Altitude Zoning for UTM

Leonid Sedov Valentin Polishchuk
Communications and Transport Systems, ITN
Linköping University
Norrköping, Sweden

Virgilio Acuna
Electrical Engineering
Florida International University
Miami, USA

***Abstract*—We give algorithms for splitting a geographical region into "approximately flat" rectangular zones. Each zone is assigned a feasible flight altitude, providing simple flight level guidance for future operations of unmanned aerial systems in low-level airspace. We consider a rural scenario with uneven ground level and an urban setting featuring many tall structures. In both cases, our solutions adapt to the underlying terrain or city landscape. In the rural scenario, operating on a fixed altitude within a rectangle allows the drone to stay within the upper limit of 120m while not flying too close to the ground; our objective is to minimize the complexity of the airspace. In the urban case, we aim at minimizing the volume of airspace reserved for drone operations, while allowing overflight over tall buildings in the city. Experiments with real landscape and city skyline data demonstrate output of our solutions with various input parameters.**

I. Introduction and Problems formulation

Future unmanned aerial systems (UAS) will often have to operate in altitude-constrained environments: the thin band of allowable Unmanned Aerial Vehicles (UAV) flight levels is bounded from above by conventional aircraft operations, while noise, safety, privacy and many other concerns will constrain the minimum flyable altitudes. These challenges of distributing the drones into the narrow altitude strips are well known to Unmanned Traffic Management (UTM) concept developers: for example, all three airspace issues identified at CORUS Exploratory Workshop [1] were related to the height; in particular, the overall view on the specific question of flight levels was that a multitude of layers would rather be an exception than the rule. The problem is further exacerbated by the altimetry equipment errors: it follows from findings in [2] (and is widely acknowledged in general) that vertical bands will be a scarce resource in UTM.

One immediate question, highlighted e.g. in Eurocontrol/EASA discussion document [3] (and followed upon in SESAR project ICARUS), is the frame of reference for UAV height (Figure 1): following constant height AGL (above ground level) may be too tricky due to a complicated terrain profile, while maintaining the same MSL (mean sea level) altitude may be infeasible for the whole duration of the mission (the vehicle may simply bump into a hill). The Eurocontrol/EASA study, reported in [3], explored also other ways of measuring UAV height and concluded that no onboard altimetry system is sufficiently accurate, and hence UTM must step in to help with designation of altitude/height for the drone flights: the winner among the potential solutions studied in [3] was "OPTION 3: EACH AIRSPACE USER WILL USE AN APPROVED ALTIMETRY SYSTEM BEST SUITED TO THE AIRSPACE OR CERTIFICATION REQUIREMENTS, AND UTM FUNCTIONALITIES ARE USED".

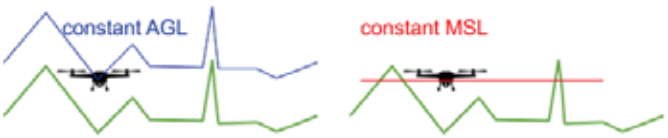

Figure 1. Neither AGL nor MSL alone will work everywhere

On the regulatory side, Commission Delegated Regulation (EU) 2019/945 [4] mandates that every drone is "equipped with a geo-awareness system that provides an interface to load and update data containing information on airspace limitations related to UA position and altitude imposed by the geographical zones". Obviously, it would be an overkill to load the full precise landscape profile on every drone and require the vehicle to follow the terrain precisely, making the operator solely responsible for hedging against height violations. Instead, in line with the main conclusion in [3], it may be more reasonable to partially outsource the height designation to UTM system which would structure the airspace accordingly, to simplify fight planning and execution.

The **main idea** explored in this paper is to split the region of interest (ROI) into few simple (rectangular) *zones* and assign a fixed flight altitude to each zone. This may serve as a potential way to implement the UTM service functionality of guiding the determination of flights vertical profiles. But how such a zoning should be done, what constraints must be satisfied by the zones? The next two subsections give possible answers to these questions by taking a closer look at the flight height restrictions in a rural (Section I-A) and an urban (Section I-B) setting.

In both settings, we assume that our ROI is given as a square $N \times N$ grid, over which function $T : (1, \dots, N) \times (1, \dots, N) \mapsto \mathbb{R}$ is specified: for any $(x, y) \in (1, \dots, N) \times (1, \dots, N)$, the value $T(x, y)$ is the elevation of the terrain (in the rural case) or the roof of the building (in the city) at the point (x, y). The domain of T is assumed to be a square only for ease of exposition; our methods apply to any ROI shape, possibly even with holes representing no-fly zones, which can be split into rectangles.

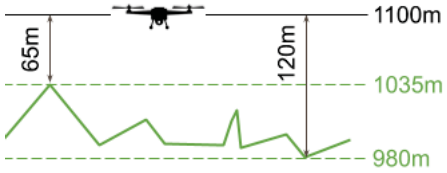

Figure 2. A piece of terrain, with minimum and maximum elevations 980m and 1035m resp. and assigned altitude 1100m: even in the closest (resp. furthest) point the distance from the ground is 65m (resp. 120m).

