



OPEN

Robust digital-twin airspace discretization and trajectory optimization for autonomous unmanned aerial vehicles

Mo ElSayed & Moataz Mohamed

The infiltration of heterogenous fleets of autonomous Unmanned Aerial Vehicles (UAVs) in smart cities is leading to the consumerization of city air space which includes infrastructure creation of roads, traffic design, capacity estimation, and trajectory optimization. This study proposes a novel autonomous Advanced Aerial Mobility (AAM) logistical system for high density city centers. First, we propose a real-time 3D geospatial mining framework for LiDAR data to create a dynamically updated digital twin model. This enables the identification of viable airspace volumes in densely populated 3D environments based on the airspace policy/regulations. Second, we propose a robust city airspace dynamic 4D discretization method (*Skyroutes*) for autonomous UAVs to incorporate the underlying real-time constraints coupled with externalities, legal, and optimal UAV operation based on kinematics. An hourly trip generation model was applied to create 1138 trips in two scenarios comparing the cartesian discretization to our proposed algorithm. The results show that the AAM enables a precise airspace capacity/cost estimation, due to its detailed 3D generation capabilities. The AAM increased the airspace capacity by up to 10%, the generated UAV trajectories are 50% more energy efficient, and significantly safer.

Keywords Autonomous UAV logistics, Resilient transportation infrastructure, Energy-efficient trajectory optimization, Aerial mobility operation model, Drones, Digital twin models

List of symbols

O	Set of solid obstacles in LiDAR data
∂O	Solid boundaries of obstacle
χ_O	Indicator function of O
$\vec{N}_{\partial O}(u)$	Inward surface vector normal at point u
$\tilde{F}(f_0)$	Gaussian smoothing filter
\tilde{S}	Oriented batch matrix points
\tilde{P}	Surface Patch at point s
\vec{V}	Vector field
∇	Vector Divergence operator
β	Octree notation
F_o	Associated node function at o
$\vec{V}(q)$	Gradient field indicator function
$\Omega(s)$	Octa closest depth nodes
$\alpha_{o,s}$	Trilinear interpolation weights
δ_o	Flight policy buffer distance (minimum clearance)
\hat{v}	Polygon center projection
v'	Polygon surface sample
w_{min}	Minimum point on polygon
\vec{O}	Minimum normal vector to the tangent polygon plane
\hat{b}	Mesh base point
\mathcal{F}	UAV viable airspace volume

Department of Civil Engineering, Faculty of Engineering, McMaster University, 1280 Main Street West, JHE Building, Room 301, Hamilton, ON L8S 4L7, Canada. ✉ email: archmsayed@gmail.com

Q	Rotation matrix
ω_i	Weighing variable
T	3×1 Translation vector
N	Matrix of B-spline basis functions
\bar{P}	Matrix of curvature degrees
g	Acceleration due to gravity
β_{min}	Minimum flight altitude
β_{max}	Maximum flight altitude
F_{UB}	Urban block airspace
F_{HDR}	High density routes airspace
S	Polygon segment
q_{obs}	Obstacle geometrical center
$D(q)$	Lane disruption function
B_{obs}	Effective matrix of obstacles
\tilde{v}	Initial lane vector flight velocity
\bar{v}	Traversing UAV velocity
M	Model mesh
δ_v	Lane proximity for vertical policy parameter
u	Solid boundary points $u \in \partial O$
s	Matrix points $s \in S$
f	Field points
o	Octree nodes $o \in \beta$
e	Mesh polygons counter
h	Polygon vertices counter $h \in w$
\tilde{k}	Total number of trajectory vertices
\tilde{i}	Trajectory vertices counter
q	Trajectory vertices
γ	Total number of lanes
k	Destination section counter
v	Angular velocity
$\dot{\theta}$	Angle time derivative
τ	Torque
C_τ	Constant of torque
I	Electric current input
V	Rotor voltage feed
R	Coil resistance
C_p	Proportionality constant
P	Power consumption
T_h	Rotor thrust
v_L	Loft velocity
ρ_{air}	Density of air
a	Area covered by each rotor
C	Overall constant
\dot{v}	Angular velocity vector
I_n	Inertia
F_d	Drag force
m	Mass
σ_f	Horizontal flight angle in roll axis
ρ_{ref}	Reference lane curvature
d_l	Target lane change longitudinal distance
\dot{x}_{max}	Maximum UAV lateral acceleration
δ_H	Lane proximity for horizontal policy
$p(q)$	Perpendicular vector to the UAV path
$t(q)$	Tangential vector to the UAV trajectory
MA	Matrix of rotation within both body and inertial frames
ϕ, θ, ψ	Pitch, Roll, and Yaw angles

According to the United Nations, the world population is expected to reach 10.1 billion by 2100, cities are growing exponentially across the globe¹. Given the limited space and resources, the concept of a smart city emerged, which is designed for the optimum usage of space and supplies along with an efficient distribution of resources. Smart cities are, by default, designed to achieve resilient communities that maximize the integration between humans and robotics². Accordingly, the use of autonomous systems is considered a dire need to enable cities' resilience and to cope with the economic, social, and environmental disruptions arising from expansions and increased population density. This has been highlighted recently with novel corona virus (COVID-19) pandemic, which necessitated quarantining and city lockdowns worldwide.

Autonomous systems' integration in cities is featured in several applications such as robotic manufacturing, robotic construction, and autonomous transportation systems. Unmanned Aerial Vehicles or System (UAVs or

UAS) or 'drones' are utilized in a variety of civil and military tasks such as cargo transport, emergency management, and search and rescue missions^{3,4}.

This technological transformation is often equated to the paradigm shift created in automobiles by Henry Ford in the early twentieth century. In other words, the creation of roads, traffic design, and planning can be applicable to the consumerization of city air space. While UAV applications are relatively easier in rural areas, however, several challenges arise with the anticipated proliferation of heterogeneous UAV fleets in low-altitude airspace of dense urban areas given characteristics of cities and the definition of relevant decision variables⁵. These challenges can be bundled according to the elements of the autonomous UAV system, the UAV itself, and the city airspace as the hosting infrastructure.

As it relates to the UAV, and to maintain the overall weight of the UAV, a trade-off is inevitable between the on-board power source, processing unit, and sensors essential for autonomy, and the transported payload limits and/or the flight range. This trade-off decreases the practicality of the whole independent UAV system⁶. Significant advancements in UAV technologies promise an increased energy efficiency, lighter airframes, and improved power-to-weight ratio for DC motors. However, these improvements are not expected to reflect substantially on the existing performance in the near future^{7,8}. Therefore, research has depended on developing routing algorithms or trajectory planning and optimization heuristics to tackle current UAV limitations⁹. Most relevant to low altitude airspace management (LAAM) applications is a UAV Routing and Trajectory Optimization Problem (RTOP), where a fleet of UAVs visits a set of waypoints (missions) assuming UAV kinematics (position, velocity, and acceleration), and dynamic (forces and moments) constraints. This represents half of the solution, since it depends on the presence of a viable discretized airspace that respects all other constraints.

Considering city airspace challenges, different variables exist. First, massive fleets of UAVs operating in highly dense cities raises serious safety issues as huge damage can be sustained to pedestrians and public or private property. This damage can be caused by the crashing of a UAV due to a technical malfunction or mid-air collisions due to airspace interference and congestion^{10–12}. Second, UAV onboard communication and GPS navigation modules are vulnerable to security breaches due to the unencrypted nature, which makes it easily spoofed^{13,14}. Third, given their data collection abilities, sensors, and high-precision onboard cameras, UAVs can be perceived as remotely controlled surveillance equipment¹⁵ as they can be hacked to collect personal data or track individuals using wireless localization techniques. Fourth, the proximity to public operation causes pedestrians to feel uncomfortable or dwellers to feel that their privacy is being invaded³.

Given these challenges and the traditional concepts of city security, liability, and aviation airspace regulations, the need to regulate UAV operation pushed international, federal, and local governments to navigate uncharted territories, with boundaries of civil regulatory authority over UAVs ill-defined^{12,16}. Currently, several countries have imposed UAV operational restrictions based on proximity to population and man-made structures. While these regulations alone can control leisure UAV use, however, heterogeneous fleet operation with projections of massive volumes of UAVs is too large for the current Air Traffic Control structure to handle^{17,18}.

Reacting to that, two types of research exist, one group focusing on enabling safe urban airspace operation through geofencing and airspace discretization or air traffic control. The other group works on developing routing algorithms or path/trajectory planning heuristics for conventional cartesian airspace. While some of the previous research consider some of the aforementioned challenges/parameters, others fall short in providing comprehensive applicable frameworks/solutions^{3,4}. The adoption of UAVs in autonomous transportation within smart cities hinges on the development of a full city aerial-infrastructure framework for operation (airspace discretization and geofencing), navigation (trajectory planning), and traffic control of swarms of UAVs running on robotic operation systems (ROS)¹⁹. This framework must consider all challenges raised by previous studies across all parameters before proceeding into a real-life execution phase.

In this respect, the present study develops a novel autonomous Advanced Aerial Mobility (AAM) system for high density city centers. The AAM system integrates a digital-twin city-airspace discretization, planning, and trajectory optimization algorithm for heterogeneous UAV fleets.

To the authors' knowledge, this is the first study to integrate live, updated, precise digital models with airspace planning for exterior complex urban environments. An extensive review of the most recently developed methodologies integrating GPS data and LADAR for UAV pose estimation and trajectory is provided by Vaidis²⁰, and the latest LiDAR 3D processing techniques by Wang et al.²¹. This also is the first study addressing the intertwined city airspace regulatory challenges and the multiple parameters for efficient UAV operation within digital-twin models. The primary aim of this study is to develop a novel comprehensive algorithm that allows autonomous AAM operation within civil airspace. The model depends on dynamically updated real-time LiDAR data to simulate the actual civil airspace and converge energy-efficient pre-planned obstacle-avoiding trajectories instead of active path planning for each UAV. The proposed system solves airspace planning and UAV control/navigation challenges, accommodating variable UAV sizes, types, and speeds. Furthermore, ensuring abidance to respective airspace regulations and maximizing capacity.

To achieve this aim, the present study,

1. Develops a real-time 3D geospatial mining framework for geo-referenced allocation of trips and UAV task assignment based on LiDAR data to create a digital twin model.
2. Proposes a novel city airspace dynamic 4D discretization method (Skyroutes) for UAVs based on legal regulations incorporating real-time constraints coupled with external factors. The discretization converges a network of keep-in lanes allocated outside the keep-out geofence (dual geofencing).
3. Utilizes a flexible energy use model for multi-rotor UAVs based on the kinematics and dynamic operational capabilities and calibrated to measurements from representative experimental UAV flights²².

4. Develops a dynamic trajectory optimization method tailored for the proposed discretization method coupled with a novel 3D lane-change and compare the solutions' efficiency to the existing algorithms in the literature.
5. The developed models are applied to a real-world case study to computationally simulate UAV transportation operations delivery applications.

In this study, after presenting the airspace discretization model, we formally define the UAV routing and trajectory optimization based on quadcopter kinematics and dynamics. We utilize Newton–Euler derived differential equations to simulate the operation of UAV brushless DC rotors. Thereafter, we utilize a complementing simplified real-time Dynamic Programming (DP) arc-routing method to determine minimum Snap and energy trajectory for a fleet of UAVs visiting a set of arcs between origin locations and destinations. In this respect, the presented study provides an original contribution to the AAM challenge.

After this introduction, a literature review focusing primarily on different approaches to UAV-city-integration through civil air-space discretization and UAV trajectory planning research is presented in “[Literature review](#)”. “[Methods](#)” introduces the study methodology, while “[City digital twin model](#)” and “[Airspace discretization model description](#)” include the Digital-twin model and the proposed urban airspace discretization derivation model respectively. The modified energy-optimal trajectory planning and UAV task assignment framework are detailed in “[Trip generation, Cartesian routing, and UAV energy consumption](#)”. “[Case study, results, and discussion](#)” reports on the case study, the results, and the discussion, while conclusions are presented in “[Conclusions and future studies](#)”.

Literature review

Currently, UAVs' operation is limited below flying altitude of commercial aircraft to avoid collision potential. Globally, this can be generally defined as zero to 150 m over ground level²³. Although autonomous UAV mission control can be performed onboard with the reliance on sensors, GPS, and computation. However, in proximity to buildings and in case of severe weather conditions, UAVs are prone to loss of GPS signal or sensor failure jeopardizing the efficiency and stability of the entire network^{24,25}. Hence, off-board preloaded mission planning maximizes safety, utility, reliability, and mitigates the need for onboard multiple sensors saving weight for payload and decreasing costs. Autonomous operation within densely built-up areas could interfere with UAVs²⁶.

In response to that, earlier research on LAAM recommended the development of a unified urban airspace system or ‘urban air mobility’ to manage the safe operation of UAVs within low-altitude civilian airspace. For instance, in 2006, the International Civil Aviation Organization (ICAO) declared the need for international harmonized terms and principles to guide the civil use of UAVs²⁷. Later in 2020, the term Advanced Aerial Mobility (AAM) was coined by NASA denoting the ecosystem incubating the emergence of these disruptive technologies in both urban and rural contexts²⁸. The published report outlines a vision for the city airspace and air traffic management environment.

The main concept is to establish a national framework through a unified infrastructure with levels of complexity for all manned and unmanned aerial vehicles of any size or type to control traffic, separation, and flight trajectories. The airspace traffic network is to be based on data sharing and utilized as a nationally controlled utility provided for various mobility operators similar to current ground road networks²⁸. To that end, the economic, social, and regulatory success of the system is dependent on addressing some fundamental challenges which can be summarized as follows:

- Safety and security: any AAM must ensure the safety of public and private property and users such as collision avoidance, limiting extreme proximity and mitigation of street level drivers' visual distraction. In addition, including cybersecurity against communications signal hacking beside other system vulnerabilities in extreme weather conditions and disruptive events^{29,30}.
- Environmental impacts: this entails minimizing or eliminating GHG emissions, noise, and impact on wildlife²².
- Flexibility and resilience: the system's ability to recover quickly from unexpected events and limit the cascading impact. Furthermore, the ability of the system to evolve with the emergence of newer UAV technologies, software, and operational concepts³¹.
- Regulation: develop standardized national policies to govern operation and allow insurance and tax or toll collection.
- Social acceptance: being a disruptive technology with a social stigma, experimental real-world operations and scenario-based analyses can convince the users that the urban obtrusiveness risk is acceptable, and efficient to overcome cost barriers.

To tackle these challenges, over the last decade ample research has aimed to provide solutions or guidelines, those can be bundled in two groups according to the targeted study area, airspace planning research and UAV navigation and control research.

Urban airspace planning

The concept of airspace planning as explained from the AAM perspective and challenges is new to the research community²⁸. However, a substantial part of this area is commonly researched under the UAV Traffic Management (UTM) keyword, where ample literature exists^{32,33}. With that said, only literature on obstacle-rich lower urban airspace is considered in this review rather than obstacle-free higher-altitude airspace.

The most researched concept of separating flyable airspace from obstacles in UTM is known as Geofencing³⁴. Geofence is a virtual perimeter applied statically or dynamically in a real-world application in positive (keep-in)

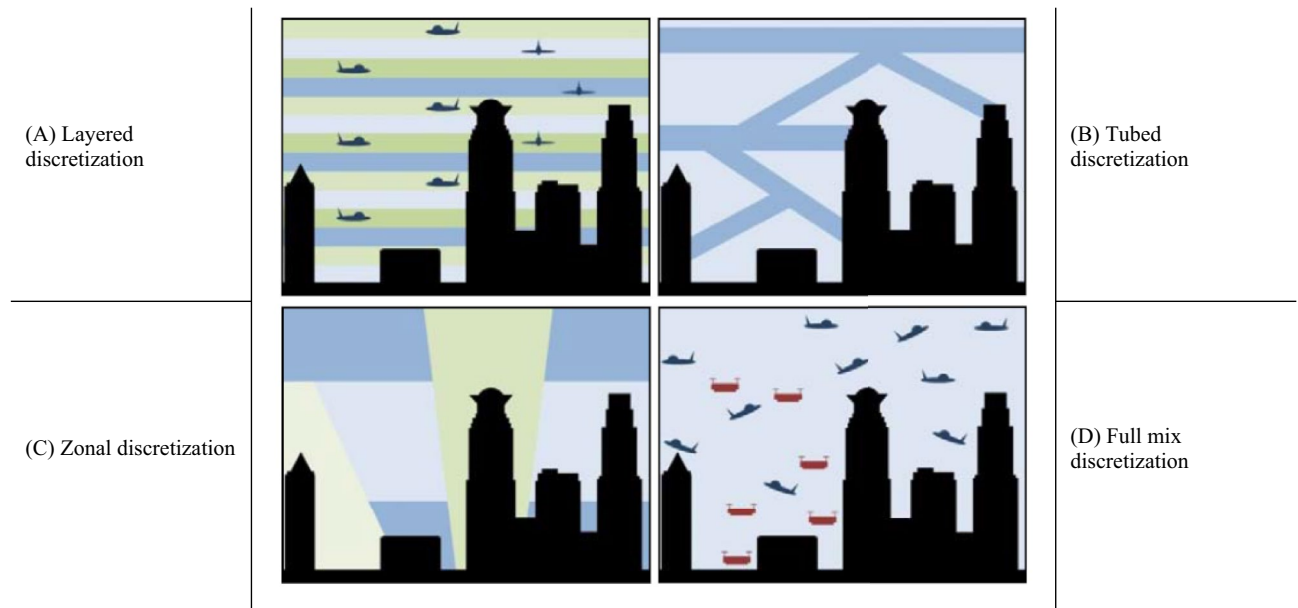


Figure 1. Different airspace discretization morphologies³⁶. Source: Patrinoopoulou et al., 2022.

or negative (keep-out). While the keep-in geofence is a 3D volume to maintain, the keep-out is applying volumetric restriction to certain extents. With current UAV regulation generally including a minimum distance or protection boundary around static objects (e.g., people, buildings, and structures) and altitude limit²³, keep-out is the most deployed and researched methodology³⁵.

Urban airspace planning depends on two factors, namely the quality of the 3D environment model and the geofencing technique utilized. The accuracy of estimating a real-time state of UAVs is highly dependent on digitally replicating the real-world environment. This relies primarily on the collected surrounding sensor data. Literature has mainly depended on 3D GIS, Digital Surface Model (DSM), or Google's 3D city data for their system simulations^{36–38}. While 3D maps provide viable results, they fall short to include details and dynamic changes to the real-world environment. The missing details and changes include transmission towers, utility poles, power lines, construction equipment including cranes, and street level vegetation.

The integration of airspace planning and geofencing has been studied under various discretization morphologies^{37–41}. Most comprehensively, Hoekstra et al.³⁶ illustrate all the discretization morphologies (Fig. 1). In all morphologies, on-board avionics implement the preloaded regulation and specific flight plan autonomously, and the flight trajectory is governed through the geofence.

