Urban Metric Maps for Small Unmanned Aircraft Systems Motion Planning

Cosme A. Ochoa* and Ella M. Atkins †

University of Michigan, Ann Arbor, MI, 48109, USA

Low-altitude urban flight planning for small Unmanned Aircraft Systems (UAS) requires accurate vehicle, environment maps, and risk models to assure flight plans consider the urban landscape as well as airspace constraints. This paper presents a suite of motion planning metrics designed for small UAS urban flight. We define map-based and path-based metrics to holistically characterize motion plan quality. Proposed metrics are examined in the context of representative geometric, graph-based, and sampling-based motion planners applied to a multicopter small UAS. A novel multi-objective heuristic is proposed and applied for graph-based and sampling motion planners at four urban UAS flight altitude layers. Monte Carlo case studies in a New York City urban environment illustrate metric map properties and planner performance. Motion plans are evaluated as a function of planning algorithm, location, range, and flight altitude.

Nomenclature

β	Lidar beam elevation angle
ζ	Flight plan/path
δ_b	Bounding box buffer distance
δ_c	Graph connectivity
δ_d	Time of day
δ_r	Map resolution
δ_z	Planning altitude
ϵ_{UERE}	GPS error term
$\Gamma_{comm}, \Gamma_{resi}$	Commercial/residential area population modifier
$\mathcal{B}, \hat{\mathcal{B}}$	Regular/buffered operating bounding box
$C_{free}, C_{obs}, C_{tot}$	Obstacle-free/obstacle/total configuration space
$\mathcal{D}$	Date-time information
$\mathcal{G}, T, V, E$	Search graph/tree/nodes/edges
$\mathcal{H}_{\delta_r, \delta_z}$	Metric cost map set
$\mathcal{H}_{gps}$	GPS uncertainty map
$\mathcal{H}_{lidar}$	Lidar visibility map
$\mathcal{H}_{norm}$	Min-max normalized map
$\mathcal{H}_{obs}$	Obstacle occupancy map

---
*PhD Candidate, Robotics Institute, Student Member cosme@umich.edu
†Professor, Department of Aerospace Engineering, Fellow ematkins@umich.edu

$\mathcal{H}_{pop}$	Population density map
$\mathcal{H}_{risk}$	Proximity risk map
$\mathcal{H}_{total}$	Total cost map
$\mathcal{L}$	Total bounding box
Q_S, Q_G	Start/goal vehicle state
$\mathcal{W}$	Weighting vector
Ω_{obs}	Obstacle set
ρ_{rc}	GPS receiver range
b_{lidar}	number of Lidar beams
b_o	Lidar beams' origin
c	Speed of light
d_{close}	Distance to closest obstacle surface
d_{euc}, d_{oct}	Euclidean/octile distance
d_{thresh}	Proximity risk distance threshold
f, g, h	Total cost, cost-so-far, cost-to-go heuristic
G_{gps}	GPS pseudorange linear system
$GDOP_{thresh}$	GDOP threshold
h_{dist}, h_{plus}	Distance and multi-objective heuristics
idx, idy, idz	State indices
k_{lidar}, s_{lidar}	Number of Lidar scan positions/returns
m_m, m_p	Map/path-based metrics
$m_{dist}, c_{dist}, w_{weight}$	Distance traveled metric/cost/weight
$m_{gps}, c_{gps}, w_{gps}$	GPS pseudorange uncertainty metric/cost/weight
$m_{lidar}, c_{lidar}, w_{lidar}$	Lidar-based visibility metric/cost
m_{obs}, c_{obs}	Obstacle occupancy metric/cost/weight
$m_{pop}, c_{pop}, w_{pop}$	Overflown population density metric/cost
$m_{risk}, c_{risk}, w_{risk}$	Obstacle proximity metric/cost/weight
N_{sats}	number of visible satellites
pop_{census}	Census raw population count
pop_{mod}, pop_{norm}	Modified/maximum population count
r_{lidar}	Lidar visible range
t_{rc}, t_{sat}	GPS receiver/satellite clock
x, y, z	Aircraft position (inertial frame)
X_{gps}, Σ_{gps}	GPS pseudorange state vector/covariance
x_{rc}, y_{rc}, z_{rc}	GPS receiver position (inertial frame)
$x_{sat}, y_{sat}, z_{sat}$	GPS satellite position (inertial frame)

I. Introduction

A motion planner constructs a feasible and efficient kinodynamic path through a potentially complex environment connecting an initial location to a target or goal state [1]. Motion planning algorithms have been used for a wide range of Unmanned Aircraft Systems (UAS) applications including search & rescue [2, 3], reconnaissance missions [4, 5], sense & avoid [6, 7], and navigation through unmapped or uncertain environments [8–10]. Motion planners can complement onboard sensor suites to aid in conflict resolution [11] and alternative fail-safe protocols [12] for urban flight. Baseline flight plans are computed and approved prior to flight, but real-time planning may be required to effectively respond to changes in the mission, environment, and/or vehicle performance (e.g., system degradation, failure). Motion planners typically optimize solutions over path distance, time, and obstacle/terrain avoidance with benchmarks as discussed in [13]. Recent papers have presented flight risk metrics that augment traditional distance/time/obstacle avoidance cost terms [14–18].

This paper proposes a suite of complementary motion planning metrics designed for urban multicopter flight that further augments distance/time and risk-based metrics. We define map-based (m_m) and path-based (m_p) metrics to generate holistic cost-minimum plans in representative geometric, graph-based, and sampling-based motion planners. Map-based metrics (m_m) describe the UAS operating environment by constructing a collection of GPS/lidar navigation performance, population density, and obstacle risk exposure maps. Traditional path-based metrics (m_p) account for UAS energy consumption and distance traveled along a planned path. This paper presents a detailed analysis of map-based and path-based metrics in Monte Carlo case studies.

Map-based metrics are derived offline from open-source geospatial, satellite imagery, and census data. Each database is processed and transformed into discretized metric maps representative of the borough of Manhattan in New York City at different map resolutions and small UAS (sUAS) above ground level (AGL) flight altitudes. GPS satellite availability, lidar visibility, and risk to an overflown population are captured. Motion planning metric maps are examined with respect to a portfolio of motion planners. Distance-only and weighted multi-objective cost function results are compared. To improve performance, a multi-objective heuristic function for graph-based and sampling-based path planners is proposed. Monte Carlo case study results are presented as a function of metric weightings, planner type, and urban canyon settings. Planner metric usage and solution path properties are discussed.

The contributions of this work are as follows:

- This paper defines a comprehensive suite of urban UAS flight planning metrics and describes how to transform open-source data into metric maps applicable across different motion planners.
- This paper presents representative geometric, graph-based, and sampling-based motion planners and describes how metric maps are deployed in each.
- A novel multi-objective heuristic function is defined to improve upon a traditional distance-only heuristic. This heuristic is applied and evaluated in graph-based and sampling-based motion planners.

• Monte Carlo simulations are evaluated to analyze the properties of motion plans generated with different cost metrics and different planning algorithms.

Below, Sec. II summarizes related work followed by a problem statement (Sec. III). Sec. IV defines map-based and path-based motion planning metrics followed by a description of the process by which discretized feature maps are generated (Sec. V). A representative portfolio of motion planners is defined in Sec. VI, and our novel multi-objective admissible heuristic is introduced. Map-based metric results are presented in Sec. VII. Monte Carlo simulation process is summarized in Sec. VIII, and path planning results are evaluated in Sec. IX. Sec. X concludes the paper.

II. Related Work

This section first discusses background in metrics relevant to small UAS urban motion planning followed by background in motion planning approaches to sUAS operating in and over urban environments.

A. Planning Metrics

Qualitative and quantitative metrics inform a planner about the vehicle, its environment, preferences and constraints. Algorithm metrics can be defined from learned performance models [19, 20], statistical measures [21], abstract features [22], and classical algorithm properties [23] as summarized in Table 1. Additional metrics can be defined to incorporate application-specific considerations.

Table 1   Classical motion planning algorithm properties.

Property	Description
Completeness	A solution is returned if one exists; otherwise, failure is returned.
Soundness	If a solution is returned, it is feasible.
Complexity	Memory usage and/or execution time measured with theoretical upper bounds and/or large-scale Monte Carlo simulation.
Kinodynamics	Planning solutions are consistent with vehicular performance constraints.
Environment	Description of environment as static or dynamic.
Uncertainty	Planner accounts for uncertainty in vehicle or environment states.
Optimality	A best solution is returned with respect to a given metric or combination of metrics.

In practice, a motion planner should be complete and return an optimal feasible solution in real-time, if necessary, while satisfying all kinodynamic constraints. Motion planners trade off different objectives to find a balanced solution [24]. Distance traveled and flight risks per Table 2 may be considered. Distance traveled captures expected energy expenditure and estimated time of arrival (ETA) at a destination, while risk metrics may account for non-ideal vehicle and environment properties. Our work primarily considers an environment risk metric map since vehicle performance and weather are dynamic entities that do not map to fixed Earth-based coordinates.

For real-time aerospace applications, completeness, soundness, and bounded computational complexity are desired

Table 2   Common risks encountered by small UAS.

Type	Description	Examples
System	A hardware or software failure resulting in a system freeze, coding error, reboot, or component failure.	Deadlock [25, 26], overheating [27], electrical shorts [28], software risks [29]
Actuators	Control surfaces are irresponsive or fail to reach a target configuration given a threshold.	Shaft failures [30], PWM relay errors [31], pneumatic/hydraulic faults [32]
Sensors	Onboard sensing tools provide inaccurate representations of the world around them.	Faulty sensors, obstructed view, drifting sensor readings, urban canyon effects
Weather	Hazardous climate conditions influencing system sensing and/or performance.	Cold impact on batteries [33], poor visibility, snow/ice, turbulent winds [34]
Environment	Operating in hazardous areas that could potentially injure or harm nearby structures or people.	Proximity to buildings [35], flying over people [15], navigating unmapped areas

algorithm properties. Fixed-wing aircraft typically optimize cruise altitude (atmospheric density), airspeed, climb rate, lift/drag ratio [36, 37], and hazardous weather avoidance [38] but do not consider ground-based obstacles due to their substantial cruise altitude. Multicopter UAS operate at much lower altitudes thus typically optimize motions over clearance from obstacles, distance / time, and mission requirements [39]. Communication [40] and navigation [41] metrics are key considerations where line-of-sight signals may be blocked. A Pareto front analysis offers insight into balancing competing metrics [16, 42–44].

B. Motion Planning

The following paragraphs summarize different motion planning strategies and their respective advantages and disadvantages for small UAS urban motion planning.

Geometric motion planners provide rapid analytical solutions by constructing paths using points, lines, and arcs. In a two-dimensional Euclidean space, visibility graphs [45, 46] can be used to generate minimum length paths from intersecting lines for a holonomic system. Dubins [47] and Reeds-Shepp [48] curves account for nonholonomic turning constraints by adding turning radius arc segments to a path as needed. Geometric planners generate solutions rapidly but make simplifying assumptions, e.g., obstacle-free environments.

Graph-based planners search for solutions in a graph defined to assure mapped obstacle avoidance. A motion planning space can be covered with a uniform or nonuniform grid or with a roadmap, e.g., visibility graph [1]. By connecting the start and goal configurations to the graph, the motion planning problem is reduced to searching the graph for a minimum-cost path. A* [49] and its variants (Dijkstra [50], LPA* [51], ARA* [52], D* Lite [53], Field D* [54], Theta* [55]) are among the popular graph search strategies adapted to motion planning. Graph-based planners thrive in low-dimensional configuration spaces to provide optimal solutions with arbitrarily-complex cost functions and constraints implicitly handled in the graph. However, their performance advantage diminishes as the dimensionality of the motion planning state-space increases.

5

Sampling-based planners use randomly drawn node samples from an underlying probability distribution to generate a local graph iteratively. Probabilistic roadmaps (PRMs) [56] and rapidly exploring random trees (RRTs) [57, 58] paved the way for sampling algorithms aimed at managing the high dimensionality problem of graph-based planners. Innovations in the past decade have resulted in asymptotically optimal variants (e.g., PRM*, RRT* [59]) with improved convergence rates demonstrated in FMT* [60] and BIT* [61]. These algorithms are probabilistically complete but may not offer solutions in the presence of narrow passages or dense obstacle sets.

Optimization-based planning methods construct a solution by minimizing a cost function while satisfying constraints, i.e., the boundary value problem [62]. Potential field methods [63, 64] ignore dynamics and optimize a distance-based gradient along competing goal-attractive and obstacle-repulsive manifolds. Optimal control [65] applies physics-based constraints and costs to minimize time, energy, and potentially obstacle avoidance using smooth spatiotemporal mathematical functions. The functional nature of optimization-based methods supports analyzing nonlinear, multiple input-output, and time-varying systems but at the cost of computational complexity and convergence challenges. Model predictive control [66–68] variants limiting computations to a finite future horizon and can use lookup tables to cache complex solutions for online use. Optimization methods are susceptible to the local minima; they are not guaranteed to converge to a satisficing or globally-optimal solution particularly in complex environments.

III. Problem Statement

This paper defines a suite of map-based metrics m_m and path-based metrics m_p to offer comprehensive environment and path cost for sUAS flight planners per Table 3. Map-based metrics must be generated from a hybrid suite of data sources representing obstacles, sensor availability, and risk sources. Data must be processed, discretized, and converted into feature-rich metric maps, combined with path-based metrics, to compute an optimal obstacle-free path to a targeted landing site as shown in Fig. 1.