A. Partitioning a terrain

Let R be a zone, and let z_R be the flight altitude assigned to R. One the one hand, the drone should not fly higher than 120m above the terrain [5], or formally:

$$\forall (x,y) \in R, \quad z_R - T(x,y) \leq 120$$

Clearly, the above holds if and only iff

$$z_R - \min(R) \leq 120 \tag{1}$$

where $\min(R)$ is the minimum terrain elevation in R. On the other hand, the flight should not happen too close to the ground (due to noise and other considerations). Let h be the minimum allowable flight height. We are not aware of laws that would give a precise figure for h; for concreteness, here we assume $h = 60$m (in Section III-A we experiment with other values for h). Thus,

$$\forall (x,y) \in R, \quad z_R - T(x,y) \geq 60$$

The above holds if and only iff

$$z_R - \max(R) \geq 60 \tag{2}$$

where $\max(R)$ is the maximum terrain elevation in R. From (1) and (2), a fixed flight altitude can be assigned to R iff

$$\max(R) - \min(R) \leq 60 \tag{3}$$

Thus the question of zoning boils down to partitioning the ROI into rectangles each "sandwiched" between horizontal planes at distance 60m. Formally, the problem statement is:

> TERRAINZONING: Given the function $T : (1,\ldots,N) \times (1,\ldots,N) \mapsto \mathbb{R}$, partition $(1,\ldots,N) \times (1,\ldots,N)$ into the minimum number of rectangles so that $\max(R) - \min(R) \leq 60$ for every rectangle R.

Figure 2 illustrates the overall idea with a 1d example.

B. Urban partitioning

In a city center, the ground level (terrain elevation) may no longer serve as the limiting factor for UAS vertical flight profiles; instead, drones trajectories are influenced by avoidance of tall buildings. Therefore here the function T, specified as the problem input, will represent the altitude of the skyscrapers rooftops after each building's footprint has been fattened (in the horizontal plane) by 50m (Figure 3), i.e., for a point (x,y) in the ROI, $T(x,y)$ is the altitude of the highest rooftop within horizontal distance of 50m from

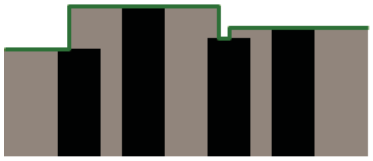

Figure 3. 1d skyscrapers (black) with offsets (gray); T is the green line.

(x,y). Our output is a decomposition of the ROI into a given number K of rectangular zones. Every rectangle R will be assigned the flight level $z_R = \max(R)$ (allowing the drones to fly over the buildings). Our goal is to find the partitioning that minimizes the total volume under the rectangles. Denoting the area of rectangle R by $\mathrm{Area}(R)$, the formal problem statement becomes:

> CITYZONING: Given the function $T : (1,\ldots,N) \times (1,\ldots,N) \mapsto \mathbb{R}$ and the target number K of the zones, partition $(1,\ldots,N) \times (1,\ldots,N)$ into K rectangles so that $\sum_R \max(R) \cdot \mathrm{Area}(R)$ is minimized.

Informally, we want to "package" the city into K boxes of minimum total volume.

We discuss our modeling choices and connect them to other research in the Related work section (Section II) below. Section III gives our solutions and presents the results. Section IV describes possible extensions and future work.

II. RELATED WORK

Use of flight levels is ubiquitous in aviation – both in ATM and UTM; in particular, Layers was the winning concept in the comprehensive study of airspace structuring performed in the Metropolis project [6]–[11]. This paper departs from the prevailing research paradigm of using the same altitude restrictions everywhere in the ROI: we explore fine-grained altitude structure adapting to the underlying landscape or skyline.

Offsetting the buildings footprints by 50m, when defining T, reflects the restriction to stay further than 50m from man-made structures [5]. We assumed that our ROI is dense in the sense that any point in it has a building within 50m, so the buildings fully define T. Note that if there is no tall building within 50m of a point (x,y), then a drone flying above 120m over the point would technically violate the height restriction (such exceptions are allowed by [5] for light drones). Since the total volume under T is a constant, our objective is equivalent to minimizing $\sum_R \sum_{(x,y)\in R}(z_R - T(x,y))$, which minimizes such exceptions.

Allowing to fly over any building is admittedly unrealistic: in practice the overflight permission may hinge on request from the building owner [5]. Guessing which owners would make such requests and whose requests will be satisfied by which authority is, however, far outside the scope of this paper; for considerations of interaction with authorities see [12] and references thereof. Anyway, our methods extend directly to work with obstacles of any nature, in particular, no-fly zones due to buildings. Setting $z_R = \max(R)$ means using what was

called the "lowest navigable altitude" in [13] which explored how the altitude changes with the minimum allowable distance to the buildings (the parameter set to 50m in our paper).

Our assignment of the zones flight levels support direct, obstacle-free routing between any origin and destination point in the city. Direct tracks are beneficial in many respects; to name one, we remark that in the presence of obstacles, shortest paths tend to bend at the obstacles vertices and go along their edges, creating congestion. However, direct routing in our setup comes at the cost of possibly changing the altitude during a flight, ignored in our model. Vertical and horizontal tradeoffs for a *single* flight were recently considered in [14] which also used clustering of the buildings by height and geographical location (our clustering by the zones is implicit and is done with completely different techniques).

Last but not least, our models are motivated by rules for UAS open category; their applicability to specific and certified categories is yet to be seen. A further step, called for in SESAR Exploratory Research [15, Area 2.7.2], is the development of a common altitude reference system (CARS) for UTM and manned aviation. CARS is one of the three distinct points for which the guidelines were considered necessary also by Eurocontrol's UAS ATM airspace assessment [16]. By minimizing the "city UTM volume" (urban airspace reserved for U-space) we maximize the room left for the conventional aviation which often needs a good deal of metropolitan airspace close to the ground (cf. Class B airspace and the like). See [17] for a quantitative study of urban airspace availability, taking into account possible interactions between UTM and ATM.

III. Solution and Application

We give integer programming formulations to our problems. For TerrainZoning (Section I-A), we start from determining $\min(R)$ for every rectangle R with corners in our $N \times N$ grid. To speed up this preprocessing step, the terrain T is put into 2D Range Minimum Query data structure [18] whose purpose is precisely to report efficiently the minimum value of T in a rectangle. We put the terrain into a similar data structure to determine also the maximum elevation $\max(R)$ in every rectangle. We keep only the rectangles satisfying $\max(R) - \min(R) \leq 60$ (3); let C denote the set of such rectangles (these are the candidate zones).