Another stream of studies targeting specific challenges exist in the literature. Although they do not provide holistic solutions, however, their conclusions and recommendations are exceptionally valuable for building on towards the objective of this study. D'Souza et al.⁴² tested the flight deviation from planned path due to wind disturbances. Their study concluded that PID controller stabilization can decrease the minimum lateral distance from buildings up to 5 m. Similarly, Johnson et al.⁴³ tested applying several minimum lateral distance alternatives on the ability of UAVs to detect and avoid buildings. Their results showed poor detection capabilities with narrow urban corridors. Recently, Cho and Yoon⁴⁴ compared three scenarios for the case study of Seoul city, namely keep-out, keep-in, and dual geofencing. They concluded that keep-in exhibited more robust behavior than the keep-out. The study recommended integrating both geofencing methods while applying dynamic parameters given the geospatial complexity and flight purposes. More recently, Torija et al.⁴⁵ compared a series of audio-visual scenarios for UAV operation in cities to investigate the impacts of UAV noise. They concluded that the UAV operations along busy roads might aid in the mitigation of the overall community noise impact.

Overall, the concept of AAM has been mirrored globally in numerous studies with the aim of establishing a comprehensive UAV airspace discretization framework. However, most research has focused on the integration of one or two challenges rather than addressing all challenges.

Table 1 presents a summary of the most relevant literature outlining the solutions and recommendations for each challenge. Although several other studies address the same topic, the following limitations were applied in the filtration process. (1) The oversimplification of the problem, which makes the solutions less robust for city-scale application. (2) Proprietary restrictions that prohibits the open collaboration on developing, integrating, and testing the suggested solutions¹⁸. (3) Solutions not targeted for autonomy and beyond visual line-of-sight (BVLOS), since solutions for piloted systems are significantly different and not viable⁴⁶.

UAV flight navigation and control

Given a fleet of UAVs and a volume of designed urban airspace, the actual navigation and mission control can be described as a series of complex mathematical problems. First, the NP-hard UAV Task Assignment (TA) problem refers to optimally assigning missions to a set of UAVs based on mission constraints⁴⁷. While TA shares similar characteristics with Vehicle Routing Problem (VRP), there are a few key differences^{48,49}. Unlike VRP, TA allows

Project	Safety and security	Environmental impacts	Flexibility and resilience	Regulation	Social acceptance
Metropolis project ³⁶	Full mix, layered, zonal, and tubed	Optimal route geometry	Flow management, separation, and conflict avoidance	Considers only average regulation strictness	Presenting alternative scenarios
	Limited static airspace planning in two models. Exclusive limitation of operation within obstacle-free airspace decreases mission range for smaller UAVs and increases mission time for emergency application				
Singapore's TM-UAS ^{37,38}	Full mix (AirMatrix), over buildings, and over roads	N/A	Flight and risk management	Varying degrees of regulation	Presenting alternative scenarios
	Limited static waypoint concept. Limited route optimization and flexibility with UAV types and sizes given only constant speed or random speed optimization limitation in traffic control				
Australia's Smart Skies project ³⁹	Automated separation management system	N/A	Sense-and-Act Systems	Local current regulation	N/A
	Limited model details. Limited route planning and optimization capabilities through the developed Mobile Aircraft Tracking System (MATS)				
NASA UTM ^{40,41}	Collision avoidance	N/A	Contingency management and re-routing	FAA regulation	Public safety illustration
	Limited multi-modal airspace discretization and modification				

Table 1. Urban airspace planning literature.

multiple stops, heterogeneous fleet operation, and mission subtours. The output for both VRP and TA is a pairing between a set of O-Ds, and assigned vehicles along with a set of waypoints.

Second, to connect these waypoints and form a UAV flyable path, the problem is known as Path Planning (PP). PP is defined as the process of constructing a geometric path from a starting point to an end point given a 2D or 3D domain. While PP can include the impact of wind or other constraints, the problem formulation is extensively simplified to be solved heuristically⁵⁰. In a real-world application, it is imperative to couple the generated path with UAV constraints, kinematics, and dynamics⁵¹. In this respect, the integration of kinematics and dynamics with routing is known as optimal motion (trajectory) planning or Trajectory Optimization (TO). Closely related to the Optimal Control (OC) problem, TO leverages motion equations to model the spatiotemporal changes of the UAV system while minimizing a scalar performance index such as flight time or fuel consumption⁵². A detailed literature review of the research focused on these problems can be found in the literature¹⁹. They conclude that on one hand, the UAV routing and task assignment literatures have mostly neglected complex UAV constraints. On the other hand, TO research has fallen short in integrating other noise and safety challenges.

Research about integrated routing and trajectory optimization is scarce¹⁹, however, there are ample studies discussing different methods for UAV navigation and mission control. Literature can be classified based on methodology into three major groups. (1) Mixed-Integer Linear Programming and exact algorithms (such as branch and bound or Euclidean minimum spanning tree), where an optimal solution is guaranteed; (2) Metaheuristics such as Evolutionary algorithms, Particle Swarm Optimization, and Ant Colony, where a solution is not guaranteed; (3) Heuristics that includes merging different heuristics or special cases of algorithms such as hybrid Tabu Search-Simulated Annealing⁵³. Several other methods are presented in the literature without belonging to a specific group. To focus on relevant literature, we only present in Table 2 studies that can perform combined routing and trajectory optimization independent of urban airspace planning. Therefore, obstacle avoidance (static or dynamic) and 3D environment operation are imperative capabilities⁵⁴.

To that end, while limitations of case-by-case studies in the literature can be addressed, all routing and trajectory optimization methods presented utilize a cartesian discretization rather than incorporating a discrete airspace planning and discretization component leading to a full-mix airspace concept. In this concept, airspace is unstructured and UAV traffic is fully dependant on onboard sensing, self-regulation, and obstacle avoidance under stochastic conditions. Although the full-mix airspace concept allows for maximum speed and freedom, given heterogenous fleet operation, it severely limits the energy efficiency, airspace capacity, as well as jeopardizing system-wide safety. Hence, a full-mix concept fails to address the aforementioned AAM challenges²².

The innovation in this study diverges from existing work primarily through its dynamic 4D discretization method known as "Skyroutes", which contrasts with traditional Cartesian discretization methods. In lieu of Table 2, Skyroutes is designed to be open-source and incorporate real-time constraints along with legal and optimal UAV operation considerations based on kinematics. This allows accounting for wind consideration. Furthermore, the simplified computational burden allows for real-time applicability to a heterogeneous fleet/swarm operation. Prior relevant studies have relied on 3D GIS Digital Surface Models^{63–65} or similar data sources for system simulations^{60–62}, which, while effective, do not account for dynamic changes to the real-world environment and often lack the details necessary for accurate urban airspace planning.

Integrating real-time, high-resolution digital twins in the proposed AAM system represents a significant advancement in urban airspace management, providing a more accurate and dynamic method for airspace capacity estimation and UAV trajectory optimization. This system not only improves upon the limitations of existing approaches with its detailed and adaptive modeling but also demonstrates the potential for increased airspace capacity and safety in urban environments. The novel Skyroutes discretization method, in particular, represents a key differentiation from traditional methods, offering a more efficient and flexible solution for airspace management in densely populated urban settings. The proposed system is not a stand-alone platform. However, it could be applied within a platform similar to the European U-Space⁶⁶.

Method/author(s)	Method	Application	Simulation/experimental verification
Trajectory optimization of multiple quadrotor UAVs in collaborative assembling task ⁵⁵	Genetic algorithm	Uncapacitated Multi UAV trajectory optimization	Simulation
	No real time applicability for heterogeneous fleet/swarm operation. No wind consideration		
3D off-line path planning for aerial vehicle using distance transform technique ⁵⁶	Multi criteria decision analysis	Off-line path planning	Simulation
	No real time applicability for heterogeneous fleet/swarm operation. Limited UAV dynamics accountability and lacking wind consideration		
A heuristic mission planning algorithm for heterogeneous tasks with heterogeneous UAVs ⁵⁷	Heuristic algorithm	Mission planning for heterogeneous tasks	Simulation
	No real time applicability and wind consideration. Limited UAV dynamics accountability		
3D multi-constraint route planning for UAV low-altitude penetration based on multi-agent genetic algorithm ⁵⁸	Genetic algorithm	Mission multi-constraint route planning	Simulation
	No real time applicability for heterogeneous fleet/swarm operation. Limited UAV dynamics accountability and lacking wind consideration		
Distributed pseudolinear estimation and UAV path optimization for 3D target tracking ⁵⁹	Gradient-descent algorithm	UAV path optimization for 3D target tracking	Simulation
	No real time applicability for heterogeneous fleet/swarm operation and wind consideration		
Online path planning for UAV using an improved differential evolution algorithm ⁶⁰	Differential Evolution Algorithm	Online path planning for UAV	Simulation
	No real time applicability and wind consideration. Limited UAV dynamics accountability		
Trajectory planning for unmanned aerial vehicles in complicated urban environments: A control network approach ⁶¹	control network and Dubins curve Algorithm	Two-stage control network approach	Simulation
	No applicability for huge-scale cities, the control network could contain billions of links and it may cause the path-finding problem computationally burdensome/over-simplification of the city model		
3D path planning and real-time collision resolution of multirotor drone operations in complex urban low-altitude airspace ⁶²	3D voxel jump and Markov decision process	Autonomous drone collision-free path planning	Simulation
	Originated from the classical 2D grid map JPS method considers only diagonal or straight directions/over-simplification of the city model		

Table 2. Relevant UAV 3D routing and trajectory optimization literature.

Methods

The study proposes a novel three-step sequential methodological approach (Fig. 2). Each step is detailed in the following sub-sections. In the first process, a digital-twin for the simulated case study is built and actively updated for real-time changes and disruptions.

Subsequently, the system obtains two streams of input for a set of variables through an online connection. The first stream of input relates to mission planning while the second stream relates to the area-specific flight regulations adopted. Using a keep-out geofence, all geometry is interpreted into physical obstacles. The second process starts by sorting the heterogeneous input UAV and applying our proposed novel *Skyroutes* algorithm to produce a discretized airspace. This triggers the third process for task assignment, routing, and trajectory optimization. The fourth process is the final system output including the maximum airspace capacity. At the end of the procedure, the framework visualizes the UAV trajectories and provides the trajectories. An active loop is initiated between the UAV proportional–integral–derivative (PID) controller to correct the trajectory navigation as the operation progresses.

City digital twin model

Detailed spatial information infrastructure crucial for the AAM system, however, it should be lightweight enough for the Ground Control System (GCS) and Central Control System (CCS) to handle in real-time. Given the number of details in urban environments and spatial approximation of object-based 3D information pose significant challenges on computational power and time. In this study we only require a level of tolerance < 1 m excluding take-off and landing, which is a function of onboard sensors and under slow flight speeds.

The city is divided into small bands using a 3D clustering method proposed by Youn et al.⁶⁷ to maintain the memory consumption and computational power within viable limits. Their UAV 3D clustering proposes a 20-level grid division. Having equal division possess two challenges, first, the obstacle details are not equal comparing urban to rural contexts, which leads to computing memory unbalance. Second, the synchronization between this independent classification versus existing addressing and GPS positioning system adds a layer of computationally demanding processing. We modify their clustering system via 3-digit postal code classification to ease the geo-referencing with existing census population density for trip generation (Fig. 3). Further explanation of this partitioning scheme is beyond the scope of this study.

For each selected airspace planning zone, first, DSM is imported and overlaid to the OpenStreetMap (OSM)⁶⁸ which acts as the base map that includes the vector data for precision 3D GIS alignment. The GIS map includes most data layers such as streets, zones, functions, and property outlines. Second, to incorporate vertical building façade details (windows and balconies), municipal open-data environment is imported, scaled, and georeferenced in the simulation model. Finally, most recent real-life LiDAR data is merged in the model for interpolation and updates to make sure the digital-twin model can truly reflect the reality once UAVs are deployed in a large scale.

Since LiDAR data are characterized by noisy patterns due to errors and the complexity of surfaces, these datasets require further processing to be usable for discretization. A variety of 3D extraction algorithms is discussed in the literature²¹. For the purpose of airspace planning and navigation for UAVs, there is no need for

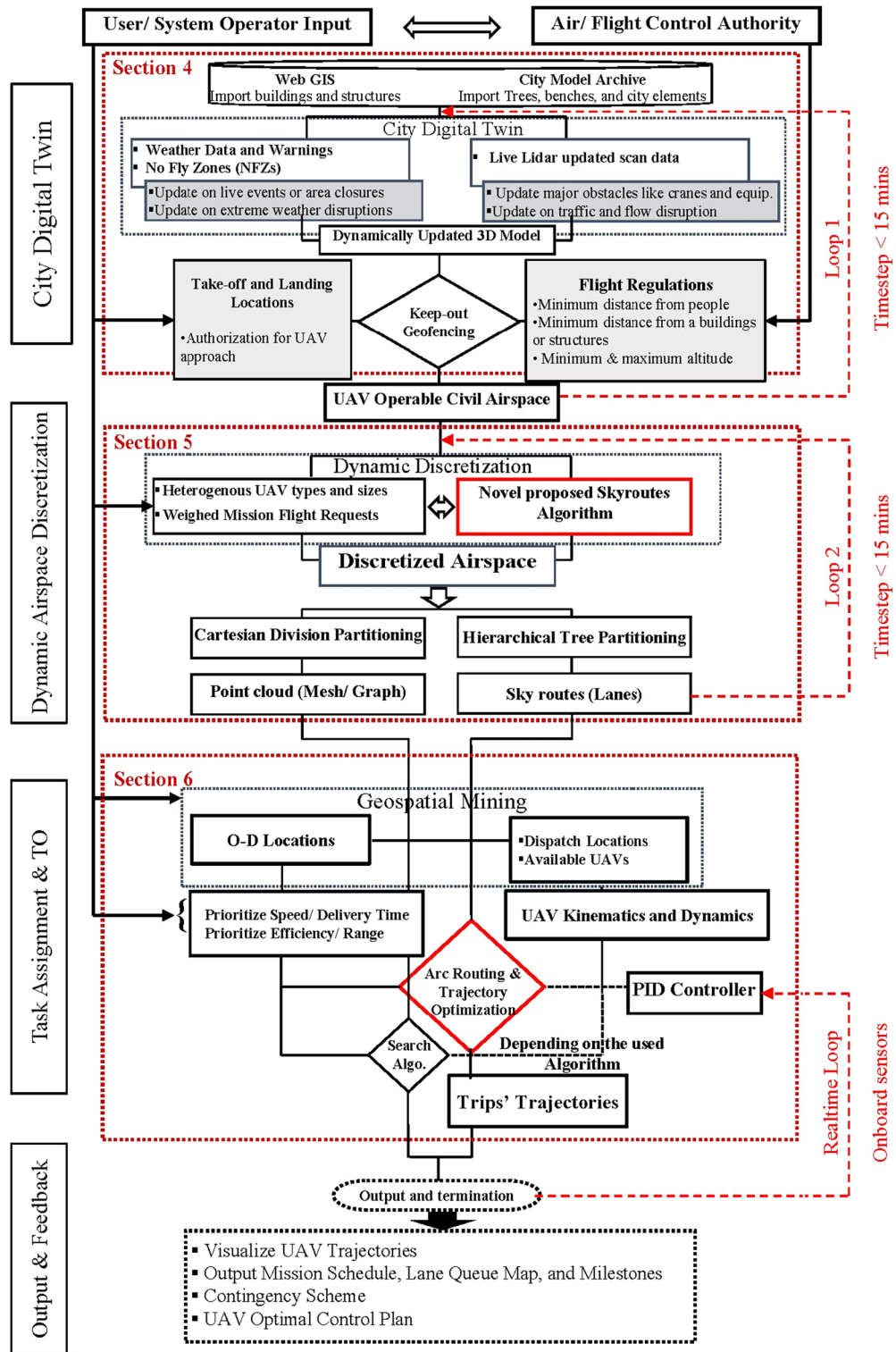


Figure 2. The developed methodology.

distinction between urban elements such as buildings and trees., (i.e., the objective is to avoid all obstacles). In this study, we modify a Freeform objects reconstruction algorithm via Poisson method proposed by Kazhdan et al.⁶⁹ instead of a topographical or building extraction algorithm. The Poisson method is widely popular due to its scalability and efficiency where it can reconstruct freeform objects fast and with reasonable accuracy²¹. The solution Eqs. (1–4) are outlined in the appendix as they are auxiliary to the research question.

At this point, we have attained realistic iso-surfaces to construct a mesh of the existing real-world city environment. However, to comply with regulations, the horizontal distance or protection boundary around objects

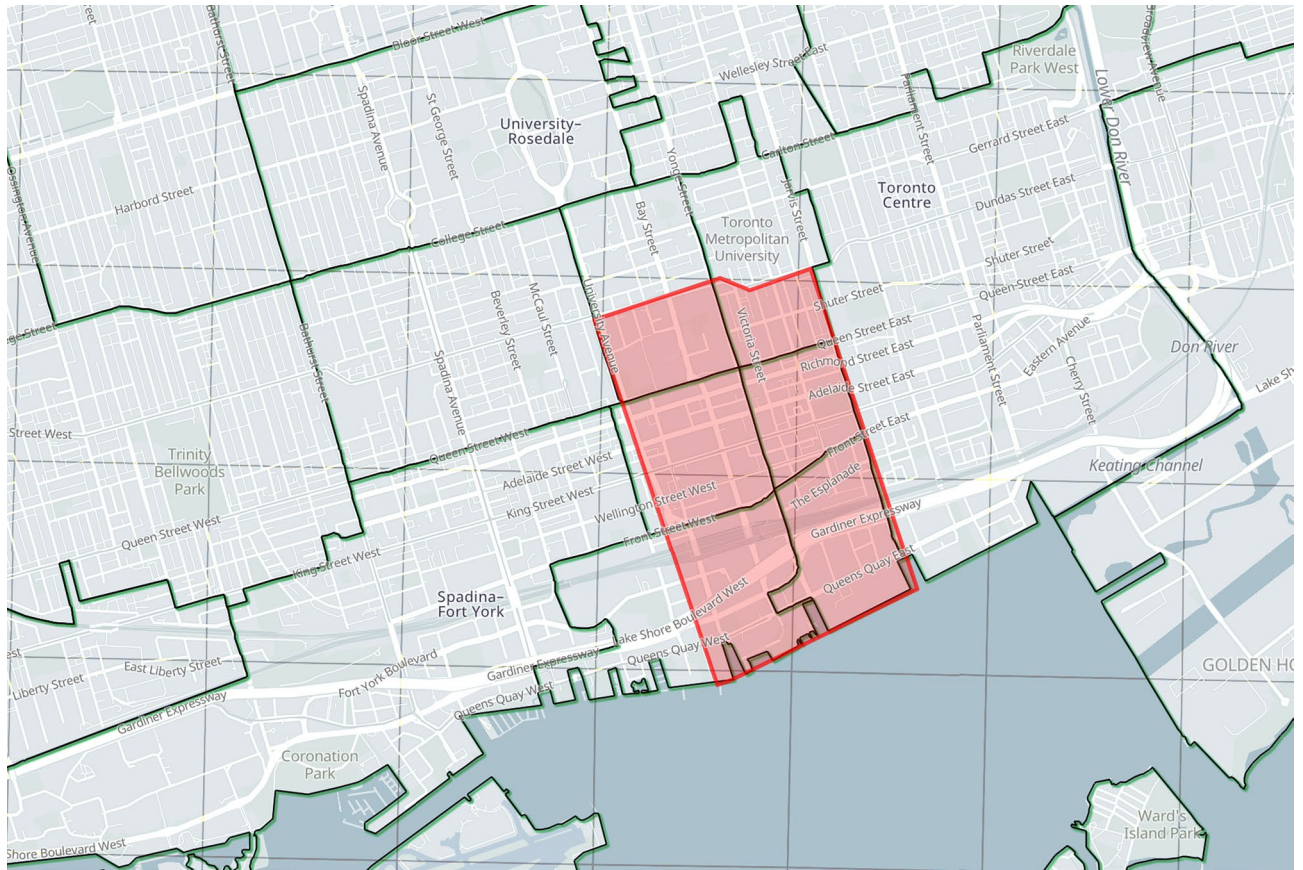


Figure 3. Overlay of 3D data sourcing, Blue (municipal), Green (postal), Black (lidar).

must be added to the obstacles model. Therefore, the model is shifted by a distance δ_o to offset obstacle meshes outward according to the applicable flight regulative distance. This is a direct application of the mathematical problem known as constant-distance offset (CDO) or specifically Minkowski sums for 3D geometries. Since the dataset contains complex non-convex polygons, we overcome this by utilizing a modified 3D scaling algorithm to shift each boundary representation (Brep)/ mesh face with the exact policy enforced distance. The Minkowski sum of two geometry sets A and B is defined as $A \oplus B = \{a + b | a \in A, b \in B\}$. If we take A to be the arbitrary input mesh and B a sphere of the given radius equal to policy-driven value δ_o centered at the origin, then an offset surface is defined as the boundary of their Minkowski sum. Detailed mathematical formulation for solid offsetting can be found in the work of Rossignac and Requicha⁷⁰.

