *Acm Sigmod Record*, volume 19, pages 322–331. ACM, 1990.

[26] P. Ngatchou, A. Zarei, and A. El-Sharkawi. Pareto multi objective optimization. In *Intelligent Systems Application to Power Systems, 2005. Proc. of the 13th Intl. Conference on*, pages 84–91. IEEE, 2005.

[27] S. Paul, F. Hole, A. Zytek, and C. A. Varela. Flight trajectory planning for fixed-wing aircraft in loss of thrust emergencies. *arXiv preprint arXiv:1711.00716*, 2017.

[28] NWR Data license Germany - Attribution - v2.0. `https://www.govdata.de/dl-de/by-2-0`, 2017. Accessed: 2017-09-05.

[29] Federal Statistical Office of Germany. Germany Census 2011. `https://www.zensus2011.de/SharedDocs/Aktuelles/Ergebnisse/DemografischeGrunddaten.html`. Visited on 2017-01-14.