Our goal is to pick the smallest subset of rectangles from C so that every point in $(1, \ldots, N) \times (1, \ldots, N)$ belongs to exactly one rectangle. The integer programs (IP) for the problem is:

$$\min \sum_{R \in C} x_R \qquad \text{s.t.}$$
$$\sum_{R \ni (x,y)} x_R = 1 \quad \forall (x,y) \in (1, \ldots, N) \times (1, \ldots, N)$$
$$x_R \in \{0, 1\}$$

The IP has a binary variable x_R for every rectangle $R \in C$, indicating whether R is picked. The constraints ensure that the picked rectangles partition the domain of T.

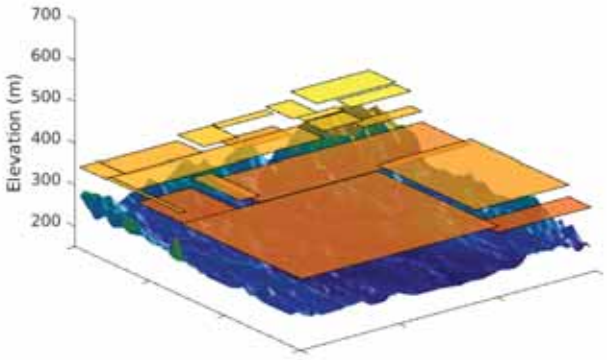

Figure 4. TerrainZoning of a 12.5x12.5km^2 area around Eksjö city; semi-transparent rectangles above the terrain represent the altitude zones. Terrain resolution is $N \times N = 60 \times 60$, and $h = 60$m. The terrain height data is courtesy of [19]. For illustrative purposes Z-axis is scaled differently than X- and Y-axes and rectangles colors range from red to yellow depending on the altitude.

For CityZoning (Section I-B), we determine only $\max(R)$ for every rectangle R with corners in the grid ($\min(R)$ is irrelevant here, and any rectangle is a feasible zone); we also calculate the area $\mathrm{Area}(R)$. Let $V_R = \max(R) \cdot \mathrm{Area}(R)$ denote the volume under R if it is chosen as a zone. As with TerrainZoning, the IP for CityZoning has the indicator variables x_R and the constraints ensuring that the zones partition the domain. We add the constraint that the total number of zones is bounded and change the objective to minimize the volume under the zones. The IP for CityZoning is thus:

$$\min \sum_{R \in C} V_R x_R \qquad \text{s.t.}$$
$$\sum_{R \ni (x,y)} x_R = 1 \quad \forall (x,y) \in (1, \ldots, N) \times (1, \ldots, N)$$
$$\sum_{R} x_R \leq K$$
$$x_R \in \{0, 1\}$$

A. Experimental results

We solved the TerrainZoning for a 12.5x12.5km^2 area around Eksjö city in Sweden. The terrain data was obtained from [19] LIDAR height data and downscaled to resolution 60×60 (that is, $N = 60$). The IP was solved using Gurobi optimization software installed on Tetralith cluster [20] of Intel HNS2600BPB nodes with 32 CPU cores, provided by the Swedish National Infrastructure for Computing (SNIC).

Figure 4 shows the resulting sectorization of the ROI into zones.

Since we are not aware of the specific values for h (the lower bound on the allowable height for drones flights), we do our experiments with several settings for h. Figure 5 illustrates

the outputs for $h = 40$ and 80m. It can be seen that with smaller h, simpler airspace structure may work (fewer zones are needed). Figure 6 shows how the number of the required zones depends on h in the range between 20 and 80m.

For the urban scenario, we used New York 3D buildings data obtained from [21]. We generated a raster representation of the vector data using QGIS software package [22] with resolution 45×45 (Figure 7 shows the area of interest before and after offsetting buildings by 50m). Figure 8 shows a division into $K = 10$ rectangles of a 2×2 km^2 area around Empire State Building.

Despite the fact that solving IP formulations allows to obtain exact optimal solutions, it takes an impractical amount of time and computational resources for both urban and rural scenarios even for very small datasets with low resolution: we were not able to produce results with terrain resolutions higher than 60×60 for rural scenario and 45×45 for urban scenario within reasonable amount of time. For example, for the small 2×2 km area around Empire State Building in New York which we used in our experiments size of the terrain 45×45 means that every "pixel" is ≈ 44 meters wide, which leads to loss of many terrain details. In order to cope with this problem we have developed heuristics which allow to solve the problem on much more dense datasets. In the following subsections we present our heuristics and benchmark them.

B. TerrainZoning *heuristic*

We used a simple greedy heuristic for TerrainZoning. Our heuristic covers the terrain T with as big rectangles as possible one by one. Specifically, we start by growing a largest-area rectangle from the bottom left corner of T, while the rectangle still satisfies the height constraint (if there is more than one largest-area feasible rectangle, we select the widest one). We then iteratively select the leftmost point not covered by any rectangle (if there are several leftmost uncovered points, we select the bottommost of them) and grow a new rectangle which has the largest area until the whole T is covered with rectangles.

While this simple heuristic gives no guarantees in terms of the quality of the solution, it allows one to sectorize much larger datasets. For example, during our experiments we successfully solved the problem on terrains with sizes over 2000×2000 pixels which is a significant improvement comparing to the IP solution. Figure 9 demonstrates the difference in the level of details between a high-resolution 2500×2500 terrain and its downscaled 60×60 version. It can be observed that many details of the terrain disappear when resolution is downscaled.