Table 3   Motion planning metrics classified by type.

Metric	Description	Type
m_{gps}	GPS pseudorange position uncertainty	map
m_{lidar}	Lidar-based local map uncertainty	map
m_{obs}	Obstacle occupancy	map
m_{pop}	Overflown population estimate	map
m_{risk}	Proximity to obstacles en route	map
m_{dist}	Distance traveled along a path	path

For a given operating bounding box $\mathcal{L}$, a collection of metric maps $\mathcal{H}$ must be generated to describe all m_m in Table 3. Each $\mathcal{H}$ is generated by explicitly calculating that metric value at every point characterized by a Cartesian grid over $\mathcal{L}$ with resolution δ_r for fixed flight altitude δ_z. To explore these cost metrics, representative *geometric*, *graph-based*, and
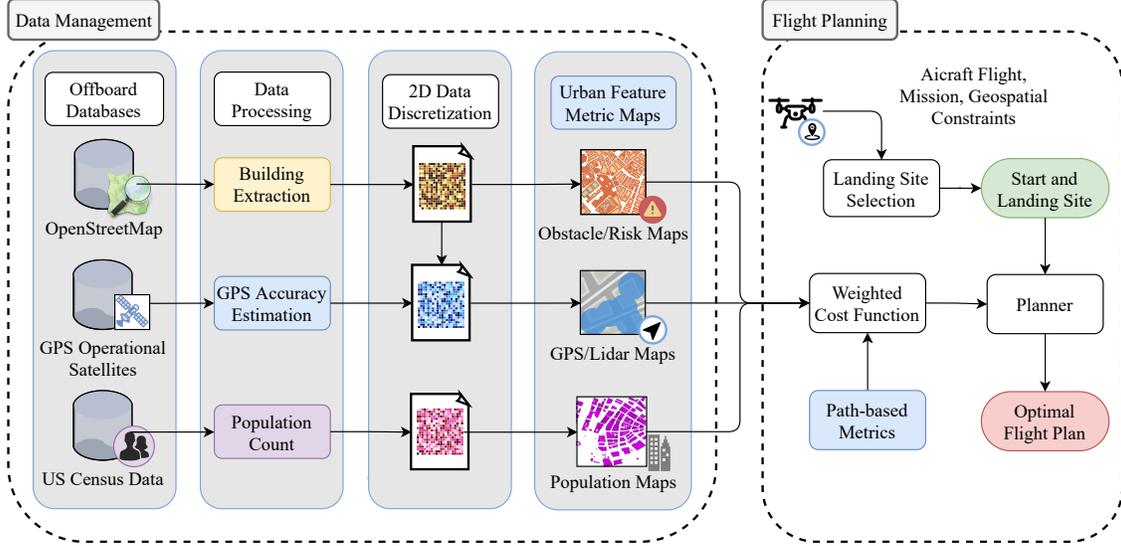
Fig. 1   Data flow for map-based metric generation in data-driven multicopter flight planning.

sampling-based motion planning algorithms are defined for urban multicopter flight. Because a traditional Euclidean distance motion planning heuristic h_{dist} does not provide information about any of the map-based metrics, a novel multi-objective heuristic h_{plus} is proposed and compared to h_{dist}. A suite of Monte Carlo case studies in Manhattan illustrate metric map and motion plan properties in a representative urban environment at four UAS flight altitude layers. Motion plans are evaluated as a function of planner, location, range, and flight altitude.

IV. Metric Definitions

A. GPS Uncertainty

GPS receivers communicate with a global navigation satellite system (GNSS) to estimate their geographical location using trilateration. Given receiver/satellite pairs, a pseudorange measurement is estimated as [69]:

$$\hat{\rho}_{rc,sat} = \rho_{rc} + c(t_{sat} - t_{rc}) + \epsilon \tag{1}$$

where ρ_{rc} is receiver range, c is the speed of light, t_{sat} and t_{rc} are the satellite/receiver clock readings, and ϵ_{UERE} captures any User Equivalent Range Errors (UEREs), e.g., atmospheric, clock, signal, and multipath errors.

Geometric dilution of precision ($GDOP$) describes error propagation from satellite geometry: dispersed satellites reduce uncertainty while clustered satellites increase it [70]. $GDOP$ can be expressed as:

$$GDOP(x, y, z, t) = \sqrt{PDOP(x, y, z)^2 + TDOP(t)^2} \tag{2}$$

where $PDOP$ and $TDOP$ are position/time dilutions of precision, respectively. DOP values between 1 to 20 [71]

quantify GPS reliability as summarized in Table 4.

DOP	Rating	Description
1	Ideal	Highest precision possible.
$1-4$	Excellent	Measurements are considered accurate except for the most sensitive applications.
$4-6$	Good	Represents the minimum acceptable loss in accuracy.
$6-8$	Moderate	May still be used but only recommended in obstacle free environments.
$8-20$	Fair	Readings should be dismissed or only serve to compute a rough estimate.
>20	Poor	Unreliable and should not be used.

Table 4 DOP Value Rating [71].

For n visible satellites, pseudo ranges offer a fast approximation of $PDOP$. Applying a first-order Taylor expansion to the true range, pseudorange $\hat{\rho}_{rc,i}$ and range r_i to the ith satellite are computed as:

$$\hat{\rho}_{rc,i} = \frac{x_{rc} - x_{sat,i}}{r_i} x_{rc} + \frac{y_{rc} - y_{sat,i}}{r_i} y_{rc} + \frac{z_{rc} - z_{sat,i}}{r_i} z_{rc} + c(t_{sat,i} - t_{rc}) \tag{3}$$

$$r_i = \sqrt{(x_{rc} - x_{sat,i})^2 + (y_{rc} - y_{sat,i})^2 + (z_{rc} - z_{sat,i})^2} \tag{4}$$

where $x_{rc}, y_{rc}, z_{rc}, t_{rc}$ and $x_{sat,i}, y_{sat,i}, z_{sat,i}, t_{sat,i}$ are the positions/clock readings of the receiver and ith satellite respectively. Assuming vehicle and receiver co-location, this information can be expressed as a linear system G_{gps} and state vector X_{gps}:

$$G_{gps} = \begin{pmatrix} \frac{x-x_1}{r_1} & \frac{y-y_1}{r_1} & \frac{z-z_1}{r_1} & -1 \\ \frac{x-x_2}{r_2} & \frac{y-y_2}{r_2} & \frac{z-z_2}{r_2} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x-x_n}{r_n} & \frac{y-y_n}{r_n} & \frac{z-z_n}{r_n} & -1 \end{pmatrix} \qquad X_{gps} = \begin{pmatrix} x \\ y \\ z \\ c \cdot t \end{pmatrix} \tag{5}$$

with a best linear unbiased estimator (BEST), covariance $\Sigma_{gps} = \left(G_{gps}^T G_{gps}\right)^{-1}$ and dilutions of precision defined per [72]:

$$PDOP = \sqrt{\Sigma_{11,gps}^2 + \Sigma_{22,gps}^2 + \Sigma_{33,gps}^2}, \ TDOP = \sqrt{\Sigma_{44,gps}^2} \tag{6}$$

Accounting for visible satelllites, we define a motion planning GPS map-based uncertainty metric m_{gps} or cost c_{gps} as:

$$m_{gps}(x, y, z, t) = \frac{GDOP_{thresh} - \min(GDOP(x, y, z, t), GDOP_{cut})}{GDOP_{thresh} - 1} \tag{7}$$

$$c_{gps}(x, y, z) = 1 - m_{gps}(x, y, z) \tag{8}$$

where $GDOP_{thresh}$ is a worst-case cutoff value for safe flight.

B. Lidar Visibility

Lidar provides a local obstacle point cloud to assure safe navigation through complex spaces and support local-area mapping. In GPS-denied areas, lidar [73] can be used for inertial navigation by tracking mapped buildings and other landmarks. Lidar uses a laser's reflection time to estimate distances to objects. Lidar can be configured as a dome or cylindrical puck for local and longer-range sUAS applications.

The puck configuration modeled in this work uses b_{lidar} equiangular beams that revolve to scan at k_{lidar} equiangular positions capturing $n_{lidar} = b_{lidar} \cdot k_{lidar}$ points per revolution. Because $k_{lidar} >> 1$, n_{lidar} is impractical for metric normalization, we propose number of returned scan readings (where an obstacle is within lidar range) as a lidar metric. A scan reading is recorded if any beam of the jth scan, $j = 1, 2, \cdots, k_{lidar}$, intersects an obstacle in Ω_{obs} within range r_{lidar} from the sUAS:

$$scan(j) = \begin{cases} 1, & \text{if } \exists i \text{ s.t. } \overleftrightarrow{b_o b_{i,j}} \cap \Omega_{obs} \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $i \in \{1, 2, \ldots, b_{lidar}\}$, b_o is the origin point of all beams, and $b_{i,j}$ is the ith lidar beam point for the jth scan a distance r_{lidar} away with an elevation angle β_i.

A count of total scan returns $s_{lidar}(x, y, z, r_{lidar}) = \sum_1^{k_{lidar}} scan(j)$ is then compared to the total number of possible scan returns in the following lidar metric m_{lidar} or cost c_{lidar}:

$$m_{lidar}(x, y, z, r_{lidar}) = \frac{s_{lidar}(x, y, z, r_{lidar})}{k_{lidar}} \tag{10}$$

$$c_{lidar}(x, y, z, r_{lidar}) = 1 - m_{lidar}(x, y, z, r_{lidar}) \tag{11}$$

C. Obstacle Occupancy

Obstacle maps allow motion planners to define free C_{free} and obstacle C_{obs} configuration spaces. We define an obstacle occupancy metric m_{obs} to penalize flight paths with points that intersect obstacles such that:

$$m_{obs}(x, y, z) = c_{obs}(x, y, z) = \begin{cases} 0, & \text{if } (x, y, z) \cap C_{obs} = \emptyset \\ 1, & \text{otherwise} \end{cases} \tag{12}$$

D. Population Density

Flying low imposes a nontrivial risk to the overflown population. Population metric m_{pop} estimates expected normalized population density for each weekday. Population can be estimated from government census data [74] or dynamic sources such as mobile phone activity [15]. For Manhattan, turnstile and taxi data have also been used to estimate population [75]. Similar information is not available across multiple cities, so we propose extrapolating population estimates directly from census data.

Table 5   Dynamic population estimates in millions for Manhattan in 2010. [76].

	Work Week	Weekend
Daytime	3.94	2.90
Nighttime	2.05	2.05

Population estimates for Manhattan are presented in Table 5. A city's population varies throughout the day. Due to typical work hours, e.g., 9-to-5, population estimates in commercial areas are higher during the day. As people return home after work residential areas become densely populated during the evening. To estimate occupancy for each census map grid, we assume census data pop_{census} for nighttime population and modify daytime population by a scaling factor Γ determined based on area zoning (commercial Γ_{comm} or residential Γ_{resi}) such that:

$$pop_{mod}(x, y, \delta_d, \Gamma) = \begin{cases} \Gamma \cdot pop_{census}[\kappa(x, y)] & \text{if } \delta_d = day \\ pop_{census}[\kappa(x, y)] & \text{if } \delta_d = night \end{cases} \tag{13}$$

where δ_d denotes time of day and $\kappa(\cdot)$ is an indexing function relating census index to world coordinates.

The following population density metric and cost pair is then defined:

$$m_{pop}(x, y, \delta_d, \Gamma) = \frac{pop_{mod}(x, y, \delta_d, \Gamma)}{pop_{norm}(\mathcal{B}, \delta_d)} \qquad c_{pop}(x, y) = m_{pop} \tag{14}$$

where $pop_{norm}(\cdot)$ is maximum daytime or nighttime population density over bounding region $\mathcal{B}$.

E. Risk Proximity Metric

For this work risk is simply defined as proximity to nearby buildings or terrain with a threshold-based rectifier function. A building map is used to compute the distance to the closest obstacle surface, $d_{close}(x, y, z)$, for each map grid or point in space. For a specified distance threshold, d_{thresh}, a proximity risk is defined as:

$$m_{risk}(x, y, z) = \min\left(\frac{d_{close}}{d_{thresh}}, 1\right) \qquad c_{risk}(x, y, z) = 1 - m_{risk}(x, y, z) \tag{15}$$

10

F. Distance-based Path Metric

The expected distance traversed is given by:

$$m_{dist}(t_0, t_f) = \int_{t_0}^{t_f} |v(t)| dt \tag{16}$$

where t_0 and t_f are initial and final planned flight times and $v(\cdot)$ is velocity magnitude. This function can also be written as a summation of N segment lengths over planned flight path ζ:

$$m_{dist}(\zeta, N) = \sum_{i=1}^{N} \sqrt{(\zeta_{x,i} - \zeta_{x,i-1})^2 + (\zeta_{y,i} - \zeta_{y,i-1})^2 + (\zeta_{z,i} - \zeta_{z,i-1})^2} \tag{17}$$

where $\zeta_i = (\zeta_{x,i}, \zeta_{y,i}, \zeta_{z,i})$ is the ith point in path ζ.

V. Map Generation

Each Cartesian map of specified resolution defines a metric value for each spatial grid. For this investigation, metric maps cover an area $\mathcal{L}$ with a width 10km and height of 20km centered in Manhattan per Fig. 2. Maps with 2m, 5m, and

Fig. 2   Planning configuration space area $\mathcal{L}$ for Manhattan case studies.