Given the city’s complex highly detailed polygonal mesh, each obstacle boundary O with Brep/mesh faces interpolates polygons $\int_{\delta_o}^e \vec{p}_o = (x, y, z) = [(x_1, y_1, z_1), (x_1, y_1, z_1), \dots, (x_w, y_w, z_w)]$. The new shifted faces are prescribed by set of vertices, $v'' = (x'', y'', z'')$. However, the new offset boundaries’ sum can include parts of spheres, cylinders, and prisms corresponding to vertices, edges, and faces of the mesh, respectively. To have closed Brep suitable for Boolean operations, the union of these different elements is essential. For the octree with depth δ D_e , the octree root cell is initialized as the bounding box of the offset surface.

To optimize memory for this model size, we further discretize these bounding boxes into smaller voxels that are merged into a unified surface later. The voxel layer is divided into overlapping tiles to ensure a tight surface. For a maximum octree refinement (\varkappa) and a grid of tiles ($grid$), the voxel grid is $((2\varkappa - 1)grid + 1)^3$. To eliminate invalid self-intersecting geometries in tight urban canyons Fig. 3a, we use a filtration condition where the invalid surface polygons are removed when they do not have a neighbor polygon with $\delta_o \leq$ minimum offset distance. Remaining polygons are processed utilizing a modified Dual Contouring Algorithm^{71,72}. The modified method is adjusted for model processing to optimize computational power and eliminate noise. In Rhinoceros modelling the obstacle boundary O for each polygon e , the minimum normal vector to the tangent polygon plane \vec{O} , center projection \hat{v} , and the surface sample projection \hat{v}' are given by:

$$\vec{O}^T = \vec{O}^T w_{min} + \delta_o - \delta_{min} \tag{1}$$

$$\hat{v}' = \hat{v} + \vec{O}(\vec{O}^T (w_{min} - \hat{v}) + \delta_o - \delta_{min}) \tag{2}$$

where δ_{min} is the minimum offset distance; w_{min} is the minimum offset distance point on polygon.

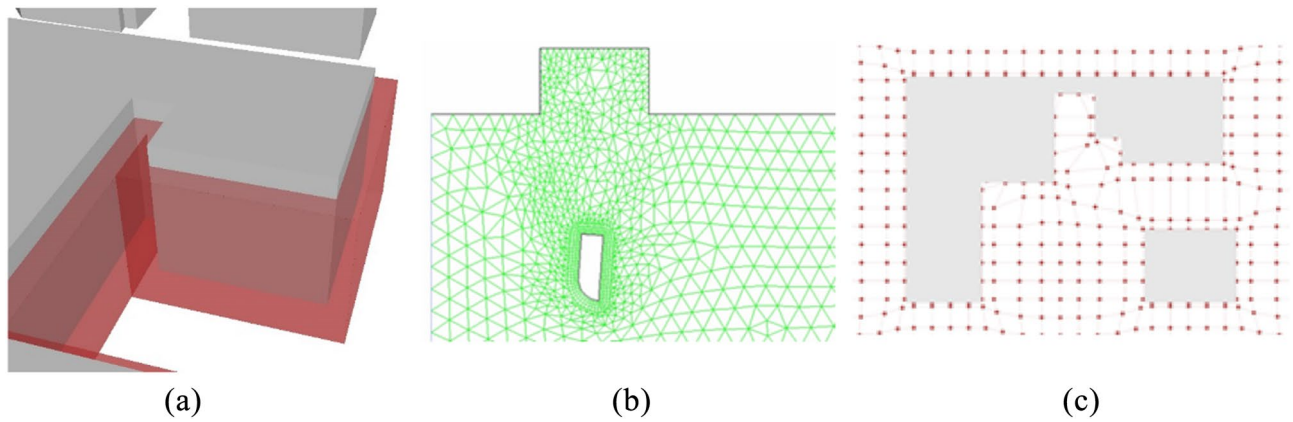


Figure 4. (a) Self-intersecting urban mass, Grey (original geometry), Red (offset geometry). (b) CFD meshing⁷⁵. (c) Dynamic meshing illustrated on urban mass after smoothing.

To guarantee the generated cells lie on the surface, a smoothing mesh function is utilized. For the final generated offset mesh M , a relaxation force pulls every vertex v_h in the mesh vertices towards v' while offset force pulls v towards v'' as follows:

$$v' = \frac{1}{w} \sum_h v_h \quad (3)$$

$$v'' = \hat{b} + \delta_o \frac{v - \hat{b}}{\|v - \hat{b}\|} \quad (4)$$

where \hat{b} is the base point on mesh M with the minimum distance to v_h . This mesh relaxation techniques allow the inclusion of tight urban geometries, which could significantly impact the airspace capacity. Furthermore, it allows a better applicability for other keep-in geofences that we are not using in the study such as the shape method by Edelsbrunner et al.⁷³.

The smoothed mesh helps with the adoption of a novel dynamic meshing technique similar to CFD in buildings simulations⁷⁴. The dynamic mesh accommodates and changes according to the model space, in self-intersecting geometries Fig. 4a or around obstacles, the mesh gets stricter (i.e., the spacing between graph vertices gets smaller) and vice versa in wider or obstacle-free areas where the mesh spacing gets wider as illustrated in Fig. 4b,c. This cartesian meshing will be explained in “Trip generation, cartesian routing, and UAV energy consumption” for graph-based solvers.

To proceed with airspace planning, we create a virtual box with the bottom as the 3D model ground surface, side boundaries taken from the simulated city patch and the top is constructed at the maximum flight altitude (β_{max}) given from the simulated policy. A Boolean subtraction process subtracts the entire 3D model with an offset value of the minimum horizontal distance from property as the δ_o offset value from the airspace boundary virtual box. The resultant volume (\mathcal{F}) is the UAV motion viable airspace.

To keep the Digital-Twin updated, loop 1 is performed within a predetermined time-step to input the updated LiDAR data with any significant changes that might cause disruption. Data is processed in (1–8); thereafter, the airspace discretization model is updated. A queen UAV is expected to update in the defined time step. In this study, we utilize a combination of variance estimation model-driven and point cloud-based Iterative Closest Point (ICP) methods to align the geometry of two roughly pre-registered, partially overlapping, rigid, noisy 3D point sets⁷⁶. The code is written in Python.

Given the stored city model mesh set M and the new LiDAR dataset M^* , for each point $v_h^* \in \partial M^*$, we allocate the closest complementing point(s) v_h in M . Consequently, we compute the incremental transformation using a weighted least-squares function given by⁷⁷:

$$\min_{(Q, \vec{T})} \sum_h \omega_h \left\| M_h - (QM_h^* + \vec{T}) \right\|^2 \quad (5)$$

where Q is rotation matrix, \vec{T} is the translation vector. The weighing variable ω_h is set to zero if the Euclidean distance (ED) between v_h^* and v_h defined as ($d_h \triangleq d(v_h^*, v_h)$) is larger than the maximum tolerance threshold δ_{max} set to 1 m in this study. This determines the motion in existing elements of the model such as limited movement urban elements (trees, scaffolds, and construction equipment). However, new elements in the LiDAR data that falls within the boundary of (\mathcal{F}) are added as identified obstacles O undergoing the process in (1–8) to be integrated with the new mesh set M .

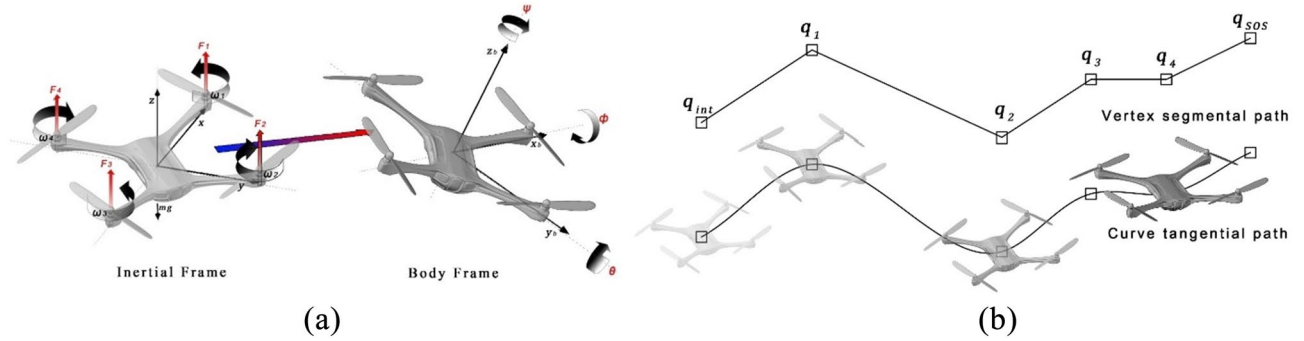


Figure 5. Quadcopter motion dynamics. Source: Authors created by Rhinoceros 3D, version 7, rhino3d.com. (a) Inertial and body frames, and Euler angles ϕ , θ and ψ . (b) Vertex transition vs curve transition path.

Airspace discretization model description

In this section, the proposed airspace discretization method is formally explained. A brief explanation of the UAV flight trajectory dynamics is presented in “[Dynamic trajectory properties](#)”. The novel airspace discretization model morphology is discussed in “[Airspace discretization morphology](#)”. The proposed keep-in geofence and geometrical disruption of UAV flight trajectories (*Skyroutes algorithm*) are discussed in “[Robust Skyroutes algorithm](#)”.

Dynamic trajectory properties

In this study, we utilize two different airspace discretization methods, namely, segment-based and cartesian-based. While segmental discretization includes a path geometrical optimality component, cartesian-based paths are, by default, generated as a set of straight-line segmented polyline paths. The complexity of the geometry depends on the mission initiation and destination locations, flight policy, and the characteristics of the obstacles to be evaded. Since UAVs propagate along a continuous trajectory, a hard-angled segmented flight path is not feasible or may lead to overshooting from the keep-in geofence. Similarly, the integration between both types of generated paths in a single flight plan requires a viable geometric transition Fig. 5a.

In the literature, this problem was tackled by Bézier curves to reform the generated flight path⁷⁸. However, common generating algorithms of Bézier curves can tend to be computationally inefficient⁷⁹. In this study, interpolated fit-point ‘cubic’ splines are adopted for computational efficiency. The method is based on B-spline interpolation function and UAV motion equations. A B-spline with fit points transitions from cartesian point cloud reference is utilized to reform vertex to curve transition (Fig. 5b). For the UAV, the generated path is a B-spline rather than a set of straight-line segments polyline. Although for cartesian discretization, most optimal trajectory generation algorithms (Table 2) adopt their individual methodologies to generate the most energy efficient trajectories. However, this base B-spline method is needed as the shortest path for optimization in some methods. The curve equation for path correction is given by:

$$\bar{C}(q) = \sum_{\hat{i}=0}^{\hat{k}} N_{\hat{i},deg}(q) \bar{P}_{\hat{i}} \quad (6)$$

where \hat{k} is the number of vertices along the trajectory; N is the matrix of B-spline basis functions for vertices $q_{\hat{i}}$ to $q_{\hat{i}+1}$; the degree of curvature is determined by (*deg*) based on the UAV kinematics, the detailed iterative process is outside the scope of this study; and $\bar{P}_{\hat{i}}$ is the matrix of curvature degrees for vertices $q_{\hat{i}}$ to $q_{\hat{i}+1}$.

Airspace discretization morphology

The four different airspace discretization morphologies are discussed in the literature and summed up in “[Urban airspace planning](#)”, Fig. 1. In Elsayed and Mohamed^{22,80–82}, the impact of airspace regulations and flight path geometry/trajectory on the energy consumption and GHG emissions was illustrated, however, the study results showed the challenge of failed missions. In this study, we overcome the mission failure and inviable trajectories by proposing a novel logistic dynamic discretization morphology that combines the advantages from each discretization method and eliminates the disadvantages.

Starting with city obstacle mesh M , the city’s viable airspace can be divided into two volumetric sets F_{HDR} and F_{UB} . F_{HDR} can be defined as High density routes (HDR) airspace where all missions connecting different city blocks will have to navigate to comply with the flight regulations. This is illustrated in Fig. 6a, and it is the volume mostly aligning with the city major road network starting from minimum flight altitude (β_{min}) up to maximum flight altitude (β_{max}). This volume is obstacle free with a minimum clearance distance of (δ_o) from the nearest obstacle. F_{HDR} is further discretized in “[Robust Skyroutes algorithm](#)” into a hybrid model between Layers, zones, and tubes.

In comparison, F_{UB} can be defined as Urban block (UB) airspace illustrated in Fig. 6b. It is the air volume above buildings aligning with city urban blocks specifically between major roadways. The airspace starts from a minimum clearance distance of (δ_o) from the obstacles (buildings and others). Similar to F_{HDR} , it extends up to the maximum flight altitude (β_{max}). Origins and destinations without major road access will only have access to the airspace through the air volume above these blocks to access the F_{HDR} network.

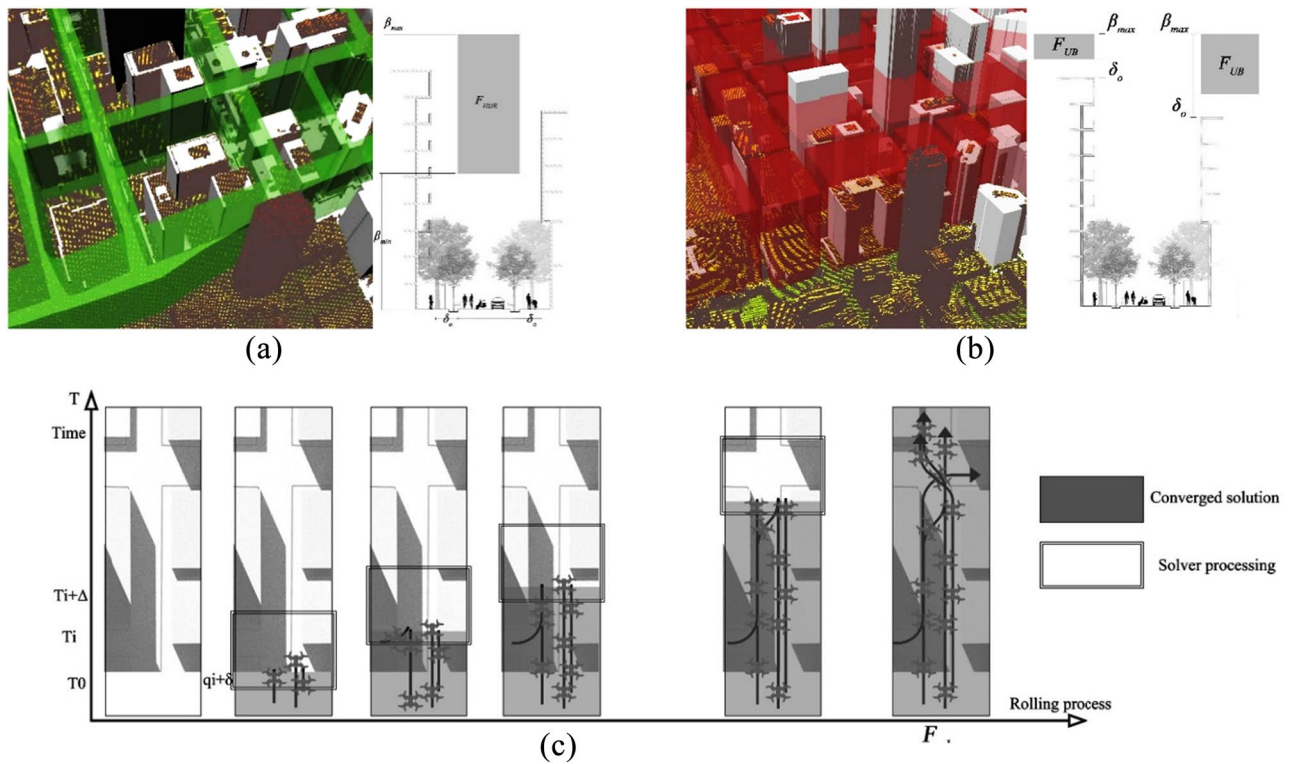


Figure 6. Proposed discretization morphology. (a) F_{HDR} High density routes airspace. (b) F_{UB} Urban block (UB) airspace. (c) Rolling horizon airspace discretization framework.

Given the size of the discretized model and complexity of the details, especially the number of geometrical intersecting features, computational complexity grows quickly with the number of obstacle patches ∂O and the timestep t . The greatest challenge of all is the enforcement of the geofence constraints simultaneously with the UAV constraints, to ensure safe operations. These constraints also become increasingly difficult to process as the 3D urban model consumes the memory allocation, and the UAV mission demand increases, thereby creating more potential conflicts. To reduce the computational complexity and maximize memory usage, we adopt a commonly used strategy to dissect the problem through a rolling horizon framework (Fig. 6c). Rolling horizon has been applied to solve a variety of time-dependent optimization problems in aerial transport such as aircraft scheduling⁸³.