To benchmark the efficiency of the heuristic we compared the solutions obtained with the heuristic with optimal solutions on 30 randomly generated terrains of size 60×60 and observed that the heuristic usually gives the result within 2× optimal number of rectangles. Figure 10 (left) shows a distribution of percentage increase in the number of rectangles produced by the heuristic compared to the number of rectangles in the optimal solution. We generated random terrains using diamond-square algorithm [23].

Figure 11 (left) shows a rectangulation produced by the heuristic on the same problem instance as on Figure 4. A solution for a high-resolution version of the same terrain is shown on Figure 11 (right).

C. CityZoning *heuristic*

For CityZoning problem we developed a heuristic based on celebrated simulated annealing technique [24]. Our heuristic is inspired by Binary Space Partitions [25] and works by repeatedly slicing the terrain into rectangles. Specifically, the algorithm always maintains a set S of rectangles that partition the terrain; initially S consists from only one rectangle, $[1, N] \times [1, N]$. The algorithm starts by choosing a set $Q = \{q_1, q_2, \ldots, q_{K-1}\}$ of $K-1$ points uniformly at random within T. Then for $i = 1, \ldots, K-1$ the algorithm takes the rectangle $R \in S$ to which point q_i belongs and splits R into two by making a horizontal or vertical cut through q_i (the algorithm alternates between horizontal and vertical cuts); R is removed from S and two new rectangles resulting from splitting are added. After doing all the splits (and thus obtaining a zoning with K rectangles in S), the algorithm computes the objective function of the resulting zoning (the objective function is the same as in the IP formulation, i.e., the total volume under the zones $\sum_{R \in S} \max(R) \cdot \mathrm{Area}(R)$). We then set the initial temperature t for the simulated annealing and make multiple simulated annealing iterations, decreasing t. At every iteration the algorithm randomly moves the points in Q (the radius within which every point can move decreases with t), runs the same routine with binary splits as during the initialization phase, and calculates the new objective function. In accordance with the simulated annealing paradigm, the algorithm keeps the new solution if the new objective function is better than the previous, but sometimes randomly allows to keep the worse solution – as the temperature decreases, keeping the worse solutions becomes more rare. Due to the random nature of the heuristic, to further reduce the effect of bad luck on the output quality, the heuristic can be run multiple times with different Random Number Generator (RNG) seed values.

Figure 12 (left) shows the splitting into $K = 10$ rectangles obtained by the heuristic on the same terrain as in Figure 8.

As with the TerrainZoning, the CityZoning heuristic is capable of solving much larger problems than IP formulation. Figure 12 (right) shows a solution produced by the heuristic for a high-definition terrain data ($N = 2000$).

We ran the heuristic on 30 40×40 real urban terrains in New York city and compared the efficiency of the heuristic with the exact solutions. To avoid effect of bad luck on the benchmarking results we ran the heuristic on the same problem instance a thousand times using different RNG seed values and picked the best result among those runs (given that one thousand runs of the heuristic on 40×40 take less than a minute, while obtaining the exact solution using IP formulation takes on average more than 3 hours, we find it

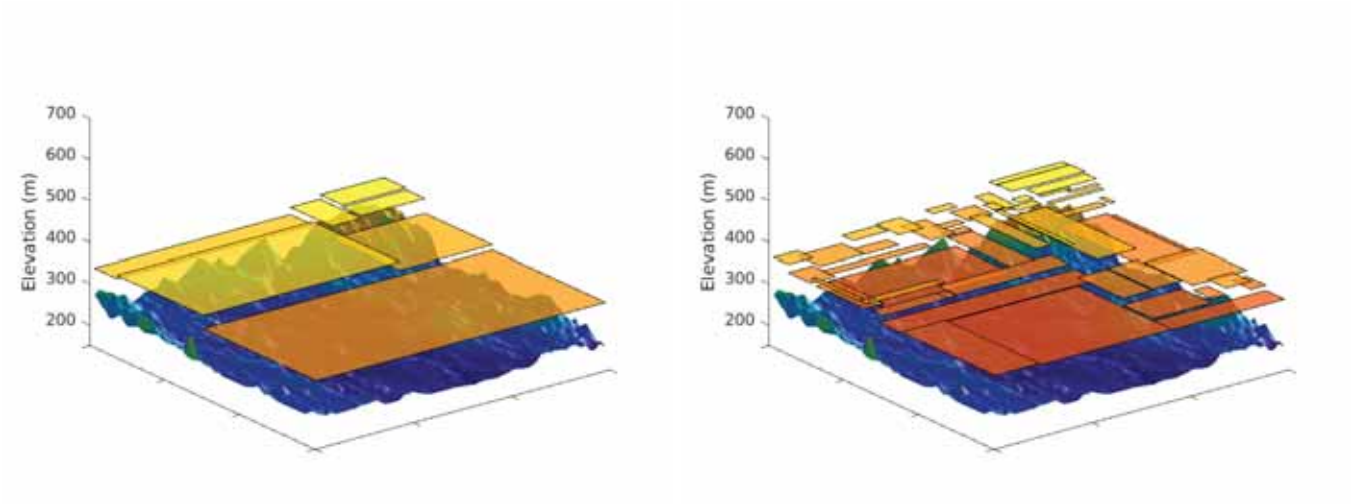

Figure 5. TERRAINZONING of area around Eksjö city for values $h = 40$ and 80m (left and right respectively).