10m resolution were generated. The 2m value coincides with current small UAS positioning and obstacle avoidance (trajectory tracking) accuracies. Height-dependent metrics were computed for UAS flight altitudes of 20m, 60m, 122m (current FAA maximum altitude for sUAS operations), and 600m AGL (above ground level), capturing low, medium, high, and ceiling-altitude flight. Note that cost map equivalents for each metric map can be computed by following the metric-to-cost conversions presented in the previous section.

A. Obstacle Maps

OpenStreetMap (OSM) [77] data was processed to extract a building-based obstacle map $\mathcal{H}_{obs}$ from ways and relations using attribute labels. OSM data was converted to a local UTM 18N (EPSG:32618) coordinate reference system (CRS). The Universal Transverse Mercator (UTM) coordinate projection allows metric calculations directly defining axes (easting, northing) in meters. Extracted polygons Ω_{obs} were rasterized at each map resolution. The height of the kth extracted polygon z_k located at grid point (x, y) was compared to UAS flight altitude z such that:

$$\mathcal{H}_{obs}(x, y, z) = \begin{cases} 1 & \text{if } z_k \geq z \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

B. GPS Maps

GPS metric maps $\mathcal{H}_{gps}$ describe expected GPS accuracy for the Manhattan urban canyon. For a given grid point (x, y, z) and date/time information $\mathcal{D}$, positions of overhead satellites are predicted using CelesTrak [78] and Skyfield [79]. Rays are cast to above-horizon satellites and checked for collisions against extruded buildings in Ω_{obs}. With less than four visible satellites ($N_{sats} < 4$), m_{gps} is set to zero; otherwise the GPS pseudorange and covariance matrices are used to calculate m_{gps}:

$$\mathcal{H}_{gps}(x, y, z) = \begin{cases} m_{gps}(x, y, z) & \text{if } N_{sats} \geq 4 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

C. Lidar Maps

Lidar metric maps $\mathcal{H}_{lidar}$ estimate metric m_{lidar}, the expected percentage of lidar range returns. It is assumed that the vehicle is equipped with b_{lidar} beams configured in a parallel configuration, i.e., the aircraft's z_{body} and the lidar's rotation axis are parallel. Hence, the ratio of scan returns per revolution at each grid point (x, y, z) is given by:

$$\mathcal{H}_{lidar}(x, y, z, r_{lidar}) = m_{lidar}(x, y, z, r_{lidar}) \tag{20}$$

D. Population Maps

Population metric maps $\mathcal{H}_{pop}$ are computed based on zoning and census data compiled into the normalized population metric m_{pop}. Census values are adjusted by Γ as described in Eq. 13 to adjust for commuting patterns between commercial Γ_{comm} and residential Γ_{resi} areas. Manhattan is divided into twelve districts starting at its southernmost neighborhood, i.e., the Financial District, to its northernmost neighborhood, i.e., Harlem, as shown in Fig. 3. The lower districts (1-6) are composed of businesses, government buildings, and tourist attractions. In contrast,

the upper districts (7-12) consist mostly of single and multi-family residences. Defined by NYC Department of City Planning [80], the twelve districts are labeled as shown on Table 6.

Table 6   Manhattan districts with their primary zoning types.

Number	Neighborhoods	Type
01	Financial District, Civic Center	Commercial
02	West Village, Greenwich Village, Soho	Commericial
03	Chinatown, East Village, Noho	Commericial
04	Chelsea, Clinton, Hell's Kitchen	Commericial
05	Union Square, Madison Square, Times Square	Commericial
06	Gramercy, Murray Hill, Turtle Bay	Commericial
07	Lincoln Square, Upper West Side, Manhattan Valley	Residential
08	Lenox Hill, Upper East Side, Yorkville	Residential
09	Morningside Heights, Hamilton Heights	Residential
10	Central Harlem	Residential
11	East Harlem	Residential
12	Inwood, Washington Heights	Residential

Population data for this study was derived from the 2010 United States Census [74]. The WGS84 CRS census block polygons represent the smallest geographic unit used by the US Census Bureau to estimate the number of residents in a block. Each census block entry includes a cumulative population count for that block and is assigned a district number 1-12 if the census block and district outline fully intersect. Any census block overlapping multiple outlines is assigned the district polygon's label with the largest intersection by area. Any census block within $\mathcal{L}$ but not in Manhattan, i.e., the Bronx or Queens, is given a district label of 13 and labeled as residential. All geospatial data is converted to the UTM 18N CRS for consistency. The population cost map is then defined as:

$$\mathcal{H}_{pop,\delta_d}(x,y) = \begin{cases} m_{pop}(x,y,\delta_d,\Gamma_{comm}) & \text{if } label[x,y] = \text{commercial} \\ m_{pop}(x,y,\delta_d,\Gamma_{resi}) & \text{if } label[x,y] = \text{residential} \end{cases} \tag{21}$$

E. Risk Maps

The final metric map set $\mathcal{H}_{risk}$ quantifies building obstacle risks in the urban canyon as a function of the proximity risk metric m_{risk} as shown below:

$$\mathcal{H}_{risk}(x,y,z) = m_{risk}(x,y,z) \tag{22}$$

(a) Figure of Manhattan district types. Commercial regions are red; residential regions are blue. Central Park with no permanent tenants is green.

(b) Manhattan census data reported in distinct polygonal regions.

Fig. 3 Manhattan community districts and census data blocks.

F. Composite Metric Maps

All the metric maps described above are collected into set $\mathcal{H}_{\delta_r, \delta_z}$ defined by:

$$
\mathcal{H}_{\delta_r, \delta_z} = \begin{pmatrix} \mathcal{H}_1 \\ \mathcal{H}_2 \\ \mathcal{H}_3 \\ \mathcal{H}_4 \\ \mathcal{H}_5 \end{pmatrix}_{\delta_r, \delta_z} = \begin{pmatrix} \mathcal{H}_{obs} \\ \mathcal{H}_{gps} \\ \mathcal{H}_{lidar} \\ \mathcal{H}_{pop, \delta_d} \\ \mathcal{H}_{risk} \end{pmatrix}_{\delta_r, \delta_z}
\tag{23}
$$

where δ_r is map resolution and δ_z is UAS flight altitude assumed constant for each planning instance in this work. A distinct $\mathcal{H}_{\delta_r, \delta_z}$ is stored for each (δ_r, δ_z) used in our case studies, and time of day δ_d as needed.

VI. Planning Algorithms

A. Point-to-Point: PTP

The simplest path a multicopter can take is direct, i.e., point-to-point (PTP). $\mathbf{\Lambda}_{\text{PTP}} = (\mathcal{L}, Q_S, Q_G, \mathcal{H}_{\delta_r, \delta_z}, \mathcal{W}, \delta_z, \delta_r)$ defines all relevant multicopter PTP flight planning parameters where $\mathcal{W}$ is a cost weighting vector defined below. PTP is a simple geometric construct that assumes no obstacles are present. A PTP solution must therefore be post-processed to check for obstacle collisions and evaluate path cost. The operating environment is described by the collection of

metric maps $\mathcal{H}_{\delta_r,\delta_z}$ defined above. Each map is rasterized with metric values generated for each grid in the map search space $\mathcal{L}$ at a given height and resolution pair (δ_z, δ_r). Using start and goal positions Q_S and Q_G, the path's grid-based map indices $(Q_{k,idx}, Q_{k,idy})$ given origin $(\mathcal{L}_{x,min}, \mathcal{L}_{y,min})$ are calculated as:

$$Q_{l,idx} = \left\lfloor \frac{Q'_{k,x} - \mathcal{L}_{x,min}}{\delta_r} \right\rfloor \quad Q'_{l,x}(\alpha_{k,x}) = Q_{S,x} + \alpha_{l,x} \tag{24}$$

$$Q_{l,idy} = \left\lfloor \frac{Q'_{k,y} - \mathcal{L}_{y,min}}{\delta_r} \right\rfloor \quad Q'_{l,y}(\alpha_{k,y}) = Q_{S,y} + \alpha_{l,y} \tag{25}$$

where $\alpha_{k,x}$ and $\alpha_{k,y}$ are component-wise steps from Q_S to Q_G for $l = 0, 1, \ldots \lceil \frac{\lambda}{\delta_r} \rceil$:

$$\alpha_{l,x} = \begin{cases} \lambda \cos(\theta) & \text{if } l = \lceil \frac{\lambda}{\delta_r} \rceil \\ l\delta_r \cos(\theta) & \text{otherwise} \end{cases} \quad \alpha_{k,x} = \begin{cases} \lambda \sin(\theta) & \text{if } l = \lceil \frac{\lambda}{\delta_r} \rceil \\ l\delta_r \sin(\theta) & \text{otherwise} \end{cases} \tag{26}$$

where $\theta = \text{atan2}(Q_{G,y} - Q_{S,y}, Q_{G,x} - Q_{S,x})$ and $\lambda = \sqrt{(Q_{G,y} - Q_{S,y})^2 + (Q_{G,y} - Q_{S,y})^2}$. Altitude δ_z is considered constant at one of the four designated layers for this study.

To test validity, a PTP solution path ζ is masked onto obstacle map $\mathcal{H}_{obs} \in \mathcal{H}_{\delta_r,\delta_z}$. If any masked index has a non-zero value, i.e., $\mathcal{H}_{obs}(Q_{l,idx}, Q_{l,idy}, Q_{l,idz}) > 0$, the path ζ is invalid; otherwise its cost is calculated. Total path cost $f(\zeta)$ is defined by:

$$f(\zeta) = \sum_{i \in l} c(Q_{i-1}, Q_i) \tag{27}$$

where $c(\cdot)$ is the transition cost between adjacent states. When using grid-based maps, the cost of moving between grids is described by the cost maps in $\mathcal{H}_{\delta_r,\delta_z}$. Given map indices (idx, idy, idz) costs can be computed, weighted with vector $\mathcal{W}$, and summed. The transition cost from Q_{i-1} to Q_i is then given by:

$$c(Q_{i-1}, Q_i) = w_0 d_{euc}(Q_{i-1}, Q_i) + \sum_{j=1}^{k} w_j \mathcal{H}_j(Q_{i,idx}, Q_{i,idy}, Q_{i,idz}) \tag{28}$$

where $d_{euc}(\cdot)$ is the Euclidean distance between states. Per Eq. 23, $k = 4$ cost metric maps for our planning case studies.

B. Graph-based Planning: A*

A* [49] is a discrete graph-based informed search algorithm popular for its completeness, optimality, and spatial efficiency. A* searches a graph $\mathcal{G}$ to find a sequence of edge transitions that optimally navigates $\mathcal{G}$ from a start node Q_S to a goal node Q_G. In motion planning, this sequence of edge transitions is equivalent to the desired path ζ. The A* motion planning problem is defined by:

- *Parameters*: $\Lambda_{A^*} = (\mathcal{L}, Q_S, Q_G, \mathcal{H}_{\delta_r,\delta_z}, \mathcal{W}, \delta_z, \delta_r, \delta_c)$

- *Search Graph*: $\mathcal{G} = (V, E)$
- *Total Cost Function*: $f(\cdot) = g(\cdot) + h(\cdot)$

where δ_c defines map cell adjacency for search graph $\mathcal{G}$, and (V, E) are the nodes and edges forming $\mathcal{G}$, respectively. $g(\cdot)$ is the cost function from the start node to the current search node, and $h(\cdot)$ is a heuristic function estimating cost from the current search node to the goal node. Graph vertices V are defined by discretizing $\mathcal{L}$ with resolution δ_r. In an obstacle-free environment a maximum of $\frac{\mathcal{L}_{dx}\mathcal{L}_{dy}}{\delta_r^2}$ map grids may be traversed. Configuration space C_{tot} is then:

$$C_{tot} = \left\{ Q_l \mid \forall i, j \wedge Q_{l,x} = i\frac{\mathcal{L}_{dx}}{\delta_r} \wedge Q_{l,y} = j\frac{\mathcal{L}_{dy}}{\delta_r} \right\} \tag{29}$$

where $i = 1, 2, \ldots, \mathcal{L}_{dx}/\delta_r$ and $j = 1, 2, \ldots, \mathcal{L}_{dy}/\delta_r$ such that $l = (i-1)\mathcal{L}_{dy}/\delta_r + j$. Nodes with obstacle conflicts given by $C_{obs} \subseteq C_{tot}$ are defined as:

$$C_{obs} = \left\{ Q \mid Q \in C_{tot} \wedge \mathcal{H}_{obs}(Q_{idx}, Q_{idy}, Q_{idz}) = 1 \right\} \tag{30}$$

All nodes with conflicts must be removed from the search-space; the obstacle-free configuration space C_{free} is then given by:

$$V = C_{tot} \setminus C_{obs} \equiv C_{free} \tag{31}$$

Graph edges can be created for all neighboring nodes as defined by connection logic δ_c. For an *8-connected* logic, any node v_0 has potential neighbors $v_1, v_2, \ldots, v_8$ as shown in Fig. 4a, with non-diagonal (odd) and diagonal (even) edges. Due to obstacles, not all neighbors might be reachable, as shown in Fig. 4b where we assume $v_2, v_5 \in C_{obs}$ for demonstration purposes.