Instead of discretizing the entire city obstacle mesh M , we divide the set into a series of subproblems, each defined by an initial coordinates $q_o \in F$ and a rolling processing window $\{q_i, q_{i+\Delta}\}$ where $[\Delta \ll F_{edge}]$. We ensure overlapping in the solution by reiterating the last section $\{q_i, q_{i+\eta}\}$ where $[\eta \ll \Delta]$ after a subproblem converges. This overlap reduces the possibility of redundant or invalid solutions and guarantees accounting for all obstacles. The rolling horizon method is illustrated in Fig. 6c, where solid rectangles represent the ongoing discretization process at current timestep, and grey zones highlight the converged solutions saved in the memory.

Given the F_{HDR} cross section at J_o such as in Fig. 6a,b, we construct a polygonal vertical surface and contour it horizontally and vertically to dimensions \hat{A} and \hat{B} respectively, where $\{\hat{A} \leq (\text{Street width}) - (2 \times \delta_o)\}$ and $\{\hat{B} \leq \beta_{max} - \beta_{min}\}$. (\hat{A}) will determine the maximum allocation of UAV lanes horizontally, and (\hat{B}) will determine the maximum allocation of lanes vertically. Equations (11–13) for contouring are illustrated in the appendix. We can utilize the maximum area of each inscribed polygon for UAV lanes. This ensures maximum capacity and avoids the formation of bottle necks, which will require further lane traffic management and will decrease the traveling speed and energy utilization.

Whether the payload is confined in the UAV frame or suspended by a wire, during the UAV motion around the pitch, roll, and yaw angles, the payload will swing with motion, especially with aggressive maneuvers. It is crucial to reduce the payload oscillation to avoid damage and guarantee safe operation. Hence, we design the UAV keep-in geofence to account for the payload motion as illustrated in Fig. 7.

In cross section, the UAV lane can be considered a circle with radius r . Assuming an F_{HDR} airspace volume with dimensions \hat{A} and \hat{B} starting from cross section J_o to J_r , we can consider the horizontal lanes as lofted cylinders. Hence, a typical cylinder packing problem is used. In 2D, this problem is equivalent to the circle packing problem where the aim is to maximize the airspace capacity of lanes (circles) while maintaining the minimum radius r such as in Birgin et al.⁸⁴. To maximize the lane airspace capacity, given γ lanes of radius r and a polygonal F_{HDR} airspace, we utilize a nonlinear optimization model to solve the problem as follows:

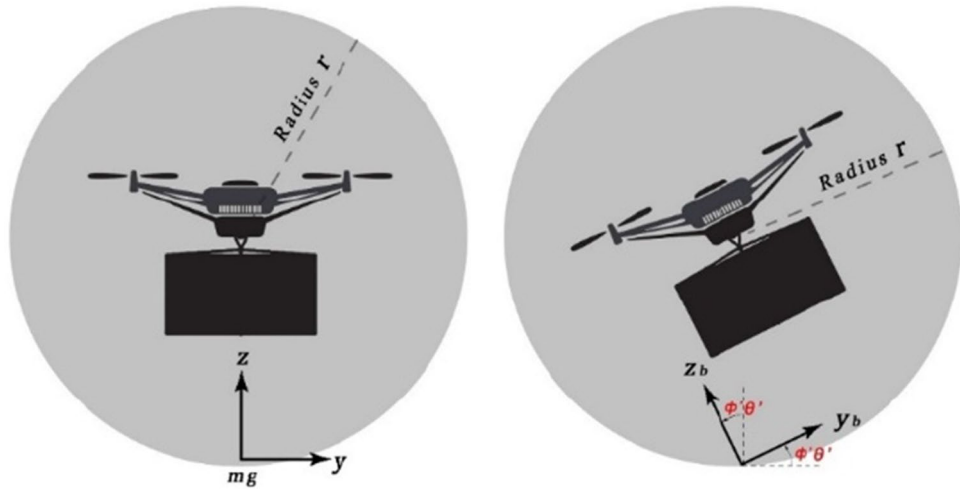


Figure 7. Payload motion within circular keep-in geofence.

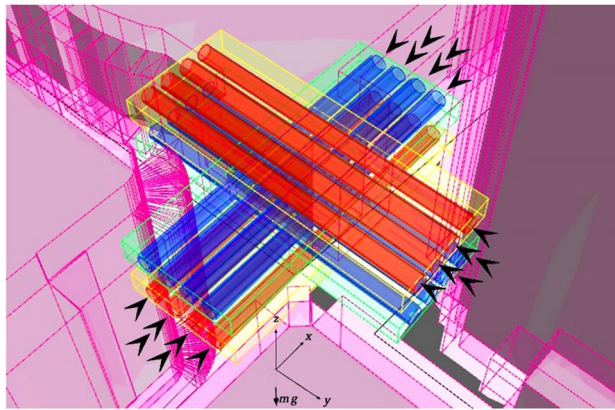


Figure 8. Proposed hybrid layered, zonal, and tubed discretization.

$$\min \sum_{\tilde{i}=1}^{\gamma} \sum_{\tilde{j}>\tilde{i}}^{\gamma} \max(0, 4r^2 - d_{ij}^2)^2 \tag{7}$$

$$\begin{aligned} s.t. & r \leq x_{\tilde{i}} \leq \hat{A} - r, \quad \tilde{i} = 1, \dots, \gamma, \\ & r \leq y_{\tilde{i}} \leq \hat{B} - r, \quad \tilde{i} = 1, \dots, \gamma. \end{aligned} \tag{8}$$

By equating the objective function to zero, if the lanes fit in the cross section, the solver terminates and inscribes the circles.

Given the centers of circles in (11–15), UAVs start at j ; and j_k is the destination. We can extrapolate the keep-in geofence volumetric tubes with vector flight velocity \tilde{v} . Where $q = (x, y, z)$ is defined as the initial location coordinates for UAV aligned with the center of circular keep-in geofence number (i) within the cartesian referencing system. The orthogonal geofence grid of UAV pathways (lanes) are modeled by scaling up Peng et al. Algorithm⁸⁵:

$$\tilde{v}(q) = \left(\frac{\tilde{v}(x - x_k)}{\sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2}}, \frac{\tilde{v}(y - y_k)}{\sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2}}, \frac{\tilde{v}(z - z_k)}{\sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2}} \right)^{Tr} \tag{9}$$

At flight velocities over 3 m/s, translational lift increases the power efficiency significantly. While the speed profile will vary based on the path geometry and the status of the UAV (loaded or unloaded). To achieve the best energy efficiency, constant \tilde{v} speeds are maintained above 10 m/s and below 20 m/s to maintain the viable route while capitalizing battery utilization. It is worth noting that UAV energy limitations will still apply based on battery capacity.

Figure 8 shows the proposed morphology combining layered, zonal, and tubed discretization. For each flight bearing (eastbound, westbound, northbound, and southbound) the lanes are superimposed (layered) for two objectives; (1) avoid potential intersection, (2) allow empty space above and below the keep-in geofence for lane merging on left and right turns. The layers are shown in yellow and green depending on the flight direction. Furthermore, the tubes (circular lane keep-in geofence) are represented in blue and red depending on the vector of flight direction. The arrows in Fig. 8. represent the heading of vector lane velocity \tilde{v} which is organized to allow slower \bar{v} speeds on the rightmost and leftmost lanes and highest \bar{v} speeds towards the middle. The zones represented in magenta are the individual property buffer acting as ‘ramps’ for UAVs taking off/landing from the street-level or balconies/ terraces. These zones are NFZs except for authorized UAVs.

While the proposed framework can function at this level efficiently, “Robust Skyroutes algorithm” illustrates the geometrical modification based on UAV kinematics, which is essential with each digital twin model update and in case of disruption or complex geometrical street grids with obstacle protrusions.

Robust Skyroutes algorithm

To model the lane disruption, we develop based on disturbed flow to generate sky lanes by scaling up the Peng et al. VF-based novel algorithm⁸⁵ and the interfered fluid dynamical system⁷², we describe obstacles as attractive fields through a function. Obstacles in mesh M registered after the smoothing process in (9) can be expressed here as a finite set of welded simplified volumes in cartesian planning space (x, y, z) , each with a geometrical center at $q_{obs} = (x_{obs}, y_{obs}, z_{obs})$, and axial dimensions $(x_{\delta}, y_{\delta}, z_{\delta})$. The obstacle function becomes:

$$F(q) = \left(\frac{x - x_{obs}}{x_{\delta}}\right)^2 + \left(\frac{y - y_{obs}}{y_{\delta}}\right)^2 + \left(\frac{z - z_{obs}}{z_{\delta}}\right)^2 \tag{10}$$

where $q = (x, y, z)$ is defined as the UAV inertial frame location coordinates within the point cloud referencing system. While the proposed method is based on the artificial potential field (APF) method by⁸⁶ in modeling the disruption, however, the proposed method is more robust with a single solution rather than a local optimum. The disruption function $D(q)$ and the modified vector flight velocity \bar{v} at any timestep can be determined utilizing the effective matrix of obstacles (B_{obs}) in obstacle boundary set (u ; where $u \in \partial O$) impacting the UAV lanes as follows:

$$\bar{v}(q) = \tilde{v}(q)D(q) \tag{11}$$

$$\bar{v}(q) = \tilde{v}(q)D(q) \tag{12}$$

$$D(q) = B_{obs} - \frac{\begin{bmatrix} \frac{\partial F(q)}{\partial x} & \frac{\partial F(q)}{\partial y} & \frac{\partial F(q)}{\partial z} \end{bmatrix}^{Tr} \cdot \begin{bmatrix} \frac{\partial F(q)}{\partial x} & \frac{\partial F(q)}{\partial y} & \frac{\partial F(q)}{\partial z} \end{bmatrix}}{\left|F(q)\right|^{\delta_V \exp\left(1 - \frac{1}{\sqrt{(x_u-x)^2+(y_u-y)^2+(z_u-z)^2} \cdot \sqrt{(x-x_k)^2+(y-y_k)^2+(z-z_k)^2}}\right)} \cdot \begin{bmatrix} \frac{\partial F(q)}{\partial x} & \frac{\partial F(q)}{\partial y} & \frac{\partial F(q)}{\partial z} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial F(q)}{\partial x} & \frac{\partial F(q)}{\partial y} & \frac{\partial F(q)}{\partial z} \end{bmatrix}^{Tr}}$$

$$+ \frac{\begin{bmatrix} \frac{\partial F(q)}{\partial y} & -\frac{\partial F(q)}{\partial x} & 0 \end{bmatrix}^{Tr} \cdot \begin{bmatrix} \frac{\partial F(q)}{\partial x} & \frac{\partial F(q)}{\partial y} & \frac{\partial F(q)}{\partial z} \end{bmatrix}}{\left|F(q)\right|^{\delta_H \exp\left(1 - \frac{1}{\sqrt{(x_u-x)^2+(y_u-y)^2+(z_u-z)^2} \cdot \sqrt{(x-x_k)^2+(y-y_k)^2+(z-z_k)^2}}\right)} \cdot \left\| \begin{bmatrix} \frac{\partial F(q)}{\partial y} & -\frac{\partial F(q)}{\partial x} & 0 \end{bmatrix}^{Tr} \right\| \cdot \left\| \begin{bmatrix} \frac{\partial F(q)}{\partial x} & \frac{\partial F(q)}{\partial y} & \frac{\partial F(q)}{\partial z} \end{bmatrix}^{Tr} \right\|}$$

where the lane trajectory angle to the obstacle is denoted by δ_V for the vertical policy parameter and δ_H the horizontal policy parameter.

Lemma 1 Assuming the perpendicular and tangential vectors form a right angle, and $p(q)^{Tr} \cdot \bar{v}(q) = 0$. It indicates that the trajectory lanes can avoid obstacles $[B_{obs}]$ within legally allowed tolerances $[\delta_{min}]$.

Lemma 2 $\bar{v}(q) \cdot \tilde{v}(q)^{Tr} \geq 0$, which indicates that the trajectory can successfully reach the segment destination $[J_k]$.

Lemma 3 $\bar{v} \propto \delta_V$ It indicates that the magnitude of the repulsive and tangential trajectory velocity is directly proportional to the lane proximity horizontal and vertical policy parameters. i.e., following the edge of the boundary of effective matrix of obstacles B_{obs} precisely is inversely proportional to the proximity of the lane to the avoided obstacles.

Theorem 1 If Lemma 1 is satisfied, Lemma 2 is satisfied, and Lemma 3 is satisfied for any obstacle set in mesh M , we can guarantee the feasibility of UAV traffic lanes and trajectories.

L.1. Proof Suppose the perpendicular vector the UAV trajectory is $p(q)$; and the tangential vector to the UAV trajectory $t(q)$ at point q_i on the surface of a single obstacle within mesh M are perpendicular the following from Eq. (17) and (18) stands true:

$$F(q) = 1 \tag{13}$$

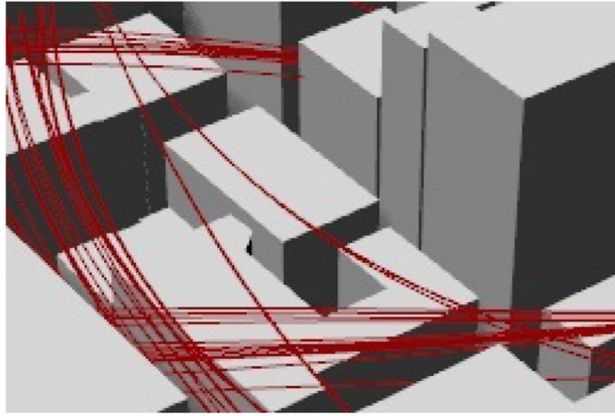


Figure 9. Lemma 1 trajectory avoiding concave obstacle trap areas [B_{obs}] within legally allowed tolerances.

$$D(q) = B_{obs} - \frac{\left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right]^T \cdot \left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right]}{\left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right] \cdot \left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right]^T} + \frac{\left[\frac{\partial F(q)}{\partial y}, -\frac{\partial F(q)}{\partial x}, 0 \right]^T \cdot \left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right]}{\left\| \left[\frac{\partial F(q)}{\partial y}, -\frac{\partial F(q)}{\partial x}, 0 \right]^T \right\| \cdot \left\| \left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right]^T \right\|} \quad (14)$$

$$\text{Therefore, } p(q)^{Tr} \cdot \bar{v}(q) = p(q)^{Tr} \cdot \tilde{v}(q) \cdot D(q) = \tilde{v}(q) \cdot \left(p(q)^{Tr} - p(q)^{Tr} + \frac{p(q)^{Tr} \cdot t(q) \cdot p(q)^{Tr}}{\|t(q)\| \cdot \|p(q)\|} \right) = 0 \quad (15)$$

From (22) we can deduce that with the absence of a value for the perpendicular component, the UAV trajectory using the artificial potential field generated path will not intersect with the obstacle mesh. Figure 9 shows the generated trajectory avoiding a concave tight obstacle trap area.

L.2. Proof Given the mission’s distance between takeoff and landing (J and J_k) is relatively short, theoretically $\bar{v} \approx \tilde{v}$, applying these yields

$$\begin{aligned} \bar{v}(q) \cdot \tilde{v}(q)^{Tr} &= \tilde{v}(q)^{Tr} \cdot \tilde{v}(q) \cdot D(q) \\ &= \|\tilde{v}(q)\|^2 \\ &= \left(1 - \frac{(\cos \alpha)^2}{|F(q)|^{\delta_V \exp \left(1 - \frac{1}{\sqrt{(x_u-x)^2 + (y_u-y)^2 + (z_u-z)^2} \cdot \sqrt{(x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2}} \right)^{-1}} \right)^{-1} + \\ &\quad \frac{\left[\frac{\partial F(q)}{\partial y}, -\frac{\partial F(q)}{\partial x}, 0 \right]^T \cdot \left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right]}{|F(q)|^{\delta_H \exp \left(1 - \frac{1}{\sqrt{(x_u-x)^2 + (y_u-y)^2 + (z_u-z)^2} \cdot \sqrt{(x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2}} \right)^{-1}} \cdot \left\| \left[\frac{\partial F(q)}{\partial y}, -\frac{\partial F(q)}{\partial x}, 0 \right]^T \right\| \cdot \left\| \left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right]^T \right\|} \end{aligned} \quad (16)$$

Taking α is the deviation angle between the vector flight velocity v_i and the obstacle perpendicular vector to the UAV trajectory. While $F(q) \geq 1$ and $(\cos \alpha)^2 \leq 1$ we can deduct:

$$\frac{1 - (\cos \alpha)^2}{|F(q)|^{\delta_V \exp \left(1 - \frac{1}{\sqrt{(x_u-x)^2 + (y_u-y)^2 + (z_u-z)^2} \cdot \sqrt{(x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2}} \right)^{-1}} \geq 0; \quad (17)$$

$$\left[\frac{\partial F(q)}{\partial y}, -\frac{\partial F(q)}{\partial x}, 0 \right]^T \cdot \left[\frac{\partial F(q)}{\partial x}, \frac{\partial F(q)}{\partial y}, \frac{\partial F(q)}{\partial z} \right] \geq 0 \quad (18)$$

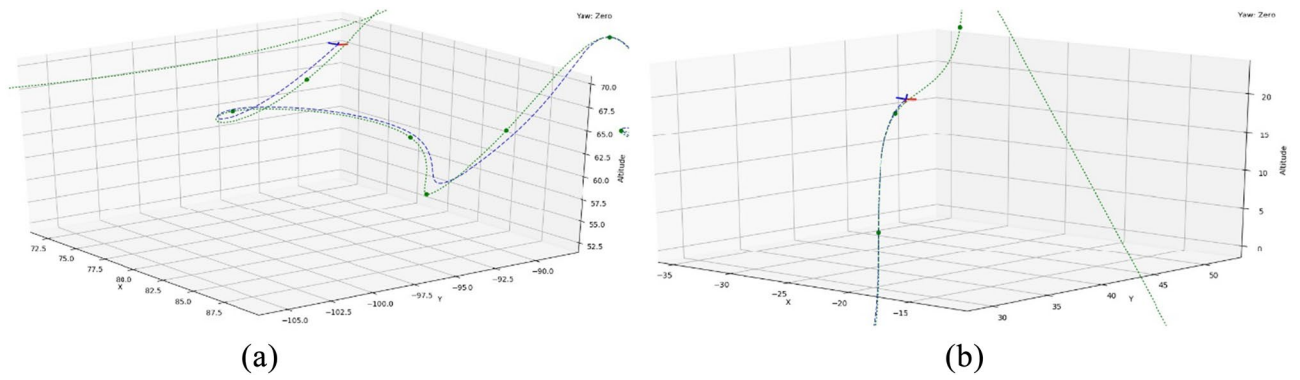


Figure 10. (a) UAV failing to maintain lane trajectory. (b) UAV following lane trajectory.