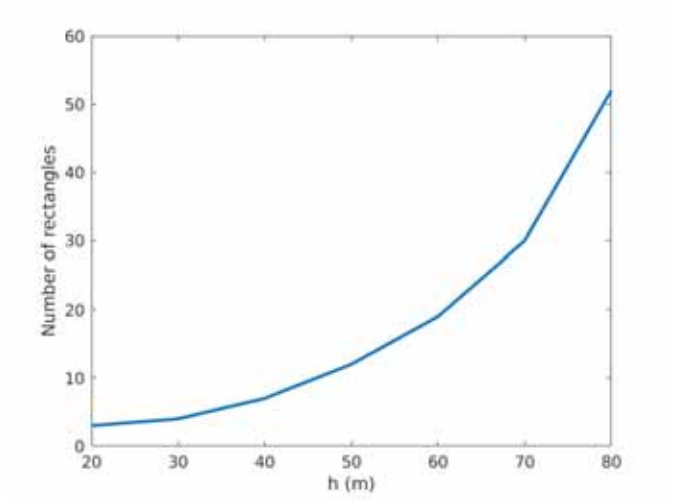

Figure 6. Number of rectangles above area around Eksjö city in TERRAINZONING as a function of h.

a fair comparison). Figure 10 (right) shows the histogram of percentage increase of the objective function value produced by the heuristic compared to the objective value of the optimal solution. We observe that the heuristic in most cases shows less than 10% increase in the value of the objective function while taking significantly less time than computing the exact solution.

IV. CONCLUSION

We considered splitting terrain/skyline into few almost-constant-height rectangular zones. In the rural scenario we minimized the number of the zones under the restriction on the height difference within a zone. In the urban setting we minimized the total volume of the airspace with the given number of zones. Several extensions are immediate.

The number of necessary zones may depend on the orientation of the rectangles. Therefore it might be of interest to experiment with rotating the grid. New York city may serve as an example where the rotation could influence the solution because of the grid-like streets pattern.

We can filter out any undesirable shapes of the rectangles. For instance, we can keep only the zones that contain a square of some given minimum size, to avoid narrow rectangles. We may also take care of constraints like incompatible zones – one reason for incompatibility may be that we do not want to jump too high between adjacent zones. This is easy to do with our IPs.

It is straightforward to add a tolerance, to allow for vertical positioning errors: if the imprecision is, say, 10m, then the maximum and minimum levels AGL would change from 120 and 60 to 110 and 70m resp., simply changing the equations (1) and (2) accordingly – the rest of our solution will stay the same. From (1) and (2), any $z_R \in [\max(R)+h, \min(R)+120]$ may be assigned as a flight altitude to R. In particular, the width $120 - h + \min(R) - \max(R)$ of the interval $[\max(R) + h, \min(R) + 120]$ indicates how robust (precise) the altimetry equipment must be when flying over R – a performance requirement towards PBN for UTM [2].

Our airspace design is traffic-oblivious. Taking the traffic into account (e.g., finding altitude patterns which disrupt the traffic as little as possible) may be an interesting research direction. For instance, one may generalize the problem by requiring that the zone boundaries do not cross too many trajectories or that the drones do not change altitude too often. Such traffic-adaptive, possibly non rectangular zones will fall into the realm of geovectoring [10]. Overall, it remains to be seen how to connect the zones ensuring safety when crossing between the zones and taking into account the energy required for the vertical movement [14].

Zoning like ours may be used not only to prescribe different flight altitudes in the different zones, but also for differentiating between different types of airspace. E.g., lowest VFR altitude above an urban area is the tallest building in the area

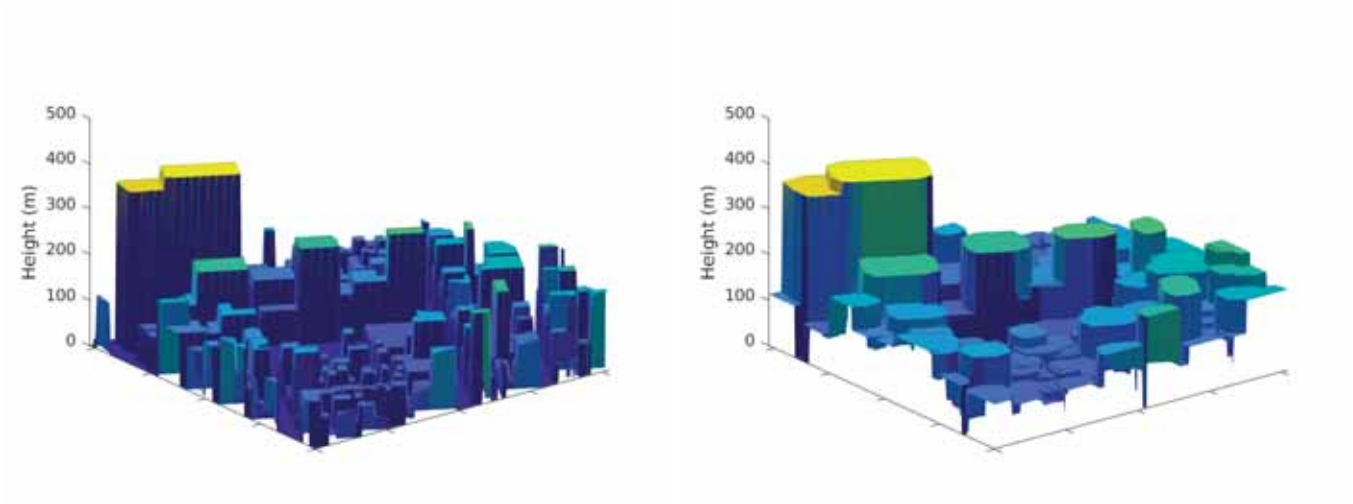

Figure 7. Buildings height profile in an area around Empire State Building in New York before and after offsetting the buildings by 50m (left and right respectively). Z-axis is scaled differently than X- and Y-axes for illustrative purposes.