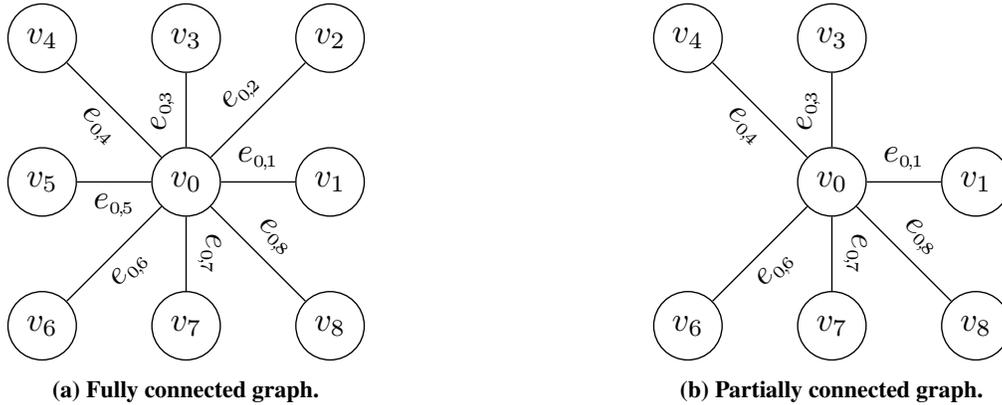(a) **Fully connected graph.** (b) **Partially connected graph.**

Fig. 4   Graph nodes, edges, and costs with 8-connected logic.

Accounting for obstacles, all feasible graph edges can be computed as follows:

$$E = \left\{ e_{m,n} = (Q_m, Q_n) \in \binom{V}{2} \right\} \tag{32}$$

where i and j serve as node identifiers or IDs.

Given graph $\mathcal{G} = (V, E)$, the start Q_S and goal Q_G nodes are matched to the closest nodes in $\mathcal{G}$ with labels assigned accordingly. An optimal path is then constructed using A* search on $\mathcal{G}$. To optimize path construction, A* uses the total cost $f(Q_n) = g(Q_n) + h(Q_n)$ where $g(Q_n)$ is the cumulative *cost-so-far* from Q_S to Q_n, and $h(Q_n)$ estimates *cost-to-go*. Similar to Eq. 27, $g(Q_n)$ is given by:

$$g(Q_n) = g(Q_m) + c(Q_m, Q_n) \tag{33}$$

where Q_m is the parent node of Q_n, and Eq. 28 calculates function $c(\cdot)$.

Built on the underlying optimalty of Dijkstra's algorithm [50], the A* heuristic function $h(\cdot)$ maintains optimality and improves search efficiency so long as:

- $h(\cdot)$ is admissible, i.e., it never overestimates the true *cost-to-go*.
- $h(\cdot)$ is consistent, i.e., for any successor configuration n, $h(m) \leq c(m, n) + h(n)$, where $c(\cdot)$ is the true cost to travel from m to n.

Under these conditions, we propose the following novel heuristic applicable to motion planning with multiple metric maps:

$$h_{plus}(Q_i) = w_0 \hat{d}(Q_i, Q_G) + \sum_{j=1}^{k} w_j \hat{s}_j(Q_i) \tag{34}$$

where $\hat{d}(\cdot)$ approximates the remaining distance to the goal and $\hat{s}_j(Q_i)$ conservatively estimates the cumulative map-based costs for the final path.

The distance function $\hat{d}(\cdot)$ is chosen to be admissible. For an 8-connected uniform grid, octile distance gives the minimum distance between any node pair. Octile distance d_{oct} extends Manhattan distance by allowing for diagonal transitions. The octile distance between two nodes m, n can be computed as:

$$\hat{d}(m, n) = d_{oct}(m, n) = \delta_r \left(|dx - dy| + \sqrt{2} \min(dx, dy) \right) \tag{35}$$

where $dx = |n_x - m_x|$, $dy = |n_y - m_y|$ represent the number of horizontal dx and vertical dy steps through the map of resolution δ_r required to reach node n from m.

Next, using the information encoded by each map in $\mathcal{H}_{\delta_r, \delta_z}$ we estimate the minimum map-based costs for any path to Q_G. From a current node Q_i an axis-aligned bounding box (AABB) $\mathcal{B}$ is constructed such that:

$$\mathcal{B} = \begin{pmatrix} \min(Q_{i,x}, Q_{G,x}) \\ \min(Q_{i,y}, Q_{G,y}) \\ \max(Q_{i,x}, Q_{G,x}) \\ \max(Q_{i,y}, Q_{G,y}) \end{pmatrix} = \begin{pmatrix} \mathcal{B}_{x,min} \\ \mathcal{B}_{y,min} \\ \mathcal{B}_{x,max} \\ \mathcal{B}_{y,max} \end{pmatrix} \tag{36}$$

with $n_r = \text{abs}(\mathcal{B}_{y,max} - \mathcal{B}_{y,min})/\delta_r$ rows and $n_c = \text{abs}(\mathcal{B}_{x,max} - \mathcal{B}_{x,min})/\delta_r$ columns.

The column and row index mappings between $\mathcal{B}$ and $\mathcal{L}$ are computed as follows:

$$\mathcal{B}_{l,idx} = \left\lfloor \frac{\mathcal{B}_{y,min} + i\delta_r - \mathcal{L}_{y,min}}{\delta_r} \right\rfloor \qquad \mathcal{B}_{l,idy} = \left\lfloor \frac{\mathcal{B}_{x,min} + j\delta_r - \mathcal{L}_{x,min}}{\delta_r} \right\rfloor \tag{37}$$

for $i = 0, 1, \dots, n_c$ and $j = 0, 1, \dots n_r$.

Using the index bounds for rows $(\mathcal{B}_{0,idy}, \mathcal{B}_{n_r,idy})$ and columns $(\mathcal{B}_{0,idx}, \mathcal{B}_{n_c,idx})$ the ith row or jth column used by the heuristic can be expressed as:

$$C_{k,j} = \{\mathcal{H}_k(\mathcal{B}_{0,idy}, \mathcal{B}_{j,idx}), \mathcal{H}_k(\mathcal{B}_{1,idy}, \mathcal{B}_{j,idx}), \dots, \mathcal{H}_k(\mathcal{B}_{n_r,idy}, \mathcal{B}_{j,idx})\} \tag{38}$$

$$\mathcal{R}_{k,i} = \{\mathcal{H}_k(\mathcal{B}_{i,idy}, \mathcal{B}_{0,idx}), \mathcal{H}_k(\mathcal{B}_{i,idy}, \mathcal{B}_{1,idx}), \dots, \mathcal{H}_k(\mathcal{B}_{i,idy}, \mathcal{B}_{n_c,idx})\} \tag{39}$$

for the kth cost map in $\mathcal{H}_{\delta_r,\delta_z}$, i.e., for $k \geq 1$.

The minimum cost for the kth map-based metric is computed as follows:

$$\hat{s}_k(n) = \max\left(\sum_{i=1}^{dx} \min(C_{k,i}), \sum_{i=1}^{dy} \min(\mathcal{R}_{k,i})\right) \tag{40}$$

By construction, this portion of the heuristic is consistent and admissible. Since both portions of the heuristic are admissible, the overall presented heuristic is admissible as well, guaranteeing A* solution optimality. To test this heuristic, two A* variants are studied in this paper. A^*_{dist} uses a traditional Euclidean distance-to-goal heuristic h_{dist} while A^*_{plus} applies the novel h_{plus} defined in Eq. 34.

C. Sampling-based Planning: BIT*

Batch Informed Trees (BIT*) [61] is a sampling-based search algorithm that improves scalability relative to classical graph-based techniques. Extending on previous work [81], BIT* utilizes an iterative search graph $\mathcal{G}$ informed by previous solutions. When a solution is found, BIT* reduces its search space C_{free}, prunes and reuses its search graph, generates a new set of samples in the new C_{free}, and restarts its search. BIT* terminates when a cost threshold has been met or all batches are complete.

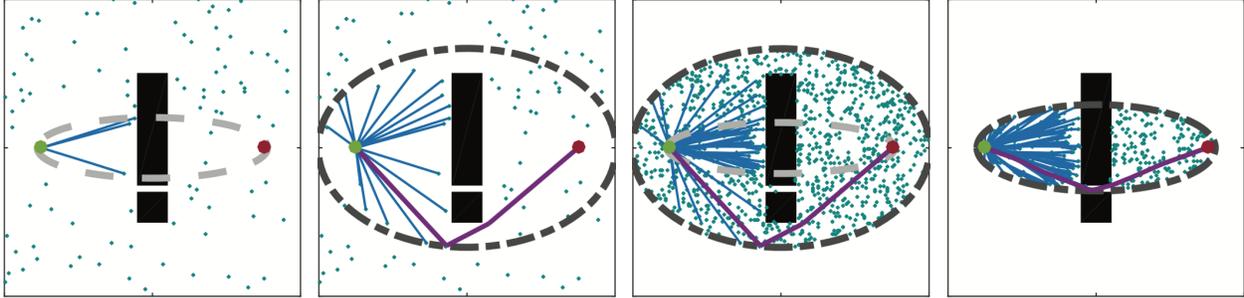For this investigation, the BIT* motion planning problem is defined by:

- *Parameters*: $\Lambda_{\text{BIT}^*} = (\mathcal{L}, Q_S, Q_G, \mathcal{H}_{\delta_r, \delta_z}, \mathcal{W}, \delta_z, \delta_r, \delta_b, \delta_s)$
- *Search Tree*: $\mathcal{T}_i = \text{BIT}^* (\mathcal{T}_{i-1}, \mathcal{H}_{obs}, \delta_s)$ for $i = 1, 2, \ldots, \delta_b$
- *Total Cost Function*: $f(\cdot) = g(\cdot) + h(\cdot)$

where $\text{BIT}^*(\cdot)$ returns a graph, and path if found, updated with δ_s samples per batch, for δ_b batches/iterations.

Similar to A*, BIT* uses a *cost-so-far* function $g(\cdot)$ and *cost-to-go* heuristic $h(\cdot)$ to search a series of increasingly dense implicit rapidly-exploring random graphs (RRGs) efficiently as illustrated in Fig. 5, adapted from [61]. When initializing the ith batch, the search for a solution expands outward from the minimum cost solution, adding feasible connections from $C_{free,i}$ to a growing tree $\mathcal{T}_i$ with nodes and edges (V_i, E_i). If a solution is found, the batch ends and the search space $C_{free,i+1}$ is redefined so new samples can only improve the current solution. The previous tree is pruned of any nodes and edges outside of $C_{free,i+1}$ such that:

$$V_{i+1} = V_i \cap C_{free,i+1} \qquad E_{i+1} = \{e_{m,n} \mid e_{m,n} \in E_i \wedge Q_m, Q_n \in V_{i+1}\} \qquad (41)$$

A new set of δ_s nodes is sampled in $C_{free,i+1}$, and the search restarts for the next batch. BIT* terminates when all δ_b batches are complete or the latest solution meets some cost-ending criteria, e.g., a percent change or total cost threshold.

(a) For each batch, the search expands out from the minimum solution. **(b) When a solution is found, the batch finishes and a new search space is defined.** **(c) A new batch of samples is added to a newly reduced search space and restarts.** **(d) The process repeats to find a better solution every batch.**

Fig. 5   BIT* batch process as adapted from [61].

During the first batch, $\mathcal{T}_1$ is initiated such that $V = \{Q_S\}$ and $E = \emptyset$. Nodes are added to the closest node in the current tree if a collision-free edge is feasible and they improve the best solution so far $\hat{\zeta}$. The costs of of adding a new node Q_n with an edge $e_{m,n}$ are computed using Eq. 33 for $g(Q_n)$ and Eq. 28 for $c(Q_m, Q_n)$. Similar to the A* variants, BIT^*_{dist} uses h_{dist} as its heuristic while BIT^*_{plus} applies the novel h_{plus} defined in Eq. 34.