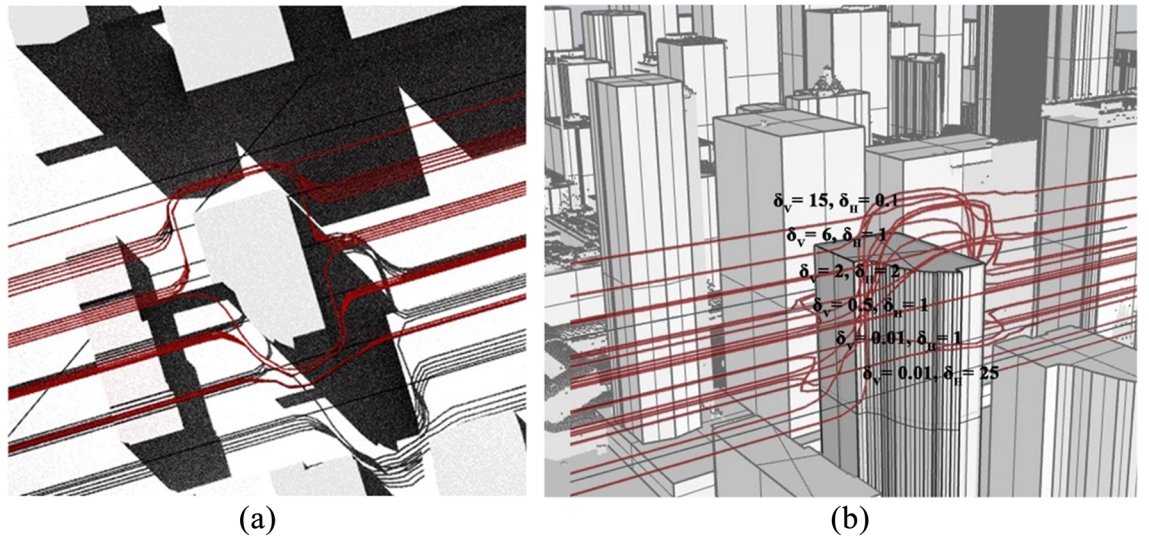


Figure 11. (a) UAV lane trajectory options. (b) Different combinations of δ_v and δ_H .

$$\therefore \bar{v}(q) \cdot \tilde{v}(q)^{Tr} \geq 0 \tag{19}$$

From (26) we can deduce that the UAV trajectory will successfully reach the segment destination $[J_k]$.
L.3. Proof. The vector flight velocity \bar{v} at any timestep which was explained in (19) can be rewritten as:

$$\bar{v}(q) = \tilde{v}(q)D(q) \tag{20}$$

$$\tilde{v}(q) = \tilde{v}(q) - \frac{p(q)^{Tr} \cdot \tilde{v}(q)}{|F(q)|^{\delta_v(q)^{-1}} \cdot p(q)^{Tr} \cdot p(q)} p(q) + \frac{p(q)^{Tr} \cdot \tilde{v}(q)}{|F(q)|^{\delta_H(q)^{-1}} \cdot \|t(q)\| \cdot \|p(q)\|} t(q) \tag{20}$$

Similar to the disruption function in (20) three terms are described; $\tilde{v}(q)$ can be explained as the attractive velocity given by the maximum UAV allowed speed by the applicable civil flight regulation policy; second term $[\frac{p(q)^{Tr} \cdot \tilde{v}(q)}{|F(q)|^{\delta_v(q)^{-1}} \cdot p(q)^{Tr} \cdot p(q)} p(q)]$ is the APF repulsive velocity; $[\frac{p(q)^{Tr} \cdot \tilde{v}(q)}{|F(q)|^{\delta_H(q)^{-1}} \cdot \|t(q)\| \cdot \|p(q)\|} t(q)]$ is the APF tangential velocity.

The concept behind controlling the trajectory is to avoid off-shooting and reduce the risk factor (ξ), this is defined as the possibility of UAV derailing from the designated lane or trajectory, hence risking potential

collision or traffic disruption. Figure 10 shows a UAV failing to maintain trajectory due to the path infeasibility or kinematic incompatibility.

To achieve the maximum speed on a feasible path while ensuring a consistent keep-in geofence, either the speed is reduced leading to time/energy consumption inefficiencies³⁰, or the trajectory is modified. Different combinations of δ_V and δ_H are tested such as in Fig. 11. As illustrated, the produced trajectory shows that the magnitude of the repulsive and tangential trajectory velocity is directly proportional to the lane proximity horizontal and vertical policy parameters.

Based on the numerical formula, an algorithm is developed for trajectory propagation dynamics. This algorithm is capable of solving both the discretization and trajectory planning problems in a three-dimensional environment. The algorithm has a unique two-step procedure. The process of the algorithm can be explained as follows:

- Initialization and loop 1: the algorithm starts by setting up the environment, including the grid, obstacles, origins, destination points (O–D), and other parameters necessary for trajectory planning. This stage aims at adjusting a 3D solid mesh for obstacles using freeform object reconstruction, tangent calculation, and offset processing to generate the mesh of obstacles with the keep-out geofence based on the updated digital twin and flight policies.
- Main loop: the core of the algorithm involves iterating through potential paths and dynamically adjusting them based on the evolving environment and obstacle positions by modifying theoretic trajectories ('sky routes' at policy-allowed elevations) with expected flight velocity $\bar{v}(q)$ at every time step, which is adjusted according to the presence of obstacles. This is achieved using a reformation disruption matrix; vertical and horizontal tangential deformations are imposed on the matrix "D (q)" and considering the UAV's maximum allowed speed, the repulsive velocity from obstacles, and the tangential velocity to navigate around obstacles. This ensures that the UAV's path avoids collisions and maintains a safe and viable trajectory.

The algorithm in pseudocode is presented as follows:

```

1: Initialize: function Skyroutes (Grid, Obstacles, O-D Matrix) // for all missions
2: Input:  $O, \partial O, q_{int}, q_{end}, Q, \forall \hat{P} \subset \partial O, \delta_V, \delta_H, \beta_{min}, \delta_o$ 
3: Initialization input // Destination location as destination vertex, input qend vertex in point cloud.
4: Initialization input // assigned UAV location as initial vertex, input qint vertex in point cloud.
5: Initialization input // Obstacles O with 3D boundary  $\partial O$ 
6: Output: Solution obtained by SKYROUTES:  $\bar{v}(q)$  *
7: Initialization
8: Obtain  $\vec{N}_{\partial O}(p)$  and  $\vec{F}(q)$  and  $\vec{V}(q)$  and  $\Delta\chi$  by solving the Freeform objects reconstruction using equations (1–4).
9: Set  $t = 1$ .
10: Loop 1
11: While  $t \leq T - \Delta; j \leq j_k$  // updating the digital twin utilizing new LiDAR dataset D.
12:   for each polygon in obstacle set O do
13:      $O^N =$  (clone origin face x, y, new elevation Z) // Construct faces on elevation Z.
14:      $\vec{O}^T = \vec{O}^T w_{min} + \delta_o - \delta_{min}$  // Calculate Tangent
15:      $\hat{v}' = \hat{v} + \vec{O}^T (w_{min} - \hat{v}) + \delta_o - \delta_{min}$  // Process offset
16:   end for
17:   for each  $v, \forall v_h \in e$  do //apply relaxation force for Mesh M
18:      $v' = \frac{1}{w} \sum_h v_h$  // Step 1 smoothing function
19:      $v'' = \hat{b} + \delta_o \frac{v - \hat{b}}{\|v - \hat{b}\|}$  // Step 2 smoothing function
20:   end for
21:   Set final state for Mesh M.
22:   for each  $v_h^*$  in M  $\forall v_h^* \in \partial M^*$  // apply transformation using a weighted least-squares function.
23:     if  $d_i \triangleq d(v_h^*, v_h) \geq \delta_{max}$ 
24:        $min_{(Q, \vec{T}) \sum_h \omega_h \|M_h - (QM^*_h + \vec{T})\|^2}$ 
25:     else
26:       if  $O_{new} \neq O \forall O_{new} \in F_{HDR}, \partial O_{new}$ 
27:          $\partial O_{new} \in O$ ; apply process in (1-8) to obtain final state for Mesh M.
28:       end if
29:     end if
30:     Use final state for Mesh M.
31:   end for
32: end while
33: End Loop 1
34: Main Loop
35:   for each  $J_i; \forall J_o \leq J_i \leq J_k$  do // construct a polygonal vertical surface  $\hat{A} \cdot \hat{B}$ 
36:      $min \sum_{i=1}^y \sum_{j>i}^y \max(0, 4r^2 - d_{ij}^2) \forall r \leq x_i \leq A - r; r \leq y_i \leq B - r$ 
37:     if  $q_{int}$  and/ or  $q_{end} \notin F_{HDR}; q_{int}$  and/ or  $q_{end} \in F_{UB}$  do set  $q_{io} = q_{int}, q_{ik} = q_{end}$  // determine nearest points  $q_o$  and  $q_k$  on generated routes corresponding to each trip function and UAV size to the origin and destination locations  $q_{io}, q_{ik}$ 
38:     Initiate RRT sequence  $(q_{io}, q_o); (q_{ik}, q_k)$ ; T.add  $(q_{io}, q_{ik}); q_n \leftarrow q_{io}; q_{ik}$ 
39:     While (DISTANCE  $(q_n, q_o); (q_n, q_k) > d_{limit}$ ) do // Limit d to remaining UAV charge.
40:        $q_k =$  TRAND  $q_{rand}()$  // nearest vertex within the tree T
41:        $q_{near} =$  T.NEAREST  $(q_i)$ 
42:        $q_n =$  EXTEND  $(q_{near}, q_k, expansion\_time)$  // connect segment
43:       If  $q_n \neq$  NULL
44:          $q_n.setParent(q_{near})$ 
45:         T.add  $(q_n)$ 
46:       end if
47:     return trajectory for first and last pairs  $(q_{io}, q_o); (q_{ik}, q_k)$  step 49 to add
48:   end while
49:   end if
50:   Compute modified  $\bar{v}(q)$  based on D  $(q), \delta_V, \delta_H \forall q_o, q_k = (x, y, z), q_i \in Q$  // calculate trajectories for all assigned UAVs,  $\delta_V, \delta_H$  are determined for each lane based on speed and geometry in sec 6.
51:   end for
52:   Output computed Skyroutes and RRT trajectories to assigned UAV trips by order of queue.
53: End Function.

```

Algorithm 1. Pseudocode for the Robust Skyroutes Algorithm

Trip generation, Cartesian routing, and UAV energy consumption

To test the operability and assess the efficiency of the proposed algorithm, a high-traffic load operation duration has to be simulated. An urban transportation simulation requires access to the specific location demand data. However, real-life georeferenced demand data is protected under different privacy laws. In this study, we model the origin and destination trips by adopting a realistic approximation from statistical prediction models that have been used in trip generation models and proved a high level of accuracy and robustness²².

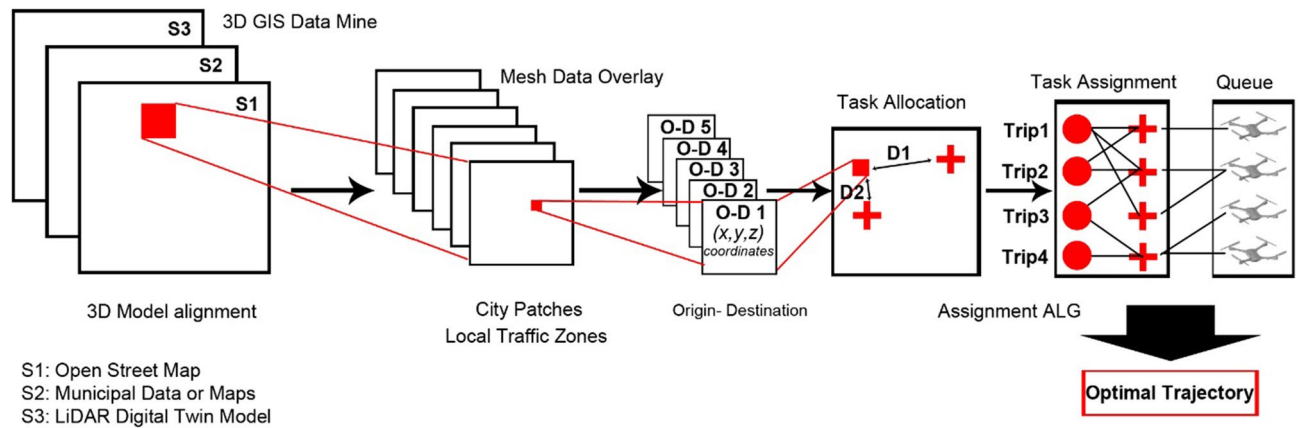


Figure 12. Proposed 3D GIS mining framework.

To generate a heterogeneous trip generation in terms of UAV size and trip nature (package delivery, flying taxi, or ambulance), we assume that the model follows a Poisson-distribution. The Poisson distribution is commonly used in various transportation demand modelling since it is considered an activity that will occur at a constant rate over a duration of time⁸⁷. The mean variation is based on the simulation area census population density. The trip generation Eqs. (28–31) are outlined in the appendix for reference. The density map and the probability generation algorithm based on this Poisson distribution are coded in Python and overlaid on the city digital twin model. The resultant O-D matrix is the base for the TA. UAVs are assumed to start the trip by Vertical Takeoff (VTO) from the roof of the origin building mesh and ending the trip by Vertical Landing (VL) at the destination roof. To link the transportation tasks among generated O-D geolocations on the city digital twin model mesh generated in “[Airspace discretization model description](#)”, an area allocation and UAV assignment planning process is applied, however, TA does not fall within the scope of this study.

To assess the robustness of the proposed algorithm in this study, a single serving coverage area is considered, and each UAV is assigned one trip per timestep. Multiple randomized trip objectives, payload, duration, and travel distance determine the UAV size. The city digital twin is divided into clusters or volumetric patches according to several parameters including urban density and maximum building-footprint area. While for traffic and lane management, a first-come-first-serve queuing protocol is implemented. A full 3D GIS mining framework similar to neural networks is proposed and illustrated in Fig. 12. After the digital twin data is processed, the autonomous UAV trip generation and TA allocation loop is provided with pairs of coordinate points (latitude and longitude) via GPS link. The trips are generated based on the pre-explained Poisson distribution randomly to produce the full range of trips length and route complexity. The *Skyroutes* algorithm routes the trip and blocks the allocated lane segment $[\bar{v}(q)]$ at the utilized timestep T for other UAV trips.

The UAV lane-trajectories resulting from the algorithm depends on δ_V, δ_H values. These values are determined for each lane based on lane designated speed, hence the resulting geometry as discussed by ElSayed and Mohamed^{22,80}. To reach the optimal energy consumption and speed, the UAV motion is simulated based on the quadrotor physics. Mainly, the power divided on rotors that define the way the UAV moves and response, such forces, torque and thrusts are the key to UAV motion.

While UAV flight dynamics differ by airframe type, the main variants are fixed-wing and multi-rotor. Unlike fixed-wing UAVs, a multirotor possesses more than two rotors with hovering capabilities, such as quadrotors and hexarotors. For Quadcopters, a multirotor is controlled by altering the relative speed of each rotor to adjust the thrust and torque produced by each propeller opposing drag vector about the center of rotation. The four propellers are positioned at the corners of a square chassis as a pair of rotating blades. The motion equations are explained in the appendix (Eqs. 32–46).

The calculated energy consumption in Eqs. (35–40) aligns with real-world experimental results given the same input parameters for an experimentally verified model for a loaded quadcopter from the literature in Stolaroff et al.⁸⁸ and ElSayed and Mohamed²². Results illustrated in Fig. 13 show high agreement at lower velocities, with a 5% discrepancy at higher velocities due to discrepancies in model assumptions.

At flight velocities over 3 m/s, translational lift increases the power efficiency significantly. While the speed profile will vary based on the path geometry and the status of the UAV (loaded or unloaded), to achieve the best energy efficiency velocities are maintained above 10 m/s and below 20 m/s in the generated lanes to maintain the viable route while capitalizing battery utilization.

Although the *Skyroutes* algorithm creates the main trajectories, the last trip leg in the local traffic zones F_{UB} operate under a full-mix airspace pattern. Due to the low traffic density, this airspace hardly needs regulation, the cartesian discretization is utilized to find the first/last leg of a trajectory using any of the literature solving algorithms. In this study, we utilize a modified Random Reduction Tree (RRT). A basic RRT works through three functional procedures, firstly the ‘generation’, finds by calculation a path between q_{int} ‘starting vertex’ and q_{end} ‘destination’ vertex which is obtained by growing a random search tree. The tree branches out in a highly dimensional environment to search for possible vertices from the starting vertex towards the destination with bias along the direct connector vector. Secondly, ‘the expansion’, a random vertex q_{rand} is picked and a line segment ‘edge’ is interpolated between the new vertex and last tree vertex in the list. With each iteration, a new edge and

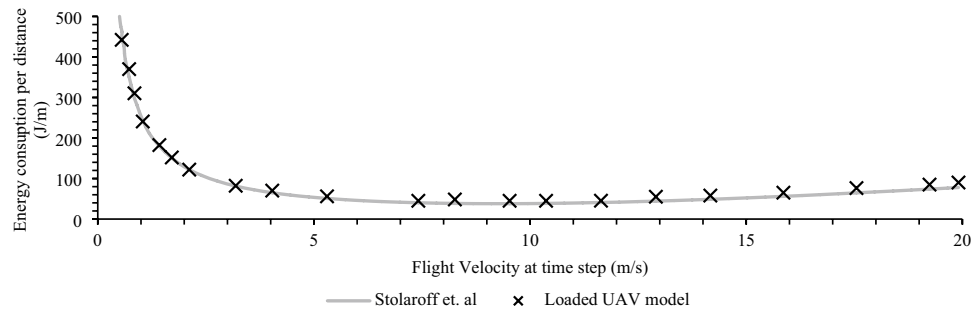


Figure 13. Experimental verification of calculation model.

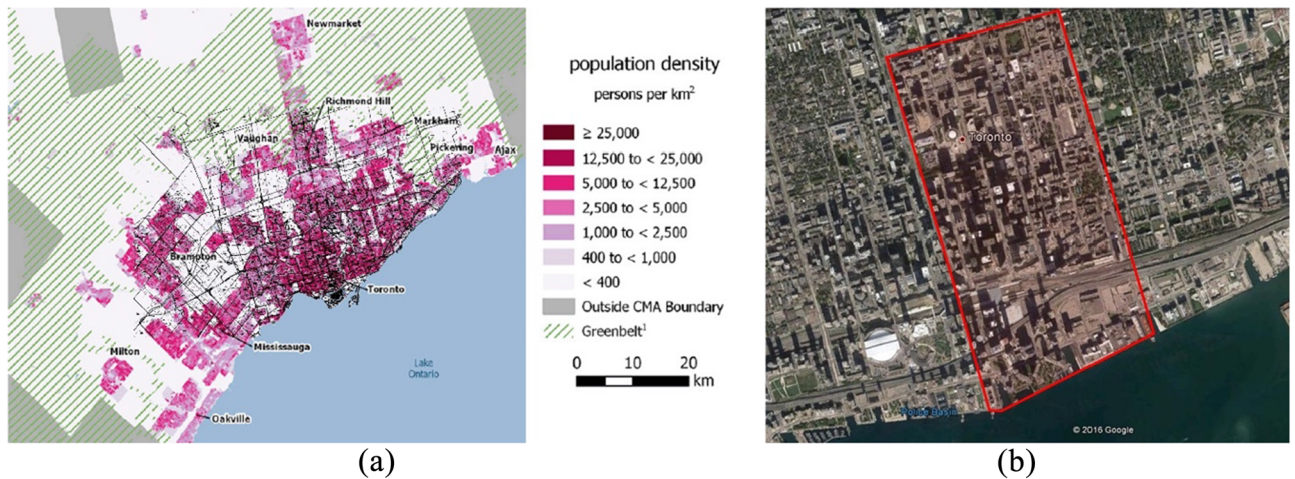


Figure 14. (a) Macro scale GTA census map⁸⁹. (b) Aerial image of the study area (marked in red) and city context⁹⁰. Source for (a): <https://www150.statcan.gc.ca/n1/en/catalogue/16-508-X>; Source for (b): Google maps with a Microsoft word 365 markup edit, google.com/maps.

vertex are added to the path and the tree list expands till the destination vertex becomes a part of the tree. This leads to the third and final process ‘termination condition’.