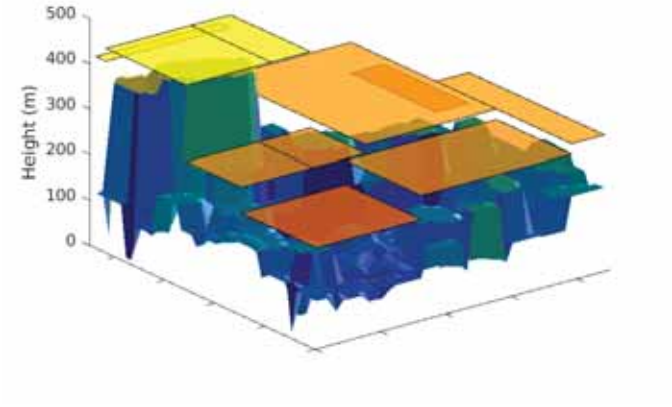

Figure 8. CITYZONING of area around Empire State Building in New York into $K = 10$ rectangles. Semi-transparent rectangles above the terrain represent the altitude zones. Rectangles colors range from red to yellow depending on the altitude. Terrain resolution is 45×45. The buildings height data is courtesy of [21].

plus 1000ft [26]; however, the definition of "urban area" is open to interpretation: sometimes it may be beneficial to set a higher lowest VFR level for a large area, while in other cases, the area may be split into smaller parts and separate lowest VFR levels may be designated to each part. Our method may be applied to see the benefits of the different possible airspace organizations.

Acknowledgements. We thank Prof. Vu N. Duong for asking the questions on UTM levels after presentation of [27] at ICRAT'18. Practical insight from Billy Josefsson, Pierre Ankartun and Roger Li (UTM project manager) from LFV is greatly appreciated. We thank Dr. Vishwanath Bulusu (Crown Consulting), Dr. Jungwoo Choi (KAIST), Dr. Parker Vascik (MIT), Helena Samsioe (Globhe) and Richard Wirén (Ericsson) for helpful discussions. This work is partially supported by Swedish Transport Administration (Trafikverket) and Swedish Research Council (Vetenskapprådet). We also acknowledge the anonymous referees whose comments helped with the presentation of the paper.

REFERENCES

[1] L. Bellesia, B. Clark, E. Malfliet, Y. Seprey, C. Ronfle-Nadaud, A. Wennerberg, C. Barrado, W. Van Leare, L. Brucculeri, and A. Hately, "CORUS 1st workshop early findings," 2018.

[2] L. Ren, M. Castillo-Effen, H. Yu, E. Johnson, Y. Yoon, N. Takuma, and C. A. Ippolito, "Small unmanned aircraft system (sUAS) categorization framework for low altitude traffic services," in *Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th*. IEEE, 2017, pp. 1–10.

[3] EASA/EUROCONTROL, "UAS ATM CARS (Common Altitude Reference System)," 2018, discussion Document. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2019-05/uas-atm-cars-v1.0-release-20181127_0.pdf

[4] European Commission, "Commission Delegated Regulation (EU) 2019/945," 2019, official Journal of the European Union. [Online]. Available: https://eur-lex.europa.eu/eli/reg_del/2019/945/oj

[5] ——, "Commission Implementing Regulation (EU) 2019/947," 2019, official Journal of the European Union. [Online]. Available: https://eur-lex.europa.eu/eli/reg_impl/2019/947/oj?eliuri=eli:reg_impl:2019:947:oj

[6] E. Sunil, J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, A. Vidosavljevic, and S. Kern, "Metropolis: Relating Airspace Structure and Capacity for Extreme Traffic Densities," in *11th USA/EUROPE Air Traffic Management R&D Seminar*, 06 2015.

[7] E. Sunil, J. Ellerbroek, J. Hoekstra, and J. Maas, "Modeling Airspace Stability and Capacity for Decentralized Separation," *DEP*, vol. 3, p. R1, 2017.

[8] M. Tra, E. Sunil, J. Ellerbroek, and J. Hoekstra, "Modeling the Intrinsic Safety of Unstructured and Layered Airspace Designs," in *2017 ATM R&D Seminar*, 2017.

[9] J. M. Hoekstra, J. Maas, M. Tra, and E. Sunil, "How do layered airspace design parameters affect airspace capacity and safety?" in *Proceedings of the 7th International Conference on Research in Air Transportation*, 2016.

[10] J. M. Hoekstra, J. Ellerbroek, E. Sunil, and J. Maas, "Geovectoring: Reducing traffic complexity to increase the capacity of uav airspace," in *International Conference for Research in Air Transportation, Barcelona, Spain*, 2018.