VII. Manhattan Metric Map Results

Metric maps over Manhattan region $\mathcal{L}$ at three different resolutions (2m, 5m, and 10m) were generated for four small UAS AGL flight altitudes: 20m (low-altitude), 60m (medium-altitude), 122m (high-altitude), and 600m (ceiling-altitude). This altitude set covers sUAS flight paths that range from deep inside the New York City urban canyon (low-altitude) to above all buildings (ceiling-altitude). Fig. 6 shows GPS maps for low (20m), medium (60m), and high (122m) altitude flight. GPS metric scores are normalized between 0 and 1, where $m_{gps} = 1$ indicates the highest accuracy. As expected, GPS accuracy is highest in building-free areas, i.e., the Hudson River or Central Park, or residential areas with single-family homes, i.e., New Jersey. GPS accuracy decreases in low-altitude urban canyon regions with tall buildings.
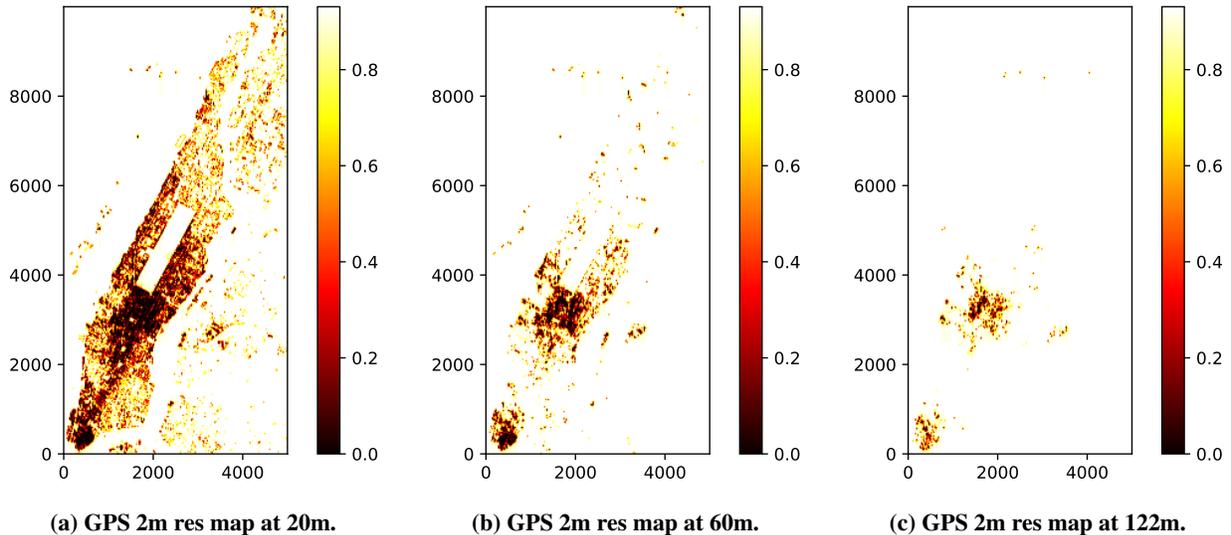

(a) GPS 2m res map at 20m. (b) GPS 2m res map at 60m. (c) GPS 2m res map at 122m.

Fig. 6 GPS metric maps for low, medium, and high-altitude urban flight.

For medium-altitude flight, the effects of urban canyon flight lessen. Upper Manhattan and Brooklyn (lower right) are now areas with high GPS accuracy. Similarly, high GPS accuracy areas now appear in Lower Manhattan but to a lesser extent. The Financial District (bottom left) and Midtown Manhattan (below Central Park) still include low GPS accuracy regions. This is to be expected as these areas are known for their tall buildings, e.g., One World Trade Center and Central Park Tower. The UAS primarily operates above the urban canyon at high and ceiling flight altitudes with near-perfect GPS accuracy.

Fig. 7 shows expected lidar performance for low-altitude and medium-altitude flight. In contrast to GPS, lidar performance is better at lower altitudes since the urban canyon offers in-range point cloud data and better visibility of its surroundings. In low-altitude flight, lidar performance is highest in the East Side, West Side, Midtown, and Downtown Manhattan areas densely packed with commercial and tourist high-rises. Weak lidar returns can be found in Uptown Manhattan, New Jersey, Brooklyn, and Queens, areas with mostly low-rise and residential buildings.

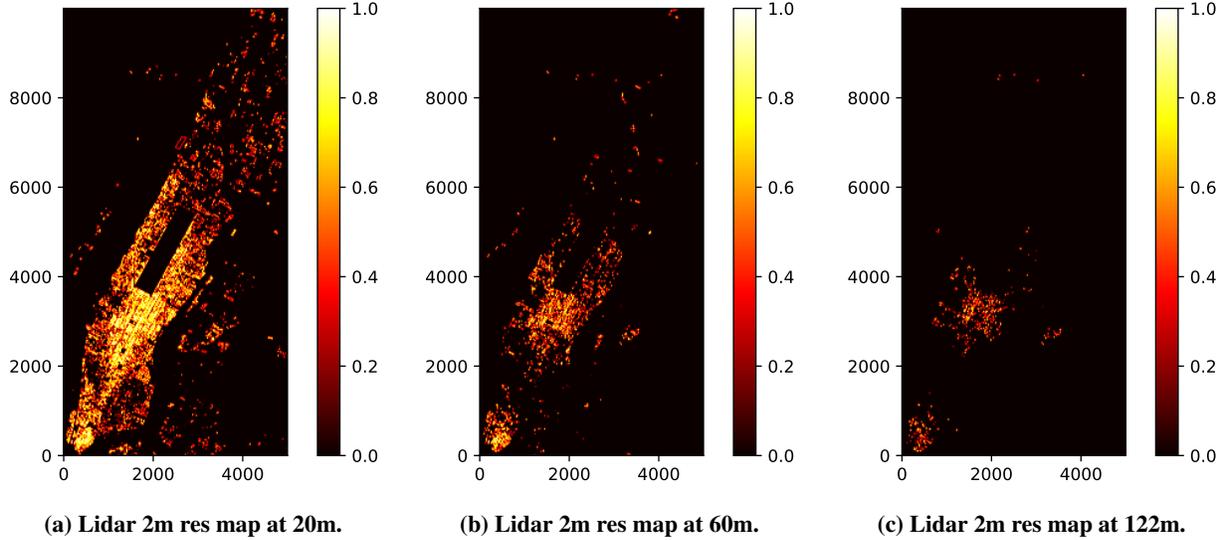Medium-altitude lidar analysis shows a significant drop in performance. Of the four predominant high m_{lidar}

(a) Lidar 2m res map at 20m.   **(b) Lidar 2m res map at 60m.**   **(c) Lidar 2m res map at 122m.**

Fig. 7   Lidar metric maps for low, medium, and high-altitude urban flight.

regions from the low-altitude analysis, only Midtown Manhattan remains. A pattern emerges at this altitude that suggests the potential for GPS to complement lidar, and vice-versa. Areas of low m_{gps} due to the urban canyon coexist with high m_{lidar} areas, and low m_{lidar} due to the absence of nearby obstacles results in high m_{gps} areas without satellite obstruction. This effect becomes more apparent at high-altitude flight and above.

Day and night population metric maps, shown in Fig. 8, are independent of flight altitude. The following daytime population scaling factors were used: $\Gamma_{comm} = 3.0$ and $\Gamma_{resi} = 0.5$. These values are biased toward a net population influx into Manhattan for the workday as show in Table 7. The population map results validate the expected residence-to-work and work-to-residence commuting patterns and constraints discussed in Sec. V.D.

Table 7   Work weekday and nighttime population estimates in millions.

	Residential	Commericial
Daytime	0.48	3.96
Nighttime	0.97	1.32

Proximity risk maps identify obstacle-free map grid points with decaying risk value over a distance d_{thresh} around buildings, the risk is one at the building, linearly decreasing to 0 at d_{thresh}. High proximity risk areas are mostly in the Manhattan borough, as shown in Fig. 9. For low altitude-flight, except for the Hudson River, New Jersey, and Central Park, a building can be found within 10m in most grids. Large portions of the Bronx, Queens, Brooklyn, and Uptown Manhattan become risk-free zones at medium-altitude flight. Only Downtown and Midtown Manhattan remain at high-altitude flight due to the congestion of tall buildings, as discussed earlier.
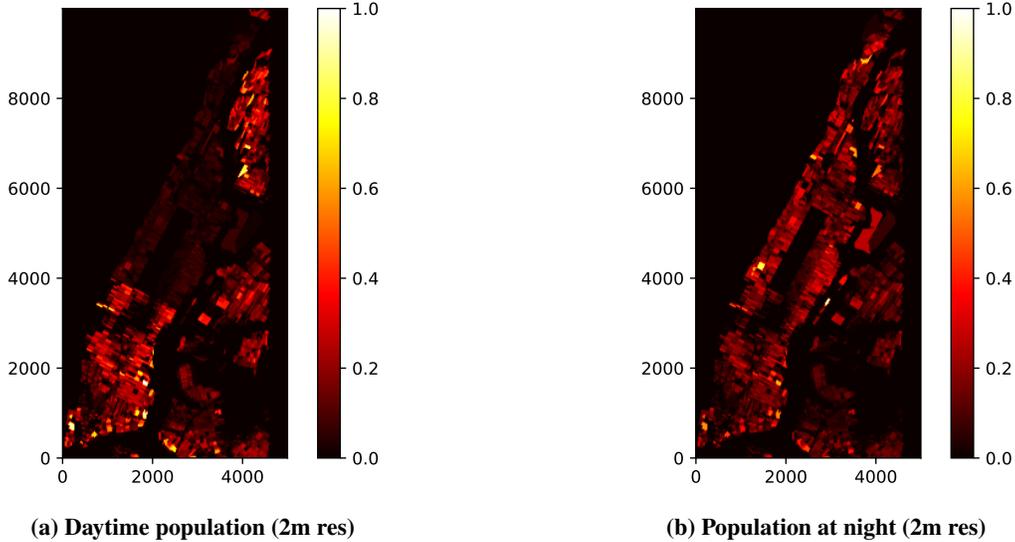
21

(a) Daytime population (2m res)

(b) Population at night (2m res)

Fig. 8   Population metric maps over Manhattan for day and night hours.

VIII. Monte Carlo Simulation Procedure

All map generation and planning simulations were performed using the Google Cloud: Compute Engine (CE). Maps and Monte Carlo planning simulations were generated using ten *n1-standard-16* virtual machines (VMs). Two geospatial datasets were used for all simulations: (1) OSM and (2) TIGER. OSM data was downloaded from PlanetOSM[†] as a 50+ GB PBF file. TIGER[‡] 2010 US Census data was downloaded directly from the US Census Bureau as a 180+ MB shapefile. The Geospatial Data Abstraction Library (GDAL) was used to uncompress and extract all Manhattan-specific data within $\mathcal{L}$. Start Q_S and goal Q_G configurations were sampled across $\mathcal{L}$ to capture all relevant subdomains, e.g., flying over water, suburban, and high rise building areas. Weighting vectors $\mathcal{W}$ were randomly generated for all problem instances. Each motion planning algorithm was implemented as discussed in Sec. VI in Cython, Python's optimized statically compiled variant. Cython takes advantage of Python's high-level, easily readable syntax while providing speeds comparable to C/C++ on execution. All planning instances were equally distributed among all VMs and ran against each planner.

IX. Path Planning Results

This section analyzes solution path properties from Monte Carlo simulations. Case studies are selected for each altitude $z^* \in \{20m, 60m, 122m, 600m\}$ AGL. Motion planning solutions generated within the allotted time (three minutes) are shown relative to the total unweighted cost map $\mathcal{H}_{total}$ referenced during planning. Total cost maps

---

[†]https://planet.openstreetmap.org/

[‡]https://www.census.gov/geographies/mapping-files.html

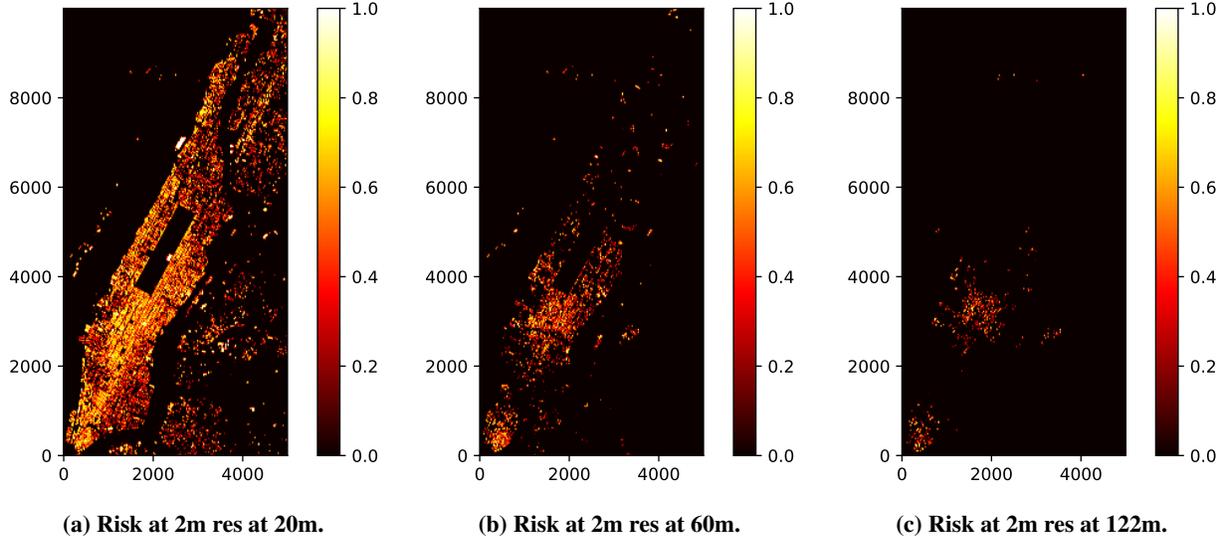(a) Risk at 2m res at 20m.　　　　(b) Risk at 2m res at 60m.　　　　(c) Risk at 2m res at 122m.

Fig. 9　Proximity risk metric maps for low-altitude and medium-altitude flight.

$\mathcal{H}_{total}(z^*)$ are defined by:

$$\mathcal{H}_{total}(z^*) = \mathcal{H}_{gps}(z^*) + \mathcal{H}_{lidar}(z^*) + \mathcal{H}_{pop}(z^*) + \mathcal{H}_{risk}(z^*) \tag{42}$$

and normalized using min-max normalization:

$$\mathcal{H}_{shift}(z^*) = \mathcal{H}_{total}(z^*) - \min(\mathcal{H}_{total}(z^*))J \tag{43}$$

$$\mathcal{H}_{norm}(z^*) = \frac{1}{\max(\mathcal{H}_{shift}(z^*))}\mathcal{H}_{shift}(z^*) \tag{44}$$

where J is a matrix of ones with the same dimensions as $\mathcal{H}_{total}$. To compare, we focus on daytime population for {20m, 60m} AGL flight and nighttime population for {122m, 600m} flight. Motion planners that found a solution are labeled on the top-left corner of each map.