Although highly successful, this basic calculation method becomes memory consuming. Moreover, the convergence rate is relatively slow in cases of complicated path planning where the chance of collision is significantly high in an obstacle rich environment such as our case study. Utilizing the A-star algorithm approach, in this case, amends this downfall and ensures the solving tree is only considering the most relevant areas of the point cloud tree. Whereas in a typical RRT the whole model space is populated with a point cloud and is considered for the solution. On the other hand, the Astar transforms the search into a function of the range of vertices confined along the direct path between q_{int} and q_{sos} , this becomes the point populated domain, and the function is formulated as follows:

$$\dot{q}_t = f(q_t - 1, \dot{u}_t - 1, v_t) \tag{21}$$

$$D_t = h(q_t, v_t) \tag{22}$$

where $\dot{q}_t \in Q$ is the initiation point vector; $\dot{u}_t \in U$ is the destination vector; v_t is a random process disturbance appropriately determined; D_t is the measurement vector and q_t is a random component of the q_t tree.

Similar to Dijkstra algorithm, the Astar algorithm contains an open list of the potential waypoints q_{free} vertices, in addition to a closed list of all the visited vertices and a simple cost equation for solving as follows:

$$T_i = C_i + E_i \tag{23}$$

where subscript i stands for the vertex call number in the RRT; T_i is the total cost (path length to minimize from q_{int} to q_{end}) similar to Eq. (15); C_i is the current i th cost from q_{int} to current vertex; E_i is the estimated cost of i th vertex from the current vertex to the q_{end} destination vertex. To simplify the solution and solving time, the algorithm is also written and compiled in Python.

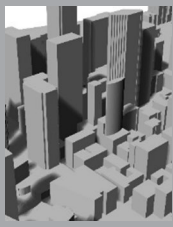
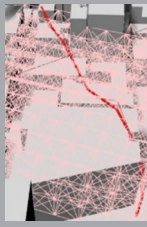
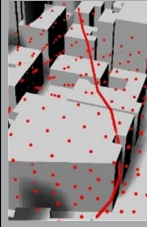
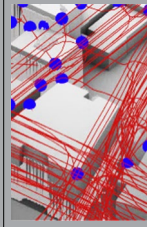
Element	Settings	Algorithm				Unit
		Base model	Dijkstra	Modified RRT	Skyroutes	
1-Digital-Twin	Solution image					
2-Simulation parameters	Solving base	Type of path generation distance between vertices solution time	B-Spline 2 221	Bezier Curves 2 89	Lanes N/A 74	Sec
3-Processing power		Processor: Intel Core I9 CPU with single core utilization of 2.20 GHz; Memory: 64 GB				

Table 3. Comparison of computational performance.

Discretization method	Cartesian and proposed <i>Skyroutes</i>
Service area	3,663,251 m ²
Poisson λ parameter	Six 3-digit allocations
Average trip distance (min, max)	811.26 (24.32, 2059.35) m
Average ED between destinations (spread)	52.1 m
Mission count (trips)	1138
Longest route ED	2059.35 m
AM peak	9–10 a.m.
PM peak	4–6:30 p.m.

Table 4. Results of the O–D trip demand model.

Case study, results, and discussion

A case study of a real 3D urban area in the densest section of the City of Toronto, one of the biggest urban centers in North America, was used to test the model and algorithm. With a population density of 4149.5 p/km², occupied with various commercial, residential, and infrastructure buildings, the area represents a typical example of a mixed-use urban center. The area is covered in clusters 50 and 51, East York Patch, with an approximate area

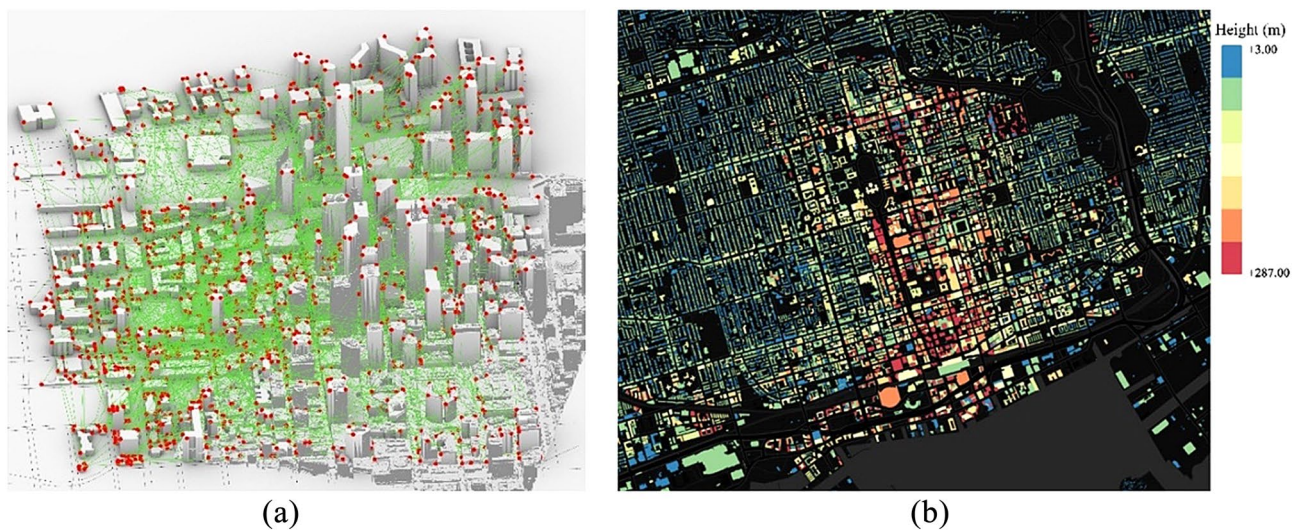
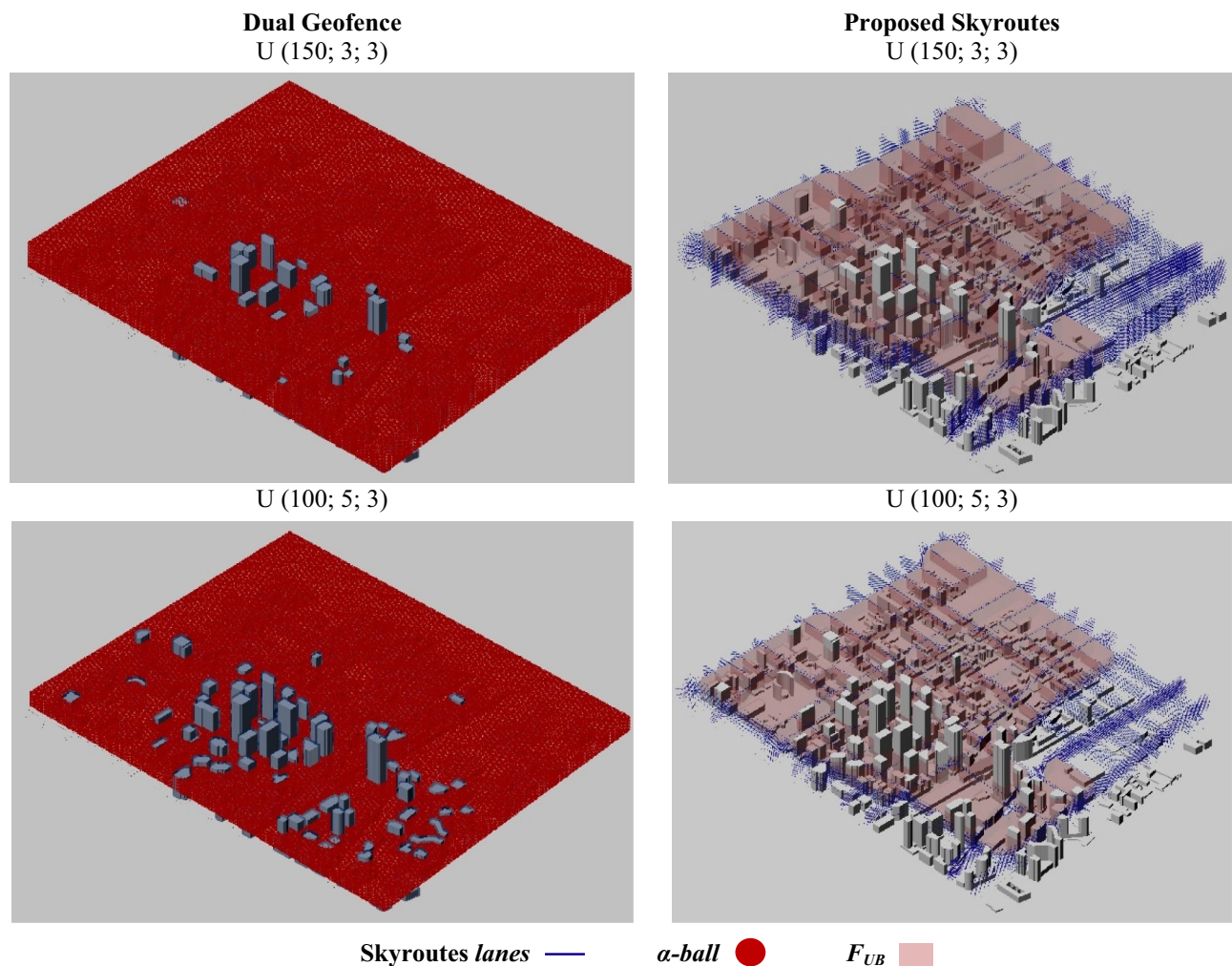


Figure 15. (a) O–D points (in red) ED of peak-hour trips (in green). (b) Study area in old Toronto showing height distribution of structures in airspace. Source for (a): Authors created by Rhinoceros 3D, version 7, rhino3d.com; source for (b): Authors created by Rhinoceros 3D, version 7, rhino3d.com.



Discretization Method	Cartesian	Skyroutes
β_{max} (min, max)	30, 150	30, 150
δ_o (min, max)	1, 21	1, 21
r (min, max)	1, 21	1, 21

Figure 16. (a) Cartesian grid discretization, keep-in (red), Keep-out (blue). (b) Skyroutes discretized airspace, lanes (blue), F_{UB} (red).

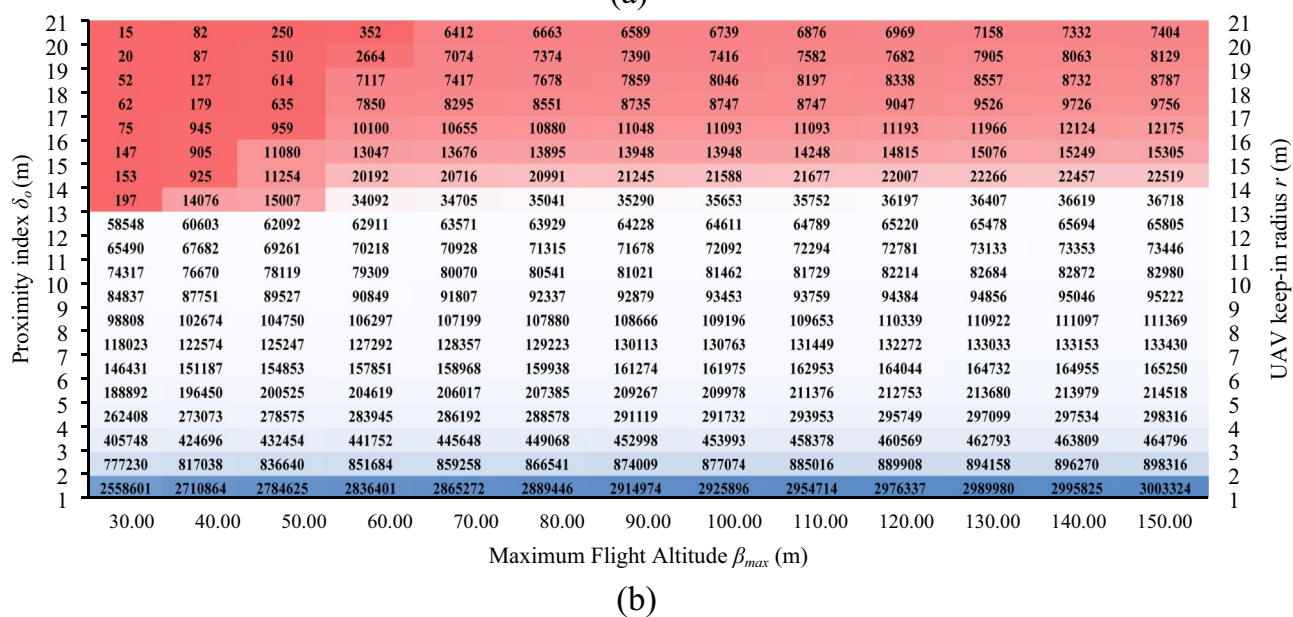
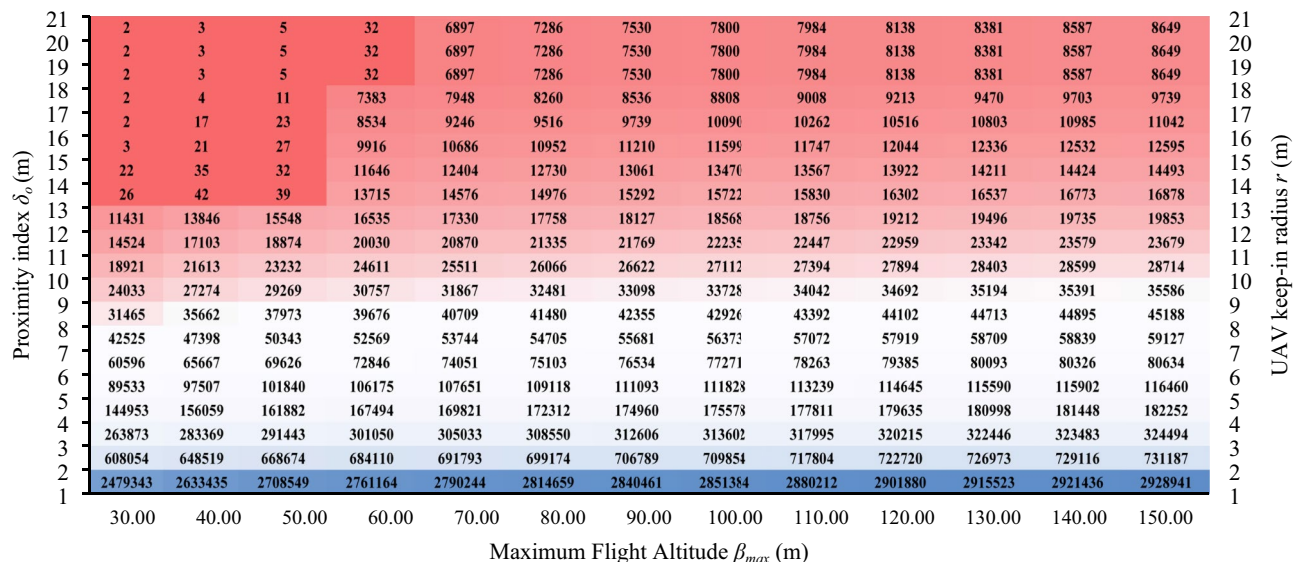
of 3.16 km², presented in Fig. 13b. It features dense high-rise buildings and airfields, which can be complex for other discretization methodologies Fig. 14b.

Based on the Canadian census population density maps (Fig. 13a), a base-case scenario operation model was conducted as outlined in the methodology section on six downtown three-digit postal code areas and the associated geocoded information. Only local trips within the study area are modeled given the UAV range limitations on a single charge roundtrip. Results of the daily O–D Poisson generation are reported in Table 3, while peak-hour (5 p.m.) trips (around 1138 trips) are visualized in Fig. 14a. The ED is shown in green lines for UAV trips.

We carried out the *Skyroutes* discretization and cartesian discretization for performance comparison. In this section, we compare the results between the two schemes on three fronts, namely, Airspace utilization represented by the utilization factor (U) metric; hazard mitigation factor represented by the risk factor (ξ) metric; and Kinematic and energy efficiency represented by the total change in Euler angles (RAD). Considering the computational performance, results in Table 4 show the proposed Algorithm is comparatively similar to the solution time of Rapidly-exploring Random Trees (RRTs) and significantly better compared to Dijkstra. A thorough discussion on the performance has been explained for complex environments with dynamic elements in a previous study⁸¹. We have also illustrated the computational effectiveness by a simple test run against the Gurobi solver demonstrating significantly low solution time⁹¹.

Geofencing results compared to cartesian discretization

In the case study, airspace between 30 and 150 m (100 m for strict regulations) is considered for the UAV traffic. Starting with city obstacle mesh M , the airspace was first divided into the two volumetric sets F_{HDR} and F_{UB}



Discretization Method	Cartesian	Proposed Skyroutes
β_{max} (min, max)	30, 150	30, 150
δ_o (min, max)	1, 21	1, 21
r (min, max)	1, 21	1, 21
Total airspace capacity (min, max, std. deviation)	2, 2928940, 614713	15, 3003324, 631581

Figure 17. Discretization airspace capacity matrices. (a) Cartesian discretization airspace capacity matrix. (b) Proposed Skyroutes discretization airspace capacity matrix.

described in the methodology section. Then, the Skyroutes morphology at $\delta_H = 1$ yielded 40 levels of keep-in lanes flowing along the F_{HDR} shown in Fig. 15a. Similarly, the airspace was discretized into a three-dimensional regular grid of 3 m and 5 m for strict regulations, resulting in a $440 \times 360 \times 40$ α -ball Cartesian grid in a dual geofence Fig. 15b.