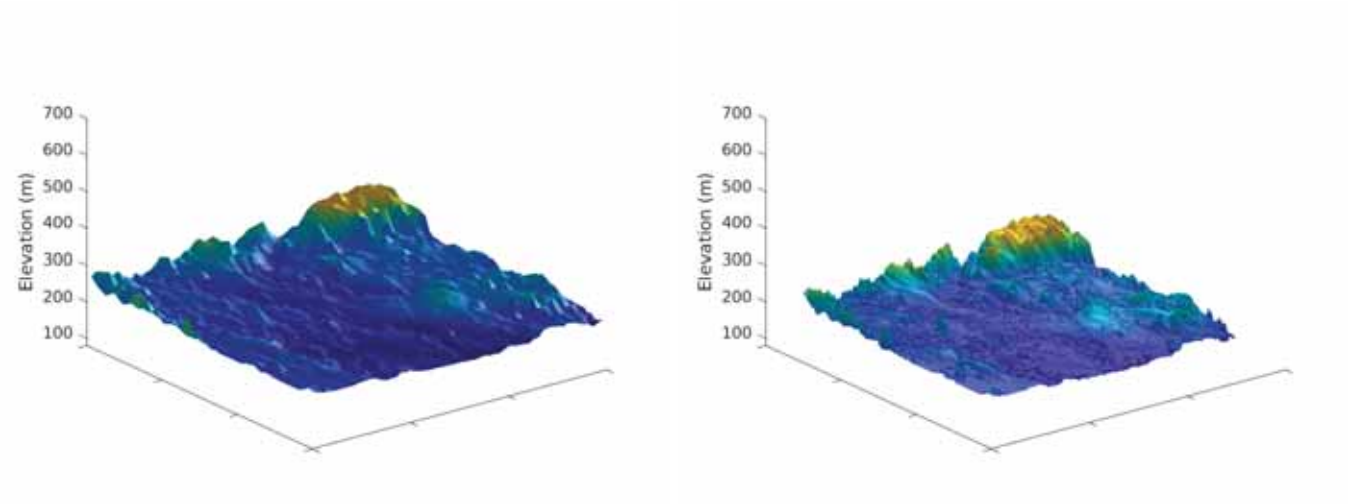

Figure 9. The terrain from Figure 4. Left: low-resolution 60×60. Right: high-resolution 2500×2500.

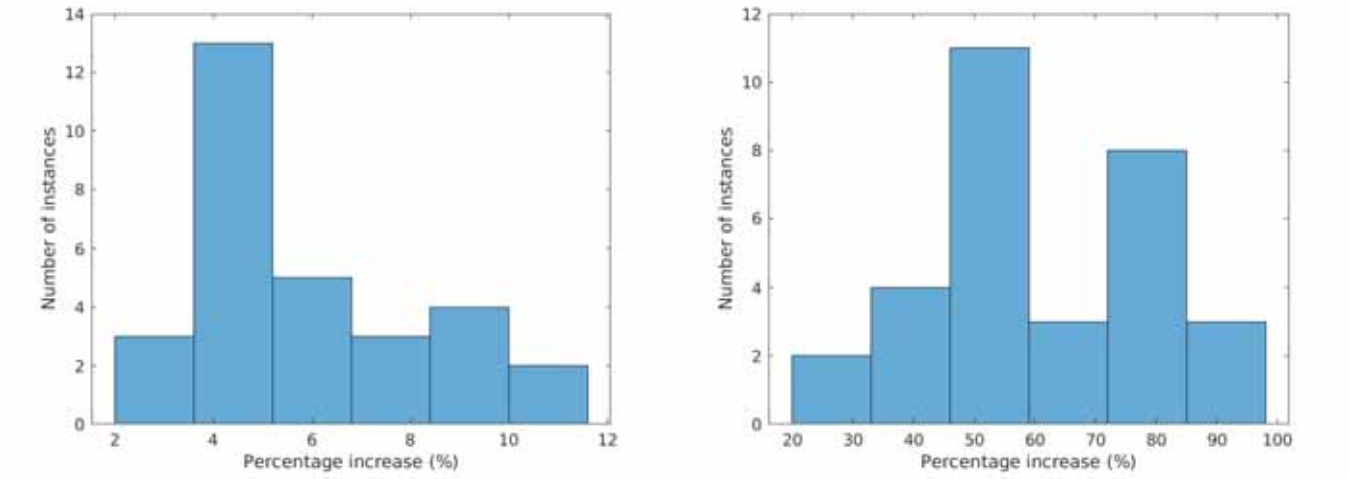

Figure 10. Histogram of the percentage of increase in the objective value obtained by heuristics compared to optimal solutions. Left: TERRAINZONING (the objective is the number of rectangles). Right: CITYZONING (the objective is the volume under the zones)

[11] E. Sunil, J. Ellerbroek, and J. M. Hoekstra, "Camda: Capacity assessment method for decentralized air traffic control," in *ICRAT2018*, 2018.

[12] P. D. Vascik, "Systems analysis of urban air mobility operational scaling," Ph.D. dissertation, MIT, 2020.

[13] J. Cho and Y. Yoon, "Extraction and interpretation of geometrical and topological properties of urban airspace for uas operations," in *13th USA/Europe Air Traffic Management Research and Development Seminar 2019, ATM 2019*. EUROCONTROL, 2019.

[14] A. Li and M. Hansen, "Obstacle clustering and path optimization for drone routing," in *ICRAT2020*, 2020.

[15] SESAR, "Exploratory research call," 2019. [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/other/call_fiches/jtis/h2020-call-doc-er4-sesar-ju_en.pdf

[16] P. Hullah, "UAS ATM Airspace Assessment," 2018. [Online]. Available: https://www.eurocontrol.int/sites/default/files/publication/files/uas-atm-airspace-assessment-v1.2-release-20181127.pdf

[17] P. D. Vascik, J. Cho, V. Bulusu, and V. Polishchuk, "Geometric approach towards airspace assessment for emerging operations," *Journal of Air Transportation*, pp. 1–10, 2020, special issue on ATM Seminar 2019.

[18] G. S. Brodal, P. Davoodi, M. Lewenstein, R. Raman, and S. R. Satti, "Two dimensional range minimum queries and fibonacci lattices," *Theoretical Computer Science*, vol. 638, pp. 33–43, 2016.

[19] Lantmäteriet, "Laserdata NH 2019 (laz)," 2019. [Online]. Available: https://www.lantmateriet.se/

[20] "Tetralith server, NSC, Linköping University," https://www.nsc.liu.se/systems/tetralith/.