For low-altitude flight (20m AGL) obstacle-related costs are prominent in $\mathcal{H}_{norm}$, where $\mathcal{H}_{norm} = 0$ is depicted in black with a gradient to white for $\mathcal{H}_{norm} = 1$ in Fig. 10. Manhattan, the Bronx, and portions of Queens/Brooklyn display high cost values attributed to tall buildings and urban canyon effects. At such a low altitude, a motion planner requires efficient obstacle-avoidance to find a feasible solution. As shown in Fig. 10a, for a long-range flight traversing through Manhattan only A^*_{dist} was able to find a solution. In contrast, for short-range flights over New Jersey, all planners were able to generate a feasible flight path as shown in Fig. 10b. Fig. 10c shows a mid-range flight with some obstacles present over parts of Queens and Manhattan. The modest number of obstacles allowed three out of the five motion planners to terminate but with different path traits. As described below, A^*_{dist} followed a grid-based path that is minimum distance only with respect to that grid, while the BIT* variants took another option that is more direct because

BIT* does not rely on the 5*m* resolution map grid apart from estimates of cost.
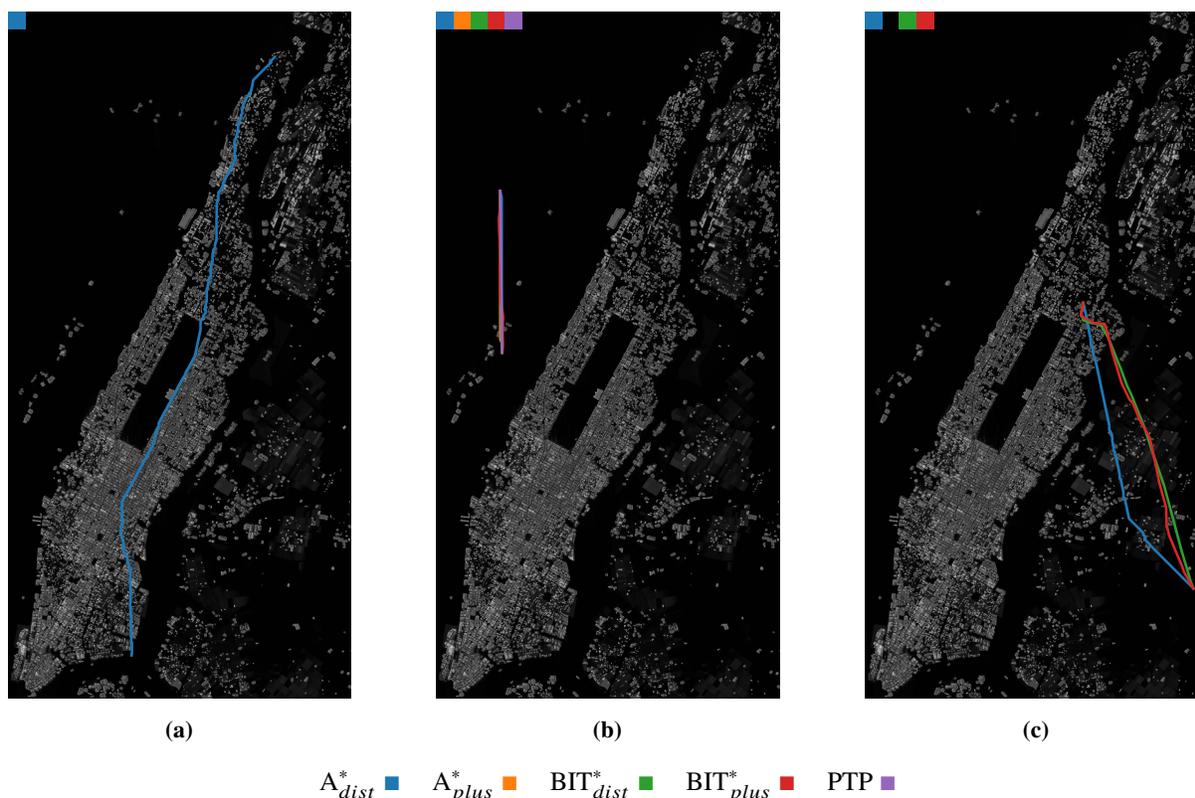

Fig. 10 Example solution paths at 20m AGL, 5m resolution maps in New York City.

For mid-altitude flight (60m AGL), similar path and $\mathcal{H}_{norm}$ characteristics are observed in the Fig. 11 example paths. At this height, obstacles are only present in the Financial District (lower left) and Midtown Manhattan. Population now plays a more significant role in low-rise areas, especially the neighboring boroughs. Fig. 11a depicts a path attempting to traverse Midtown Manhattan. Motion planners circumvented the dense group of tall buildings with BIT$^*_{dist}$ taking "shortcuts" to minimize distance while BIT$^*_{plus}$ navigates through lower population and risk areas. Fig. 11b investigates paths generated over the Hudson River. With no population or obstacle-related costs, all motion planners are capable of constructing feasible paths. BIT* variants and PTP take a direct approach from Q_S to Q_G. The A* variants follow eight-connected grids. With the 5*m* resolution case study map, each A* step is either 5*m* along a primary compass direction or 7.07*m* along a 45 degree diagonal. This grid-based routing process leads to longer thus higher cost paths compared with direct routes, e.g., a distance cost of 5625*m* for PTP versus 6092*m* for A$^*_{dist}$ in the example from Fig. 11b. This phenomenon is also observed in Fig. 11c.

For high-altitude flight (122m AGL), tall buildings only remain in highly concentrated areas of the Financial District and Midtown Manhattan. Fig. 12a and Fig. 12b illustrate the success of motion planners when flying in these areas for short and long-range flight. In the first case, paths are generated from New Jersey, across the Hudson, and into Midtown Manhattan. Given the long range and abundance of obstacles upon approach, only A$^*_{dist}$ and the BIT* variants
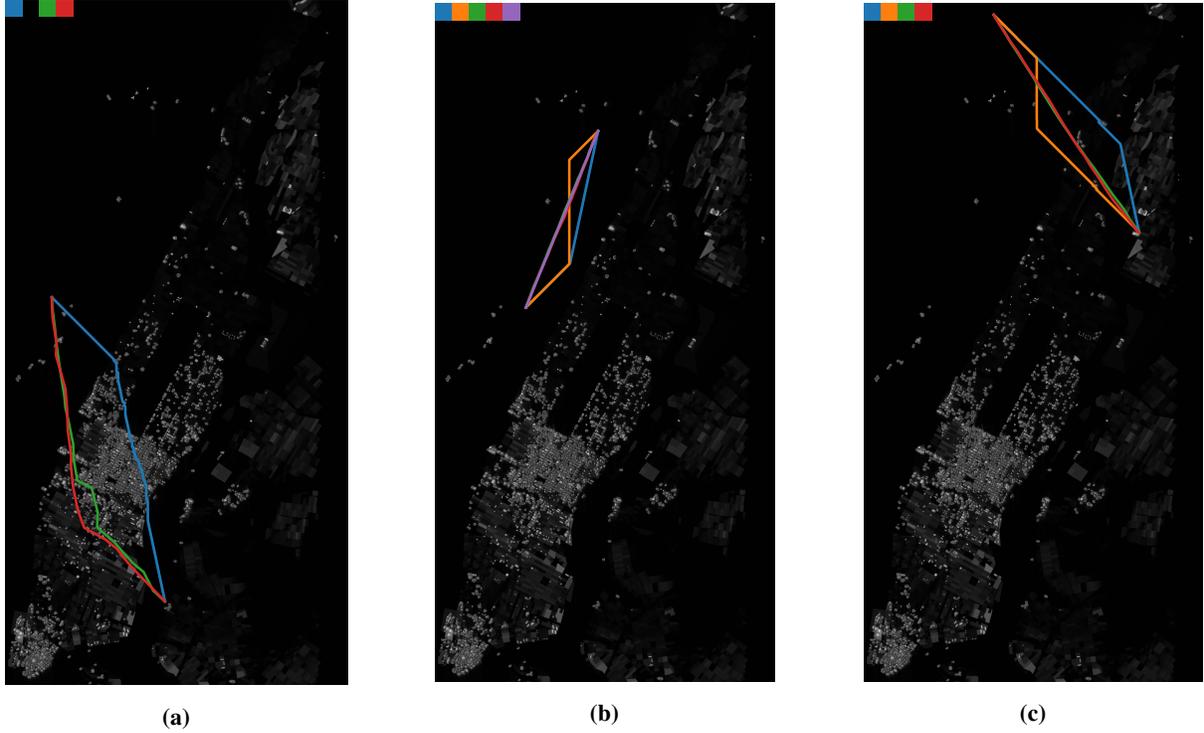
Fig. 11   Example solution paths at 60m AGL, 5m resolution maps in New York City.

successfully terminated. However, with a reduced distance between Q_S and Q_G, A^*_{plus} now terminates and takes a safer path than the rest. Furthermore, range can also be an issue for BIT^*_{plus}. As shown in Fig. 12c, BIT^*_{dist} and BIT^*_{plus} generate noticeably different paths. Given BIT^*_{plus} had to search more nodes to minimize non-distance costs, it had fewer batches, or iterations, to return its best-cost solution by the planning deadline.

Above all buildings at 600m AGL, only distance and population remain as nontrivial costs. As shown in Fig. 13a, lack of obstacles and short travel distance is ideal for all planners. However, this may not be the case as range increases per Figs. 13c and 13c. Along the Hudson River, distance is the only cost to optimize, making PTP the best motion planner in this example. However, upon entering Manhattan, BIT^*_{plus} becomes more suitable as it selects a route over lower population areas. The distance-population tradeoff demonstrates the benefits of geometric versus sampling-based planners. Collectively, these case studies illustrate the pros and cons of each planner thus motivate motion planning algorithm selection.

X. Conclusion

This paper has defined a set of map-based and path-based metrics for sUAS urban flight planning. Map-based metrics were investigated in detail with metric maps generated over Manhattan at three different resolutions for four sUAS AGL flight altitudes. Results demonstrate the complementary nature of GPS and lidar accuracy in an urban
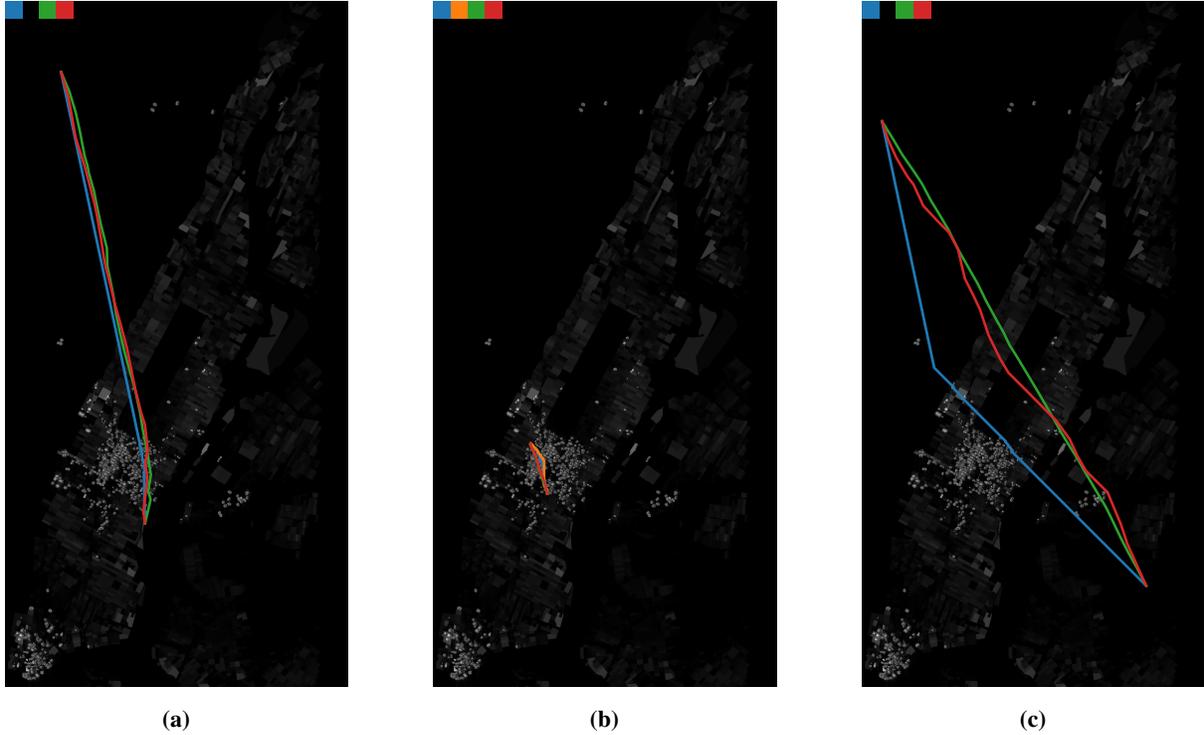
(a)   **(b)**   **(c)**

A$^*_{dist}$ ■   A$^*_{plus}$ ■   BIT$^*_{dist}$ ■   BIT$^*_{plus}$ ■   PTP ■

Fig. 12   Example solution paths at 122m AGL, 5m resolution maps in New York City.

canyon as a function of altitude. By generating these metric maps a priori, an sUAS can predict risk and sensor data quality before a flight, i.e., GPS will provide valid position data if $m_{gps} > m_{lidar}$; lidar will offer better data otherwise.