To compare the results across both methods, we use a utilization factor $U(\beta_{max}; \delta_o; r)$ for cartesian discretization, where (δ_o) is the minimum clearance distance from the nearest obstacle; and r is the keep-in radius for UAVs; (β_{max}) is the maximum flight altitude dictated by the applicable flight policy. For the Skyroutes algorithm, O-Ds without major road access utilize the F_{UB} for the first/last leg in the trip connecting to the lanes. Figure 15a highlights the results of maximized utilization of airspace with lean flight policies, the α -ball utilization coverage in 3D is $U(100; 5; 3) = 88.1\%$ and $U(150; 3; 3) = 93.1\%$, respectively. This is due to the added airspace volumes

in the cartesian discretization, the same utilization gains are reflected in the Skyroutes algorithm results as the maximum flight altitude β_{max} increases, hence adding more lanes for UAVs.

These results highlight several observations, first, the benefits of using the Mesh M generated from LiDAR point cloud (lines 1 to 33 in Algorithm1) as a more precise tool in airspace capacity estimation as compared to other 2D methods in the literature where capacity is calculated by a series of horizontal 2D slicing planes. Second, the benefits of dual (keep-in and keep-out) geofencing technique allow higher control to apply airspace flight policies and NFZs.

While the airspace utilization increases significantly in both discretization methods with leaner flight policies (5% for cartesian and 10% for Skyroutes), however, the cartesian discretization shows higher sensitivity to the δ_o value, as compared to the Skyroutes method, which relies heavily on the β_{max} . The proposed Skyroutes algorithm shows a higher level of robustness in eliminating the inconsistency of airspace utilization variance with altitude through a linear behaviour compared to an exponential behaviour in the cartesian morphology. By eliminating bottlenecks as UAVs propagate in lower airspace for main trajectories in the F_{HDR} , the exponentially higher estimates of utilization in F_{UB} indicates higher UAV traffic above buildings.

This proves the substantial benefits of using the Skyroutes algorithm over the cartesian method in airspace capacity estimation that further aligns with the AAM civil airspace safety and privacy objectives.

Similarly, Fig. 16 shows the results of airspace capacity utilization and loss in terms of different β_{max} at variable values for δ_o and r. The results' matrix heatmaps show the robustness of the proposed methodology in estimating airspace capacity with extreme precision. Overall, the effect of geofencing was most restrictive in the lower altitude levels, when $\beta_{max} \leq 60$ due to the high-density obstacles. However, the impact on cartesian discretization is significantly more severe compared to the impact on the Skyroutes discretization. Also, generally it can be noticed in all altitudes and across different δ_o and r combinations, the Skyroutes discretization yields a higher airspace capacity. This is due to the advantage of using the cylinder/circle packing subroutine Eqs. (14–15) to fit more lanes as compared to the cartesian division.

Furthermore, due to the island effect in cartesian discretization if an airspace patch has less than 25 m (this is dominant in lower altitudes with higher δ_o and r combinations) of travel range, the entire discretized patch is not considered in the capacity estimation. This is not the case for Skyroutes since the lane discretization is performed in 3D, which sometimes allow only a narrow path in higher altitudes to utilize this entrapped discretized airspace void in lower altitudes. The matrices presented in Fig. 16 can guide policymakers in finding the regulation combinations to achieve a desired level of civil airspace utilization, and to evaluate the operational feasibility based on trade-offs between β_{max} , δ_o , and r.

Airspace utilization and loss matrices prove more efficient and robust in airspace capacity estimation as compared to 2D graphs and curves. Matrices highlight the severe impact of higher δ_o and r combinations ≥ 10 m in lower airspace levels $\beta_{max} \leq 60$. This highlights the sensitivity to altitude in tighter urban scenarios such urban centers and high-rise downtown areas. It also highlights the flexibility of dual geofencing (keep-in and keep-out) in determining the safe airspace utilization. Whereas higher airspace altitude $\beta_{max} \leq 60$ shows a slightly greater advantage to cartesian discretization over Skyroutes, results show a 10% increase in airspace capacity estimation as the free-mix airspace model is applicable. In general, digital-twin volumetric 3D approach shows robust capability to assess airspace capacity with different policy permutations.

Air traffic safety and hazard mitigation performance

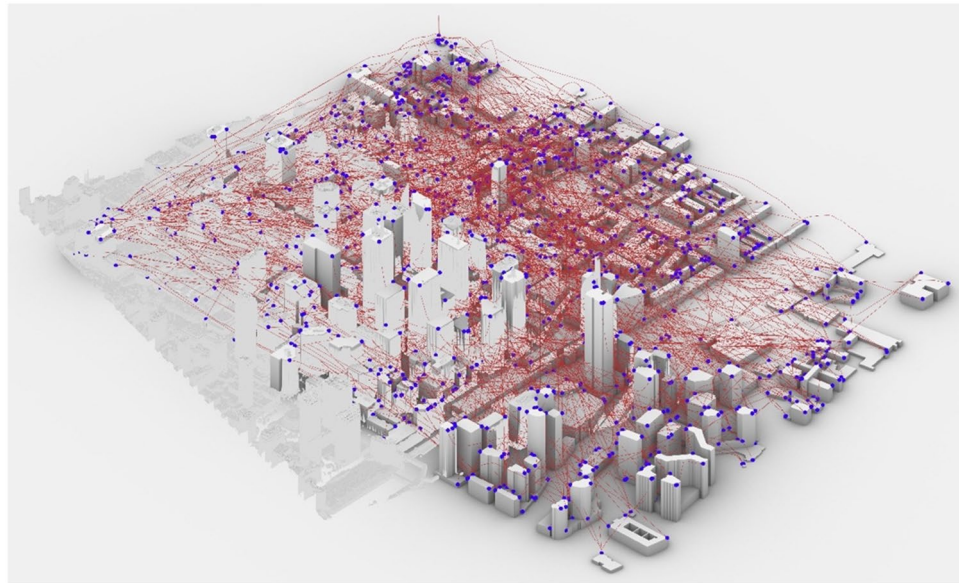
In this section, we present the differences in airspace safety and hazard mitigation between cartesian and Skyroutes discretization. While, noise reduction is illustrated by visualizing the UAV trajectories around the study area, safety is defined by a risk factor (ξ), which is the proximity of the UAV trajectories to moving obstacles and other UAV trajectories or the possibility of the UAV derailing from the designated lane or trajectory. Figure 17 shows Cartesian and Skyroutes discretization airspace UAV trajectories at 5 pm for the study area. To assess the robustness of the proposed algorithm, we utilize the modified RRT* as well as several relevant UAV 3D routing and trajectory optimization literature from Table 2 for each UAV trip and only use the most efficient results for the cartesian method.

The results show several trends, first, the significant difference in noise reduction, as UAV trajectories avoid the utilization of airspace above urban blocks in Skyroutes trajectory optimization versus the cartesian trajectories. This is with the exception of take-off and landing (last leg) performed as part of the TO task and amalgamated to the total given mission trajectory to avoid the outlined accident risk. Second, the Skyroutes results show a significant airspace order as compared to cartesian methods due to the aggressive use of the airspace above urban blocks F_{UHD} to achieve the shortest trajectory possible. The proposed algorithm regulates all trajectories in the F_{HD} volume mostly aligning with the study area's major road network starting from the minimum flight altitude (β_{min}) up to the maximum flight altitude (β_{max}).

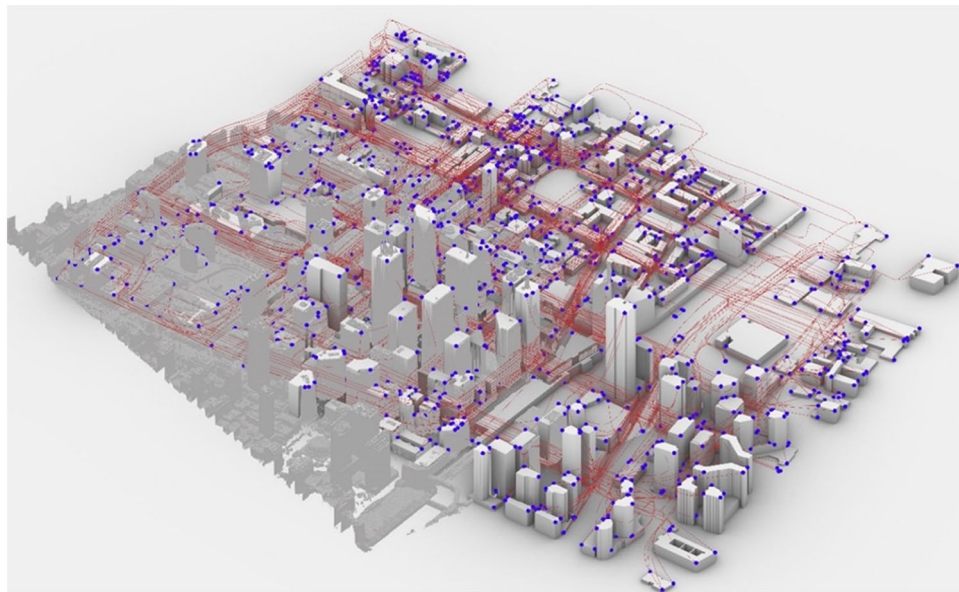
Further, Fig. 18 shows the results of cross trajectory proximity for both discretization methods. Skyroutes algorithm shows a significant reduction in the instances of cross trajectory proximity where trajectories are in closer proximity (distance between trajectories at any point is < 3 m or intersecting) at a critical time window ≤ 30 s. The lane geometrical design and timestep queuing method allows optimizing the trajectories by spacing them whenever possible mitigating multiple trajectory collision. Along the same lines, Fig. 19 shows the significant reduction in the trajectory Euler transformations (explained in Fig. 7) which ensures the integrity of the payload within the keep-in geofence and reducing the risk factor (ξ) of UAV derailing from the designated trajectory.

Kinematic and energy efficiency

The results show up to 30% lengthier trajectories (in 8.2% of the cases), and up to 10% increase for the rest of the trajectories for the Skyroutes discretization as compared to cartesian discretization, Fig. 19. This increase



(a)



(b)

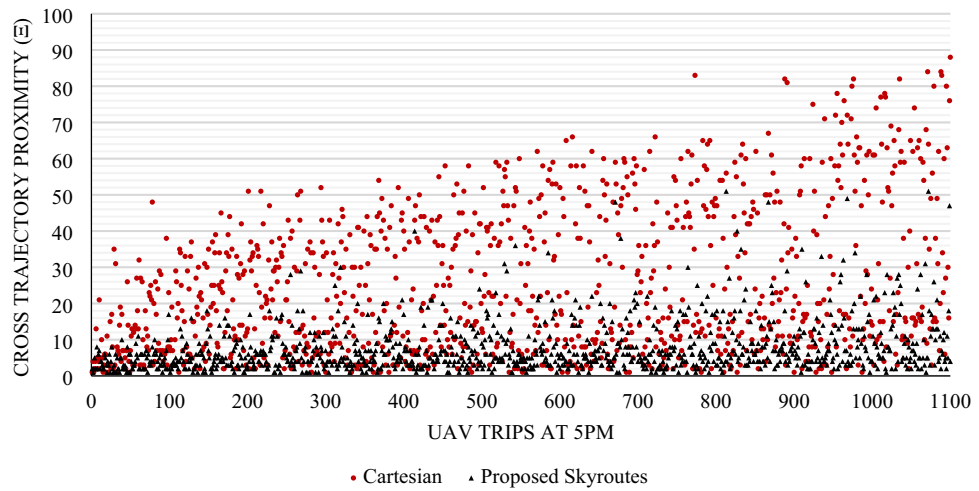
Figure 18. Discretization airspace trip trajectories. (a) Cartesian discretization airspace trip trajectories 5 pm. (b) Proposed Skyroutes discretization airspace trip trajectories 5 pm.

in route length comes from the instances where origins or destinations are deep in the congested areas of the airspace or from the need for multiple lane changes due to the queue. However, the overall energy consumption is up to 50% lower in more than 60% of the trips for the proposed *Skyroutes* discretization and trajectory optimization algorithm.

This is due to the consistency of the trajectory as stretches of straight lines in the keep-in lanes allow UAVs to maintain the maximum efficient speed of 20 m/s without the need for deceleration on maneuvers as in the case with the cartesian trajectories. This is illustrated in Fig. 20 as the total change in Euler angles along the UAV trajectories. Less change in trajectory angular motion means significant reduction in rotor torque changes allowing the UAVs to travel a longer distance at the optimal discharge rate and decreases the depletion of charge⁸⁰.

Conclusions and future studies

In this study, we proposed a novel autonomous Advanced Aerial Mobility (AAM) system for high density city centers that dynamically discretizes the viable airspace into UAV trajectories. By incorporating the city's digital-twin model through interpolating LiDAR data and a dual keep-in and keep-out geofence, our method expands the functionality beyond airspace capacity assessment to test different flight policies and measure the tradeoffs



Discretization Method	Cartesian	Proposed <i>Skyroutes</i>
UAV Trips at 5 pm	1138	1138
Cross trajectory proximity (ξ) (min, max, std. deviation)	1, 92, 21.0	1, 54, 7.2

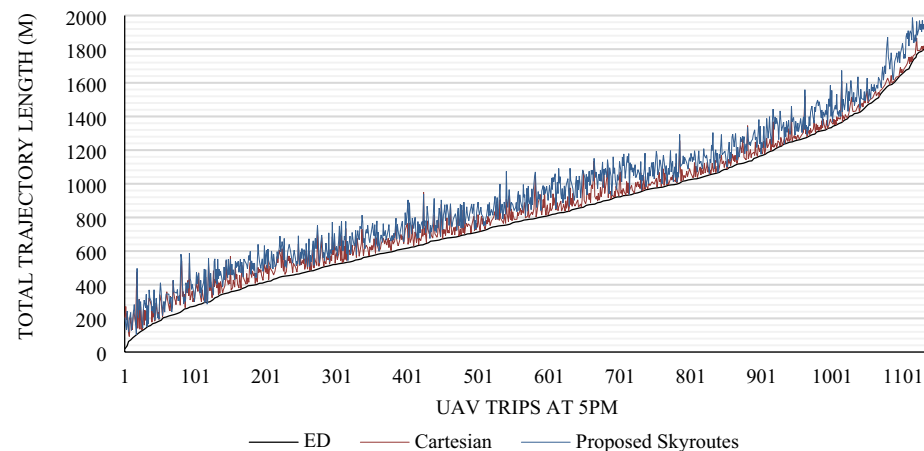
Figure 19. Cross trajectory proximity results.

between them. Furthermore, the proposed algorithm converges energy-efficient UAV trajectories while minimizing the safety hazards and sound pollution.

Since UAVs are assumed to be automatically piloted by an embedded mission control system, in a heterogeneous fleet situation or a multi-user traffic control narrative, the onboard flight controller on each UAV requires a pre-planned trajectory with multiple contingencies (alternative routing) for specific mission assignment and teamwork logistics. This highlights the benefits of the proposed *Skyroutes* with multiple lanes rather than a full-mix airspace morphology.

In the hypothetical case of a complex urban scenario, we demonstrated that the digital-twin model is crucial for the precision and safety of pre-planned UAV trajectories. The proposed *Skyroutes* algorithm was able to identify narrow urban corridors and maximize the airspace capacity up to 10% increase in a severely restricted airspace by connecting isolated airspace volumes through a circle packing sub-routine as compared to cartesian discretization, which was unable to tackle this challenge efficiently. A case study of Toronto city center, Canada illustrated the robust capabilities of the proposed algorithm in a real 3D environment.

The cartesian airspace discretization allows the applicability of a variety of trajectory optimization algorithms in a full-mix airspace morphology, while the *Skyroutes* capitalizes on the energy-efficient trajectories and regulating the airspace traffic management through combining several airspace morphologies. For cartesian discretization, on the one hand, a tight mesh (waypoint vertices) results in a slower and more complicated graph-solving task due to the significantly large size of the solving domain. On the other hand, a wider mesh results in less



Discretization Method	Cartesian	Proposed <i>Skyroutes</i>
Total trajectory length (min, max, std. deviation)	92.1, 1995, 394.1	102.1, 2000.8, 412.9

Figure 20. Total trajectory length results.

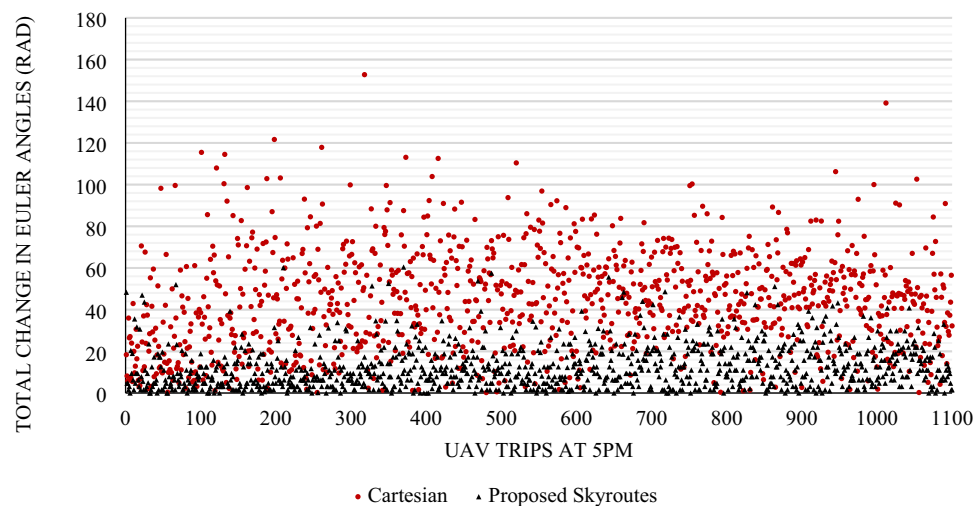


Figure 21. Total change in Euler angles along UAV trajectories results.

available solutions and more unutilized tight spaces within the dense city urban form where spacing between the towers can be less than three meters wide. The application of dynamic meshing method in digital-twin models shows the agility of capturing urban details, where building protrusions, setbacks, construction tools (such as cranes) and other architectural features such as street vegetation and landscape elements within the urban setting are taken into consideration. This allows the solving algorithm to diminish collision chances and relieve the reliance of on-board sensors. Also utilize tight spacing within the study area while avoiding the probability of algorithm's solution errors that could cause obstacle collisions. The *Skyroutes* discretization is more adaptive and can deliver significantly higher airspace usability coupled with more challenging capabilities especially in highly restrictive airspace.

The proposed *Skyroutes* algorithm successfully demonstrated the ability to analyze the flight policy combinations in the case study. The precision in estimating the airspace capacity showed high sensitivity to the variables, which suggests that the current approach that relies on 2D or cartesian discretization measures needs further evaluation for effective urban UAV operations. The proposed algorithm illustrated the difference in safety and energy-efficiency of the converged trip trajectories. The results also show significant improvements over cartesian discretization, the overall energy exerted by UAVs to overcome a lengthier trajectory is outweighed by lower torque changes, lower energy consumption, and lower noise levels avoiding urban airspace over inhabited areas. Furthermore, reduced cross-trajectory proximity and the proposed lane change sub-routine allows higher coordination and safety by providing alternate routing in case of disruptive events.