[21] NYC DoITT, "NYC 3-D Building Model," 2014. [Online]. Available: https://www1.nyc.gov/site/doitt/initiatives/3d-building.page

[22] Open Source Geospatial Foundation Project, "QGIS Geographic Information System," 2020. [Online]. Available: http://qgis.org

[23] A. Fournier, D. Fussell, and L. Carpenter, "Computer rendering of stochastic models," *Commun. ACM*, vol. 25, no. 6, p. 371–384, Jun. 1982. [Online]. Available: https://doi.org/10.1145/358523.358553

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: https://science.sciencemag.org/content/220/4598/671

[25] M. S. Paterson and F. Frances Yao, "Optimal binary space partitions for orthogonal objects," *Journal of Algorithms*, vol. 13, no. 1, pp. 99 – 113, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/019667749290007Y

[26] European Commission, "Commission Implementing Regulation (EU) 2012/923: Laying down the common rules of the air," 2019. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/

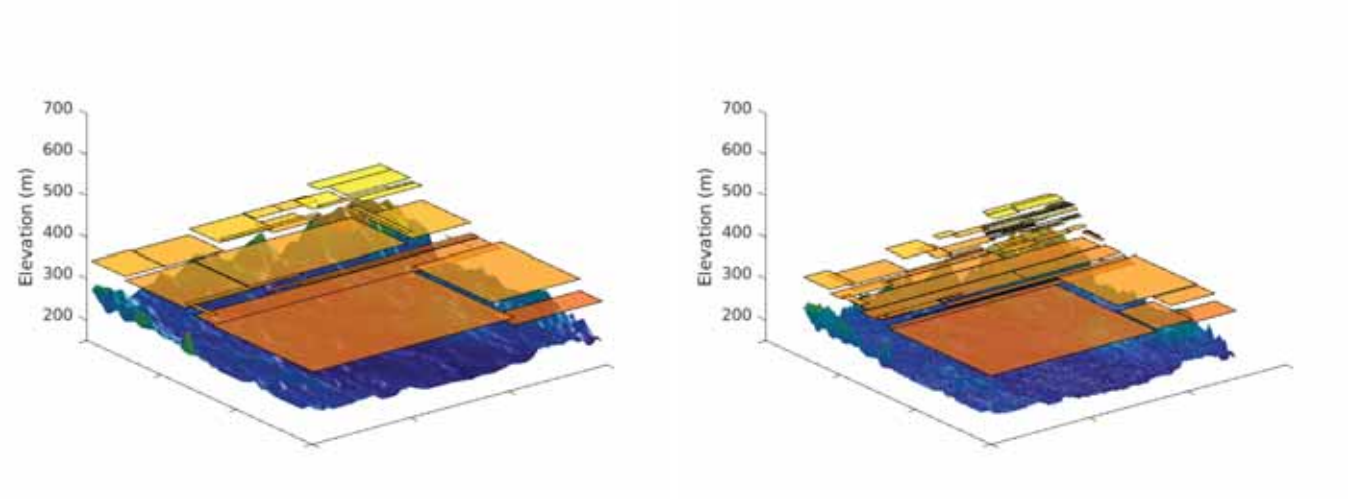

Figure 11. Solution for TERRAINZONING produced by the heuristic. The same terrain was used as in Figure 4. Left: low-resolution 60×60 terrain. Right: high-resolution 2500×2500 terrain.

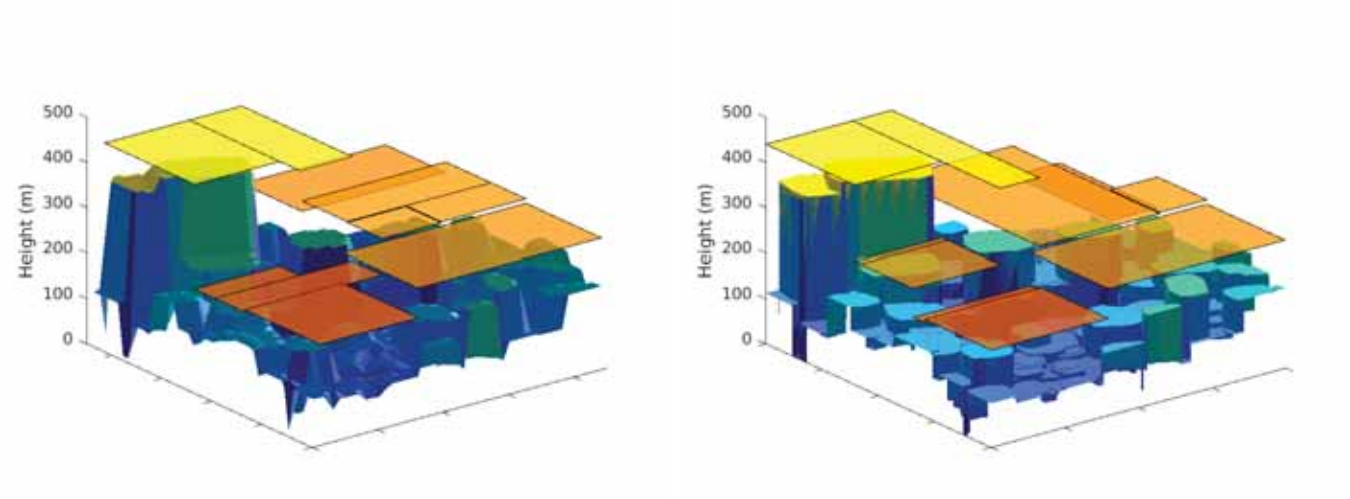

Figure 12. Solution for CITYZONING produced by the heuristic. The same terrain was used as in Figure 8. Left: low-resolution 45×45 terrain. Right: high-resolution 2000×2000 terrain.

?qid=1529918737045&uri=CELEX:02012R0923-20171012
[27] L. Sedov and V. Polishchuk, "Centralized and distributed utm in layered airspace," *8th ICRAT*, 2018.