Population metric maps support residence-to-work and work-to-residence commuting patterns using as simplified as work-week daytime and nighttime models. In the future, this model should be extended to weekends with a time-based population function offering more resolution over 24-hour population patterns. When deep in the urban canyon, proximity-based risk is high, but it quickly decreases at higher altitudes due to fewer obstacles. For path planning, if risk is the primary cost, data indicate that flying to a higher altitude is preferable. Additional research is needed to incorporate risk metrics for urban flight planning, such as system, actuator, sensor, and weather-related risks, to extend current fixed-altitude maps to full 3D cost maps to support full 3D flight planning.

References

[1] LaValle, S. M., *Planning Algorithms*, Cambridge University Press, Cambridge ; New York, 2006. OCLC: ocm65301992.

[2] Colas, F., Mahesh, S., Pomerleau, F., Ming Liu, and Siegwart, R., "3D path planning and execution for search and rescue ground robots," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Tokyo, 2013, pp. 722–727. https://doi.org/10.1109/IROS.2013.6696431, URL http://ieeexplore.ieee.org/document/6696431/.

[3] Berger, J., and Lo, N., "An innovative multi-agent search-and-rescue path planning approach," *Computers & Operations*

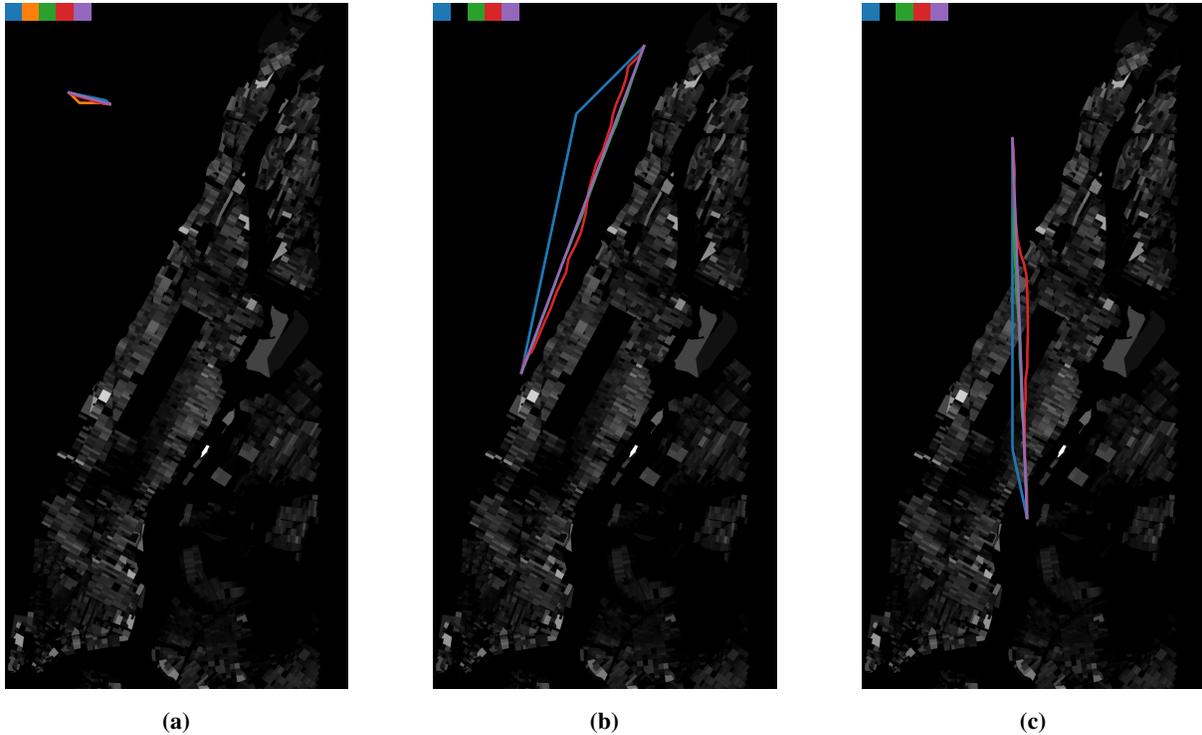**(a)**   **(b)**   **(c)**

A^*_{dist} ■ A^*_{plus} ■ BIT^*_{dist} ■ BIT^*_{plus} ■   PTP ■

Fig. 13   Example solution paths at 600m AGL, 5m resolution maps in New York City.

Research, Vol. 53, 2015, pp. 24–31. https://doi.org/10.1016/j.cor.2014.06.016, URL https://linkinghub.elsevier.com/retrieve/pii/S0305054814001749.

[4] Obermeyer, K., "Path Planning for a UAV Performing Reconnaissance of Static Ground Targets in Terrain," *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, Chicago, Illinois, 2009. https://doi.org/10.2514/6.2009-5888, URL http://arc.aiaa.org/doi/10.2514/6.2009-5888.

[5] Obermeyer, K. J., Oberlin, P., and Darbha, S., "Sampling-Based Path Planning for a Visual Reconnaissance Unmanned Air Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 619–631. https://doi.org/10.2514/1.48949, URL https://arc.aiaa.org/doi/10.2514/1.48949.

[6] Reif, J., and Sharir, M., "Motion planning in the presence of moving obstacles," *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, IEEE, Portland, OR, USA, 1985, pp. 144–154. https://doi.org/10.1109/SFCS.1985.36, URL http://ieeexplore.ieee.org/document/4568138/.

[7] Radmanesh, M., Kumar, M., Guentert, P. H., and Sarim, M., "Overview of Path-Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study," *Unmanned Systems*, Vol. 06, No. 02, 2018, pp. 95–118. https://doi.org/10.1142/S2301385018400022, URL https://www.worldscientific.com/doi/abs/10.1142/S2301385018400022.

[8] Rathbun, D., Kragelund, S., Pongpunwattana, A., and Capozzi, B., "An evolution based path planning algorithm for autonomous

motion of a UAV through uncertain environments," *Proceedings. The 21st Digital Avionics Systems Conference*, Vol. 2, IEEE, Irvine, CA, USA, 2002, pp. 8D2–1–8D2–12. https://doi.org/10.1109/DASC.2002.1052946, URL http://ieeexplore.ieee.org/document/1052946/.

[9] Butenko, S., Murphey, R., and Pardalos, P. M. (eds.), *Cooperative Control: Models, Applications and Algorithms*, Cooperative Systems, Vol. 1, Springer US, Boston, MA, 2003. https://doi.org/10.1007/978-1-4757-3758-5, URL http://link.springer.com/10.1007/978-1-4757-3758-5.

[10] Dadkhah, N., and Mettler, B., "Survey of Motion Planning Literature in the Presence of Uncertainty: Considerations for UAV Guidance," *Journal of Intelligent & Robotic Systems*, Vol. 65, No. 1-4, 2012, pp. 233–246. https://doi.org/10.1007/s10846-011-9642-9, URL http://link.springer.com/10.1007/s10846-011-9642-9.

[11] Peinecke, N., and Kuenz, A., "Deconflicting the Urban Drone Airspace," *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, IEEE, St. Petersburg, FL, 2017, pp. 1–6. https://doi.org/10.1109/DASC.2017.8102048, URL http://ieeexplore.ieee.org/document/8102048/.

[12] Ochoa, C. A., and Atkins, E. M., "Fail-Safe Navigation for Autonomous Urban Multicopter Flight," *AIAA Information Systems-AIAA Infotech @ Aerospace*, American Institute of Aeronautics and Astronautics, Grapevine, Texas, 2017. https://doi.org/10.2514/6.2017-0222, URL http://arc.aiaa.org/doi/10.2514/6.2017-0222.

[13] Moll, M., Sucan, I. A., and Kavraki, L. E., "Benchmarking Motion Planning Algorithms: An Extensible Infrastructure for Analysis and Visualization," *IEEE Robotics & Automation Magazine*, Vol. 22, No. 3, 2015, pp. 96–102. https://doi.org/10.1109/MRA.2015.2448276, URL http://ieeexplore.ieee.org/document/7214252/.

[14] Shan, T., and Englot, B., "Sampling-based Minimum Risk path planning in multiobjective configuration spaces," *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, Osaka, 2015, pp. 814–821. https://doi.org/10.1109/CDC.2015.7402330, URL http://ieeexplore.ieee.org/document/7402330/.

[15] 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, 2017, pp. 259–278. https://doi.org/10.2514/1.I010513, URL https://arc.aiaa.org/doi/10.2514/1.I010513.

[16] Castagno, J., Ochoa, C., and Atkins, E., "Comprehensive Risk-based Planning for Small Unmanned Aircraft System Rooftop Landing," *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, Dallas, TX, 2018, pp. 1031–1040. https://doi.org/10.1109/ICUAS.2018.8453483, URL https://ieeexplore.ieee.org/document/8453483/.

[17] Ippolito, C. A., "Dynamic ground risk mitigation for autonomous small uas in urban environments," *AIAA Scitech 2019 Forum*, 2019, p. 0961.

[18] Rudnick-Cohen, E., Herrmann, J. W., and Azarm, S., "Modeling Unmanned Aerial System (UAS) Risks via Monte Carlo Simulation," *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2019, pp. 1296–1305.

[19] Roberts, M., Howe, A., and Flom, O., "Learned models of performance for many planners," *In ICAPS 2007, workshop AI planning and learning*, 2007.

[20] Roberts, M., and Howe, A., "Learning from planner performance," *Artificial Intelligence*, Vol. 173, No. 5-6, 2009, pp. 536–561. https://doi.org/10.1016/j.artint.2008.11.009, URL https://linkinghub.elsevier.com/retrieve/pii/S0004370208001896.

[21] Saxena, A., Celaya, J., Saha, B., Saha, S., and Goebel, K., "Evaluating algorithm performance metrics tailored for prognostics," *2009 IEEE Aerospace conference*, IEEE, Big Sky, MT, USA, 2009, pp. 1–13. https://doi.org/10.1109/AERO.2009.4839666, URL http://ieeexplore.ieee.org/document/4839666/.

[22] Jain, A., and Zongker, D., "Feature selection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, 1997, pp. 153–158. https://doi.org/10.1109/34.574797, URL http://ieeexplore.ieee.org/document/574797/.

[23] Russell, S. J., Norvig, P., and Davis, E., *Artificial Intelligence: A Modern Approach*, 3rd ed., Prentice Hall Series in Artificial Intelligence, Prentice Hall, Upper Saddle River, 2010.

[24] Lunenburg, J., Coenen, S., Naus, G., van de Molengraft, M., and Steinbuch, M., "Motion Planning for Mobile Robots: A Method for the Selection of a Combination of Motion-Planning Algorithms," *IEEE Robotics & Automation Magazine*, Vol. 23, No. 4, 2016, pp. 107–117. https://doi.org/10.1109/MRA.2015.2510798, URL http://ieeexplore.ieee.org/document/7493595/.

[25] Gligor, V., and Shattuck, S., "On Deadlock Detection in Distributed Systems," *IEEE Transactions on Software Engineering*, Vol. SE-6, No. 5, 1980, pp. 435–440. https://doi.org/10.1109/TSE.1980.230491, URL http://ieeexplore.ieee.org/document/1702759/.

[26] Singhal, M., "Deadlock detection in distributed systems," *Computer*, Vol. 22, No. 11, 1989, pp. 37–48. https://doi.org/10.1109/2.43525, URL http://ieeexplore.ieee.org/document/43525/.

[27] Dadvar, P., and Skadron, K., "Potential thermal security risks," *Semiconductor Thermal Measurement and Management IEEE Twenty First Annual IEEE Symposium, 2005.*, IEEE, San Jose, CA, USA, 2005, pp. 229–234. https://doi.org/10.1109/STHERM.2005.1412184, URL http://ieeexplore.ieee.org/document/1412184/.

[28] Mohla, D., McClung, L., and Rafferty, N., "Electrical safety by design," *Industry Applications Society 46th Annual Petroleum and Chemical Technical Conference (Cat.No. 99CH37000)*, IEEE, San Diego, CA, USA, 1999, pp. 363–369. https://doi.org/10.1109/PCICON.1999.806455, URL http://ieeexplore.ieee.org/document/806455/.

[29] Boehm, B., "Software risk management: principles and practices," *IEEE Software*, Vol. 8, No. 1, 1991, pp. 32–41. https://doi.org/10.1109/52.62930, URL http://ieeexplore.ieee.org/document/62930/.

[30] Bonnett, A., "Root cause AC motor failure analysis with a focus on shaft failures," *IEEE Transactions on Industry Applications*, Vol. 36, No. 5, 2000, pp. 1435–1448. https://doi.org/10.1109/28.871294, URL http://ieeexplore.ieee.org/document/871294/.

[31] Richardeau, F., Baudesson, P., and Meynard, T., "Failures-tolerance and remedial strategies of a PWM multicell inverter," *IEEE Transactions on Power Electronics*, Vol. 17, No. 6, 2002, pp. 905–912. https://doi.org/10.1109/TPEL.2002.805588, URL https://ieeexplore.ieee.org/document/1158980/.