One of the possible limitations of the proposed algorithm is the universal applicability on any urban scenario. Since the urban density and city morphology adds limitations for every unique situation. If a civil authority seeks a specific flight policy that can apply to all cases of diverse geospatial complexity to operate autonomous civil UAV flights, it can either be prone to higher risk factors or severely restrict the viable airspace and UAV size/type choice. While the proposed method can efficiently determine the adequate policy combination (β_{max} ; δ_0 ; r), simulations are inevitable for precise results. In addition to evaluating the airspace usability, our approach generates a crucial dataset to model civil airspace in 3D. Identifying the continuity of trajectories will be necessary for structured urban airspace design and path planning. This will strategically serve developers, planners and decision-aiding authorities such as the Model Aeronautics Association of Canada (MAAC) to operationalize UAVs in the near future. The integration of smart, sustainable and autonomous robotics for transportation in smart cities represents a silver bullet solution for the aforementioned challenges. In the future, we plan to add more uncertainties such as wind dynamics to add robustness to the proposed airspace discretization algorithm and increase energy-efficiency.

Data availability

The LiDAR and the 3D city model utilized in this study are available through the Toronto City Data Catalog at <https://mdl.library.utoronto.ca/collections/geospatial-data/toronto-lidar-2015>. <https://www.toronto.ca/city-government/data-research-maps/open-data/>.

Received: 30 January 2023; Accepted: 15 May 2024

Published online: 31 May 2024

References

1. Parliament, E. *Mapping Smart Cities in the EU* (European Parliament, 2014).
2. Mohammed, F., Idries, A., Mohamed, N., Al-Jaroodi, J., & Jawhar, I. UAVs for smart cities: Opportunities and challenges. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* 267–273. (IEEE, 2014).
3. Khan, M. A., Alvi, B. A., Safi, A., & Khan, I. U. Drones for good in smart cities: A review. In *Proc. Int. Conf. Elect., Electron., Comput., Commun., Mech. Comput. (EECCMC)* 1–6 (2018).
4. Shakhathreh, H. *et al.* Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* 7, 48572–48634 (2019).
5. Lemardel, C., Estrada, M., Pagès, L., & Bachofner, M. Potentialities of drones and ground autonomous delivery devices for last-mile logistics. *Transp. Res. Part E Logist. Transp. Rev.* 149, 102325 (2021).
6. BinJunaid, A., Lee, Y., & Kim, Y. Design and implementation of autonomous wireless charging station for rotary-wing UAVs. *Aerosp. Sci. Technol.* 54(2016), 253–266 (2016).
7. Merkert, R. & Bushell, J. Managing the drone revolution: A systematic literature review into the current use of airborne drones and future strategic directions for their effective control. *J. Air Transp. Manag.* 89, 101929 (2020).
8. Morbidi, F., Cano, R., & Lara, D. Minimum-energy path generation for a quadrotor UAV. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* 1492–1498. (IEEE, 2016).
9. Chen, C., Demir, E., Huang, Y., & Qiu, R. The adoption of self-driving delivery robots in last mile logistics. *Transp. Res. Part E Logist. Transp. Rev.* 146, 102214 (2021).
10. Truong, D. & Choi, W. Using machine learning algorithms to predict the risk of small Unmanned Aircraft System violations in the National Airspace System. *J. Air Transp. Manag.* 86, 101822 (2020).
11. Song, L., Wanke, C., Zobell, S., Greenbaum, D., & Jackson, C. Methodologies of estimating the impact of severe weather on airspace capacity. In *The 26th Congress of ICAS and 8th AIAA ATIO* 8917 (2008).
12. Nesbit, P. R., Barchyn, T. E., Hugenholtz, C. H., Cripps, S., & Kucharczyk, M. Reported UAV incidents in Canada: analysis and potential solutions. *J. Unmanned Veh. Syst.* 5(2), 51–61 (2017).
13. Vattapparamban, E., Güvenç, İ., Yurekli, A. İ., Akkaya, K., & Uluagaç, S. Drones for smart cities: Issues in cybersecurity, privacy, and public safety. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)* 216–221 (IEEE, 2016).
14. Altawy, R. & Youssef, A. M. Security, privacy, and safety aspects of civilian drones: A survey. *ACM Trans. Cyber Phys. Syst.* 1(2), 1–25 (2016).
15. Rao, B., Gopi, A. G., & Maione, R. The societal impact of commercial drones. *Technol. Soc.* 45, 83–90 (2016).
16. Dung, N. D., & Rohacs, J. The drone-following models in smart cities. In *2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)* 1–6 (IEEE, 2018).
17. Barr, L. C., Newman, R., Ancel, E., Belcastro, C. M., Foster, J. V., Evans, J., & Klyde, D. H. Preliminary risk assessment for small unmanned aircraft systems. In *17th AIAA Aviation Technology, Integration, and Operations Conference* 3272 (2017).
18. Foina, A. G., Sengupta, R., Lerchi, P., Liu, Z., & Krainer, C. Drones in smart cities: Overcoming barriers through air traffic control research. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)* 351–359 (IEEE, 2015).
19. Coutinho, W. P., Battarra, M., & Fliege, J. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Comput. Ind. Eng.* 120, 116–128 (2018).
20. Vaidis, M. Survey on UAV, lidar and underground mapping. (2019). <https://norlab.ulaval.ca/publications>.
21. Wang, R., Peethambaran, J., & Chen, D. LiDAR point clouds to 3-D urban models: A review. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 11(2), 606–627 (2018).
22. ElSayed, M., & Mohamed, M. The impact of airspace regulations on unmanned aerial vehicles in last-mile operation. *Transp. Res. Part D Transp. Environ.* 87, 102480 (2020).
23. Stöcker, C., Bennett, R., Nex, F., Gerke, M., & Zevenbergen, J. Review of the current state of UAV regulations. *Remote Sens.* 9(5), 459 (2017).
24. Luo, C., McClean, S. I., Parr, G., Teacy, L., & De Nardi, R. UAV position estimation and collision avoidance using the extended Kalman filter. *IEEE Trans. Veh. Technol.* 62(6), 2749–2762 (2013).
25. Masiero, A., Fissore, F., Guarnieri, A., Pirotti, F., & Vettore, A. UAV positioning and collision avoidance based on RSS measurements. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 40(1), 219 (2015).
26. Department of Transport, Part I: Vol. 151. Regulations Amending the Canadian Aviation Regulations (Unmanned Aircraft Systems) (2017).
27. International Civil Aviation Organization ICAO. Manual on Remotely Piloted Aircraft Systems (RPAS). International Civil Aviation Organization (2015).
28. National Academies of Sciences, Engineering, and Medicine. Advancing aerial mobility: A national blueprint. (The National Academies Press, 2020). <https://doi.org/10.17226/25646>.
29. Cheung, K. F., Bell, M. G., & Bhattacharjya, J. Cybersecurity in logistics and supply chain management: An overview and future research directions. *Transp. Res. Part E Logist. Transp. Rev.* 146, 102217 (2021).
30. Zhi, Y., Fu, Z., Sun, X., & Yu, J. Security and privacy issues of UAV: A survey. *Mob. Netw. Appl.* 25(1), 95–101 (2020).
31. Rajendran, S., & Srinivas, S. Air taxi service for urban mobility: A critical review of recent developments, future challenges, and opportunities. *Transp. Res. Part E Logist. Transp. Rev.* 143, 102090 (2020).
32. Majumdar, A., Ochieng, W., & Polak, J. Estimation of European airspace capacity from a model of controller workload. *J. Navig.* 55(3), 381–403 (2002).
33. Krozel, J., Mitchell, J. S., Polishchuk, V., & Prete, J. Maximum flow rates for capacity estimation in level flight with convective weather constraints. *Air Traffic Control Q.* 15(3), 209–238 (2007).
34. Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., & Robinson, J. E. Unmanned aircraft system traffic management (UTM) concept of operations (2016).
35. Dill, E. T., Young, S. D., & Hayhurst, K. J. SAFEGUARD: An assured safety net technology for UAS. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)* 1–10 (IEEE, 2016).
36. Hoekstra, J., Kern, S., Schneider, O., Knabe, F., & Lamiscarre, B. *Metropolis-Urban Airspace Design* (Technical University of Delft National, Tech, 2015).
37. Salleh, M. F. B., & Low, K. H. Concept of operations (ConOps) for traffic management of Unmanned Aircraft Systems (TM-UAS) in urban environment. In *AIAA Information Systems-AIAA Infotech@ Aerospace* 0223 (2017).
38. Salleh, M. F. B., Wanchao, C., Wang, Z., Huang, S., Tan, D. Y., Huang, T., & Low, K. H. Preliminary concept of adaptive urban airspace management for unmanned aircraft operations. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace* 2260 (2018).
39. Clothier, R. *et al.* The smart skies project. *IEEE Aerosp. Electron. Syst. Mag.* 26(6), 14–23 (2011).
40. Kopardekar, P. H. Unmanned aerial system (UAS) traffic management (UTM): Enabling low-altitude airspace and UAS operations (2014).
41. Johnson, S. C., Petzen, A., & Tokotch, D. Exploration of detect-and-avoid and well-clear requirements for small UAS maneuvering in an urban environment. In *17th AIAA Aviation Technology, Integration, and Operations Conference* 3074 (2017).

42. D'Souza, S., Ishihara, A., Nikaido, B., & Hasseeb, H. Feasibility of varying geo-fence around an unmanned aircraft operation based on vehicle performance and wind. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)* 1–10 (IEEE, 2016).
43. Johnson, M., Jung, J., Rios, J., Mercer, J., Homola, J., Prevot, T., *et al.* Flight test evaluation of an unmanned aircraft system traffic management (UTM) concept for multiple beyond-visual-line-of-sight operations. (2017).
44. Cho, J. & Yoon, Y. How to assess the capacity of urban airspace: A topological approach using keep-in and keep-out geofence. *Transp. Res. Part C Emerg. Technol.* **92**, 137–149 (2018).
45. Torija, A. J., Li, Z. & Self, R. H. Effects of a hovering unmanned aerial vehicle on urban soundscapes perception. *Transp. Res. Part D Transp. Environ.* **78**, 102195 (2020).
46. Atkins, E. Autonomy as an enabler of economically-viable, beyond-line-of-sight, low-altitude UAS applications with acceptable risk. In *AUVSI unmanned Systems* (2014).
47. Khamis, A., Hussein, A., & Elmogy, A. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks 2015* 31–51 (Springer, 2015).
48. Darrach, M. *et al.* Using genetic algorithms for tasking teams of raven UAVs. *J. Intell. Robot. Syst.* **70**(1–4), 361–371 (2013).
49. Hu, X., Cheng, J., & Luo, H. Task assignment for multi-UAV under severe uncertainty by using stochastic multicriteria acceptability analysis. *Math. Probl. Eng.* (2015).
50. Rathinam, S., & Sengupta, R. Algorithms for routing problems involving uavs. In *Innovations in Intelligent Machines-1* 147–172 (Springer, 2007).
51. Zhang, Y., Chen, J. & Shen, L. Hybrid hierarchical trajectory planning for a fixed-wing UCAV performing air-to-surface multi-target attack. *J. Syst. Eng. Electron.* **23**(4), 536–552 (2012).
52. Betts, J. T. Survey of numerical methods for trajectory optimization. *J. Guid. Control Dyn.* **21**(2), 193–207 (1998).
53. Moshref-Javadi, M., Lee, S. & Winkenbach, M. Design and evaluation of a multi-trip delivery model with truck and drones. *Transp. Res. Part E Logist. Transp. Rev.* **136**, 101887 (2020).
54. Shen, L. *et al.* Synergistic path planning of multi-UAVs for air pollution detection of ships in ports. *Transp. Res. Part E Logist. Transp. Rev.* **144**, 102128 (2020).
55. Chen, Y. *et al.* Trajectory optimization of multiple quad-rotor UAVs in collaborative assembling task. *Chin. J. Aeronaut.* **29**(1), 184–201 (2016).
56. Jaishankar, S. & Pralhad, R. N. 3D off-line path planning for aerial vehicle using distance transform technique. *Proc. Comput. Sci.* **4**, 1306–1315 (2011).
57. Wang, J., Zhang, Y. F., Geng, L., Fuh, J. Y. H. & Teo, S. H. A heuristic mission planning algorithm for heterogeneous tasks with heterogeneous UAVs. *Unmanned Syst.* **3**(03), 205–219 (2015).
58. Wu, J. P., Peng, Z. H. & Chen, J. 3D multi-constraint route planning for UAV low-altitude penetration based on multi-agent genetic algorithm. *IFAC Proc. Vol.* **44**(1), 11821–11826 (2011).
59. Xu, S., Doğançay, K. & Hmam, H. Distributed pseudolinear estimation and UAV path optimization for 3D AOA target tracking. *Signal Process.* **133**, 64–78 (2017).
60. Zhang, X., Chen, J., Xin, B. & Fang, H. Online path planning for UAV using an improved differential evolution algorithm. *IFAC Proc. Vol.* **44**(1), 6349–6354 (2011).
61. Lin, X., Wang, C., Wang, K., Li, M. & Yu, X. Trajectory planning for unmanned aerial vehicles in complicated urban environments: A control network approach. *Transp. Res. Part C Emerg. Technol.* **128**, 103120 (2021).
62. Zhang, N., Zhang, M. & Low, K. H. 3D path planning and real-time collision resolution of multirotor drone operations in complex urban low-altitude airspace. *Transp. Res. Part C Emerg. Technol.* **129**, 103123 (2021).
63. Zhao, P., Yang, Y., Zhang, Y., Bian, K., Song, L., Qiao, P., & Li, Z. Optimal trajectory planning of drones for 3D mobile sensing. In *2018 IEEE global communications conference (GLOBECOM)* 1–6 (IEEE, 2018).
64. Bagherian, M. & Also, A. 3D UAV trajectory planning using evolutionary algorithms: A comparison study. *Aeronaut. J.* **119**(1220), 1271–1285 (2015).
65. Bahabry, A. *et al.* Low-altitude navigation for multi-rotor drones in urban areas. *IEEE Access* **7**, 87716–87731 (2019).
66. U-space Blueprint, SESAR Joint Undertaking, SESAR Joint Undertaking, 2017, ISBN: 978-92-9216-087-6.
67. Youn, J., Kim, D., Kim, T., Yoo, J. H., & Lee, B. J. Development of UAV air roads by using 3D grid system. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, No. 4 731–735 (2018).
68. OpenStreetMap. <https://www.openstreetmap.org/> (accessed May 2017).
69. Kazhdan, M., Bolitho, M., & Hoppe, H. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, vol. 7 (2006).
70. Rossignac, J. R. & Requicha, A. A. Offsetting operations in solid modelling. *Comput. Aided Geometr. Des.* **3**(2), 129–148 (1986).
71. Bischoff, S., Pavic, D. & Kobbelt, L. Automatic restoration of polygon models. *ACM Trans. Graph. TOG* **24**(4), 1332–1352 (2005).
72. Wang, H., Lyu, W., Yao, P., Liang, X. & Liu, C. Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system. *Chin. J. Aeronaut.* **28**(1), 229–239 (2015).
73. Edelsbrunner, H., Kirkpatrick, D. & Seidel, R. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory* **29**(4), 551–559 (1983).
74. ElSayed, M. S. Optimizing thermal performance of building-integrated photovoltaics for upgrading informal urbanization. *Energy Build.* **116**, 232–248 (2016).
75. Ansys. <https://www.ansys.com/>. (accessed May 2018).
76. Chetverikov, D., Stepanov, D. & Krsek, P. Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm. *Image Vis. Comput.* **23**(3), 299–309 (2005).
77. Zhang, Z. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* **13**(2), 119–152 (1994).
78. Koyuncu, E., Ure, N. K. & Inalhan, G. Integration of path/maneuver planning in complex environments for agile maneuvering ucavs. *J. Intell. Robot. Syst.* **57**(1–4), 143 (2010).
79. Liu, S., & Lü, F. F. An efficient variable-step algorithm for Bezier curves. In *2010 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 3, 302–305 (IEEE, 2010).
80. ElSayed, M., & Mohamed, M. The uncertainty of autonomous unmanned aerial vehicles' energy consumption. In *2020 IEEE Transportation Electrification Conference & Expo (ITEC)* 8–13. (IEEE, 2020).
81. ElSayed, M. & Mohamed, M. The impact of airspace discretization on the energy consumption of autonomous unmanned aerial vehicles (Drones). *Energies* **15**(14), 5074 (2022).
82. ElSayed, M., Foda, A. & Mohamed, M. Autonomous drone charging station planning through solar energy harnessing for zero-emission operations. *Sustain. Cities Soc.* **86**, 104122 (2022).
83. Samà, M., D'Ariano, A. & Pacciarelli, D. Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports. *Proc. Soc. Behav. Sci.* **80**, 531–552 (2013).
84. Birgin, E. G., Martinez, J. M. & Ronconi, D. P. Optimizing the packing of cylinders into a rectangular container: A nonlinear approach. *Eur. J. Oper. Res.* **160**(1), 19–33 (2005).
85. Yao, P., Wang, H. & Su, Z. UAV feasible path planning based on disturbed fluid and trajectory propagation. *Chin. J. Aeronaut.* **28**(4), 1163–1177 (2015).
86. Chen, Y. B., Luo, G. C., Mei, Y. S., Yu, J. Q. & Su, X. L. UAV path planning using artificial potential field method updated by optimal control theory. *Int. J. Syst. Sci.* **47**(6), 1407–1420 (2016).

87. Fagnant, D. J. & Kockelman, K. M. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transp. Res. Part C Emerg. Technol.* **40**, 1–13 (2014).
88. Stolaroff, J. K. *et al.* Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nat. Commun.* **9**(1), 1–13 (2018).
89. City of Toronto. <https://www.toronto.ca/city-government/data-research-maps/maps/>. (accessed May 2018).
90. Google Earth. <https://earth.google.com/web/>. (accessed May 2018).
91. ElSayed, M., Foda, A. & Mohamed, M. The impact of civil airspace policies on the viability of adopting autonomous unmanned aerial vehicles in last-mile applications. *Transp. Policy* **145**, 37–54 (2024).
92. Shekhar, R., Fayyad, E., Yagel, R., & Cornhill, J. F. Octree-based decimation of marching cubes surfaces. In *Proceedings of Seventh Annual IEEE Visualization'96* 335–342 (IEEE, 1996).
93. Lorensen, W. E., & Cline, H. E. *Marching cubes: A high resolution 3D surface reconstruction algorithm, computer graphics, vol. 21, no. 3 (1987).
94. Paul Bourke. <http://paulbourke.net/geometry/polygonise/source1.c> (accessed June 2020).
95. Luukkonen, T. *Modelling and Control of Quadcopter, Independent Research Project in Applied Mathematics* (Aalto University, 2011).
96. Ackermann, J. *Robust Control Systems with Uncertain Physical Parameters* (Springer, 1993).

Author contributions

M.E. and M.M. conceptualized the study. M.E. developed the methodology. M.E. created the code and carried out the formal analysis and visualization. All the authors performed data curation and took part in consultation. The original draft was written by M.E. with review and editing by M.M. Funding acquisition was by M.M., who supervised the study.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-62421-4>.

Correspondence and requests for materials should be addressed to M.E.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024