[32] Graves, J. C., Turcio, W. H. L., Alvarez, J., and Yoneyama, T., "Spectral Signatures of Pneumatic Actuator Failures: Closed-Loop Approach," *IEEE/ASME Transactions on Mechatronics*, Vol. 23, No. 5, 2018, pp. 2218–2228. https://doi.org/10.1109/TMECH.2018.2863179, URL https://ieeexplore.ieee.org/document/8424906/.

[33] Jaguemont, J., Boulon, L., Dube, Y., and Martel, F., "Thermal Management of a Hybrid Electric Vehicle in Cold Weather," *IEEE Transactions on Energy Conversion*, Vol. 31, No. 3, 2016, pp. 1110–1120. https://doi.org/10.1109/TEC.2016.2553700, URL http://ieeexplore.ieee.org/document/7452369/.

[34] Watkins, S., Mohamed, A., and Ol, M. V., "Gusts Encountered by MAVs in Close Proximity to Buildings," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, California, 2019. https://doi.org/10.2514/6.2019-0900, URL https://arc.aiaa.org/doi/10.2514/6.2019-0900.

[35] Ancel, E., Capristan, F. M., Foster, J. V., and Condotta, R. C., "Real-time Risk Assessment Framework for Unmanned Aircraft System (UAS) Traffic Management (UTM)," *17th AIAA Aviation Technology, Integration, and Operations Conference*, American Institute of Aeronautics and Astronautics, Denver, Colorado, 2017. https://doi.org/10.2514/6.2017-3273, URL https://arc.aiaa.org/doi/10.2514/6.2017-3273.

[36] Puranik, T., Harrison, E., Chakraborty, I., and Mavris, D., "Aircraft Performance Model Calibration and Validation for General Aviation Safety Analysis," *Journal of Aircraft*, 2020, pp. 1–11. https://doi.org/10.2514/1.C035458, URL https://arc.aiaa.org/doi/10.2514/1.C035458.

[37] McClamroch, N. H., *Steady aircraft flight and performance*, Princeton University Press, Princeton, N.J, 2011. OCLC: ocn611551579.

[38] Balachandran, S., and Atkins, E. M., "Flight Safety Assessment and Management to Prevent Loss of Control Due to In-Flight Icing," *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, San Diego, California, USA, 2016. https://doi.org/10.2514/6.2016-0094, URL http://arc.aiaa.org/doi/10.2514/6.2016-0094.

[39] Ten Harmsel, A. J., Olson, I. J., and Atkins, E. M., "Emergency Flight Planning for an Energy-Constrained Multicopter," *Journal of Intelligent & Robotic Systems*, Vol. 85, No. 1, 2017, pp. 145–165. https://doi.org/10.1007/s10846-016-0370-z, URL http://link.springer.com/10.1007/s10846-016-0370-z.

[40] Mardani, A., Chiaberge, M., and Giaccone, P., "Communication-Aware UAV Path Planning," *IEEE Access*, Vol. 7, 2019, pp. 52609–52621. https://doi.org/10.1109/ACCESS.2019.2911018, URL https://ieeexplore.ieee.org/document/8691430/.

[41] Bopardikar, S. D., Englot, B., and Speranzon, A., "Multi-objective path planning in GPS denied environments under localization constraints," *2014 American Control Conference*, IEEE, Portland, OR, USA, 2014, pp. 1872–1879. https://doi.org/10.1109/ACC.2014.6858731, URL http://ieeexplore.ieee.org/document/6858731/.

[42] Mitchell, I., and Sastry, S., "Continuous path planning with multiple constraints," *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, IEEE, Maui, HI, USA, 2003, pp. 5502–5507. https://doi.org/10.1109/CDC.2003.1272513, URL http://ieeexplore.ieee.org/document/1272513/.

[43] Shashi Mittal, and Kalyanmoy Deb, "Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms," *2007 IEEE Congress on Evolutionary Computation*, IEEE, Singapore, 2007, pp. 3195–3202. https://doi.org/10.1109/CEC.2007.4424880, URL http://ieeexplore.ieee.org/document/4424880/.

[44] Guigue, A., Ahmadi, M., Langlois, R., and Hayes, M. J., "Pareto Optimality and Multiobjective Trajectory Planning for a 7-DOF Redundant Manipulator," *IEEE Transactions on Robotics*, Vol. 26, No. 6, 2010, pp. 1094–1099. https://doi.org/10.1109/TRO.2010.2068650, URL http://ieeexplore.ieee.org/document/5582309/.

[45] Nilsson, N., "A Mobius Automation: an Application of Artificial Intelligence Techniques," *IJCAI'69: Proceedings of the 1st international joint conference on Artificial intelligence*, 1969, pp. 509–520.

[46] Lozano-Pérez, T., and Wesley, M. A., "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, Vol. 22, No. 10, 1979, pp. 560–570. https://doi.org/10.1145/359156.359164, URL http://portal.acm.org/citation.cfm?doid=359156.359164.

[47] Dubins, L. E., "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, Vol. 79, No. 3, 1957, p. 497. https://doi.org/10.2307/2372560, URL https://www.jstor.org/stable/2372560?origin=crossref.

[48] Reeds, J., and Shepp, L., "Optimal Paths for a Car that goes both Forwards and Backwards," *Pacific Journal of Mathematics*, Vol. 145, No. 2, 1990, pp. 367–393. https://doi.org/10.2140/pjm.1990.145.367, URL http://msp.org/pjm/1990/145-2/p06.xhtml.

[49] Hart, P., Nilsson, N., and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp. 100–107. https://doi.org/10.1109/TSSC.1968.300136, URL http://ieeexplore.ieee.org/document/4082128/.

[50] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol. 1, No. 1, 1959, pp. 269–271. https://doi.org/10.1007/BF01386390, URL http://link.springer.com/10.1007/BF01386390.

[51] Koenig, S., Likhachev, M., and Furcy, D., "Lifelong Planning A*," *Artificial Intelligence*, Vol. 155, No. 1-2, 2004, pp. 93–146. https://doi.org/10.1016/j.artint.2003.12.001, URL https://linkinghub.elsevier.com/retrieve/pii/S000437020300225X.

[52] Likhachev, M., Gordon, G., and Thrun, S., "ARA*: Formal analysis," Tech. rep., 2003.

[53] Koenig, S., and Likhachev, M., "Fast Replanning for Navigation in Unknown Terrain," *IEEE Transactions on Robotics*, Vol. 21, No. 3, 2005, pp. 354–363. https://doi.org/10.1109/TRO.2004.838026, URL http://ieeexplore.ieee.org/document/1435479/.

[54] Ferguson, D., and Stentz, A., "Field D*: An Interpolation-Based Path Planner and Replanner," *Robotics Research*, Vol. 28, edited by S. Thrun, R. Brooks, and H. Durrant-Whyte, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 239–253. https://doi.org/10.1007/978-3-540-48113-3_22, URL http://link.springer.com/10.1007/978-3-540-48113-3_22.

[55] Daniel, K., Nash, A., Koenig, S., and Felner, A., "Theta*: Any-Angle Path Planning on Grids," *Journal of Artificial Intelligence Research*, Vol. 39, 2010, pp. 533–579. https://doi.org/10.1613/jair.2994, URL https://jair.org/index.php/jair/article/view/10676.

[56] Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M., "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, 1996, pp. 566–580. https://doi.org/10.1109/70.508439, URL http://ieeexplore.ieee.org/document/508439/.

[57] LaValle, S. M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning," 1998.

[58] Kuffner, J., and LaValle, S., "RRT-connect: An efficient approach to single-query path planning," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Vol. 2, IEEE, San Francisco, CA, USA, 2000, pp. 995–1001. https://doi.org/10.1109/ROBOT.2000.844730, URL http://ieeexplore.ieee.org/document/844730/.

[59] Karaman, S., and Frazzoli, E., "Sampling-Based Algorithms for Optimal Motion Planning," *The International Journal of Robotics Research*, Vol. 30, No. 7, 2011, pp. 846–894. https://doi.org/10.1177/0278364911406761, URL http://journals.sagepub.com/doi/10.1177/0278364911406761.

[60] Janson, L., Schmerling, E., Clark, A., and Pavone, M., "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International Journal of Robotics Research*, Vol. 34, No. 7, 2015, pp. 883–921. https://doi.org/10.1177/0278364915577958, URL http://journals.sagepub.com/doi/10.1177/0278364915577958.

[61] Gammell, J. D., Srinivasa, S. S., and Barfoot, T. D., "Batch Informed Trees (BIT*): Sampling-Based Optimal Planning via the Heuristically Guided Search of Implicit Random Geometric Graphs," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Seattle, WA, USA, 2015, pp. 3067–3074. https://doi.org/10.1109/ICRA.2015.7139620, URL http://ieeexplore.ieee.org/document/7139620/.

[62] Kiguradze, I. T., "Boundary-value problems for systems of ordinary differential equations," *Journal of Soviet Mathematics*, Vol. 43, No. 2, 1988, pp. 2259–2339. https://doi.org/10.1007/BF01100360, URL http://link.springer.com/10.1007/BF01100360.

[63] Barraquand, J., Langlois, B., and Latombe, J.-C., "Numerical Potential Field Techniques for Robot Path Planning," *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, IEEE, Pisa, Italy, 1991, pp. 1012–1017 vol.2. https://doi.org/10.1109/ICAR.1991.240539, URL http://ieeexplore.ieee.org/document/240539/.

[64] Ge, S., and Cui, Y., "Dynamic Motion Planning for Mobile Robots Using Potential Field Method," *Autonomous Robots*, Vol. 13, No. 3, 2002, pp. 207–222. https://doi.org/10.1023/A:1020564024509, URL http://link.springer.com/10.1023/A:1020564024509.

[65] Spindler, K., "Motion planning via optimal control theory," *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, IEEE, Anchorage, AK, USA, 2002, pp. 1972–1977 vol.3. https://doi.org/10.1109/ACC.2002.1023924, URL http://ieeexplore.ieee.org/document/1023924/.

[66] Wang, Y., and Boyd, S., "Fast Model Predictive Control Using Online Optimization," *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 2, 2010, pp. 267–278. https://doi.org/10.1109/TCST.2009.2017934, URL http://ieeexplore.ieee.org/document/5153127/.

[67] Howard, T., Pivtoraiko, M., Knepper, R. A., and Kelly, A., "Model-predictive motion planning: Several key developments for autonomous mobile robots," *IEEE Robotics Automation Magazine*, Vol. 21, No. 1, 2014, pp. 64–73.

[68] Liu, C., Lee, S., Varnhagen, S., and Tseng, H. E., "Path Planning for Autonomous Vehicles using Model Predictive Control," *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Los Angeles, CA, USA, 2017, pp. 174–179. https://doi.org/10.1109/IVS.2017.7995716, URL http://ieeexplore.ieee.org/document/7995716/.

[69] Enge, P. K., "The Global Positioning System: Signals, measurements, and performance," *International Journal of Wireless Information Networks*, Vol. 1, No. 2, 1994, pp. 83–105. https://doi.org/10.1007/BF02106512, URL http://link.springer.com/10.1007/BF02106512.

[70] Santerre, R., "Impact of GPS satellite sky distribution," *Manuscripta Geodaetica*, Vol. 16, 1991.

[71] Azami, H., Mosavi, M.-R., and Sanei, S., "Classification of GPS Satellites Using Improved Back Propagation Training Algorithms," *Wireless Personal Communications*, Vol. 71, No. 2, 2013, pp. 789–803. https://doi.org/10.1007/s11277-012-0844-7, URL http://link.springer.com/10.1007/s11277-012-0844-7.

[72] Langley, R. B., and others, "Dilution of precision," *GPS world*, Vol. 10, No. 5, 1999, pp. 52–59.

[73] Rufa, J. R., and 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. https://doi.org/10.2514/1.I010280, URL http://arc.aiaa.org/doi/10.2514/1.I010280.

[74] US Census Bureau, "TIGER 2010 Census Block State-based Shapefile with Housing and Population Data," , 2010. URL https://catalog.data.gov/dataset/tiger-line-shapefile-2010-2010-state-new-york-2010-census-block-state-based-shapefile-with-hous.

[75] Fung, J., "Manhattan Population Explorer," , May 2019. URL https://github.com/citrusvanilla/manhattanpopulationexplorer.

[76] Moss, M. L., and Qing, C., "The Dynamic Population of Manhattan," 2012.

[77] Haklay, M., and Weber, P., "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, Vol. 7, No. 4, 2008, pp. 12–18. https://doi.org/10.1109/MPRV.2008.80, URL http://ieeexplore.ieee.org/document/4653466/.

[78] Kelso, T. S., "CelesTrak," , 1985. URL https://celestrak.com/.

[79] Rhodes, B., "Skyfield: Generate high precision research-grade positions for stars, planets, moons, and Earth satellites," , 2020. URL https://github.com/skyfielders/python-skyfield.

[80] NYC Department of City Planning, "Community Districts," , Jan. 2013. URL https://data.cityofnewyork.us/City-Government/Community-Districts/yfnk-k7r4.

[81] Gammell, J. D., Srinivasa, S. S., and Barfoot, T. D., "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Chicago, IL, 2014, pp. 2997–3004. https://doi.org/10.1109/IROS.2014.6942976, URL https://ieeexplore.ieee.org/document/6942976/.