Masters
Program
in **Geospatial
Technologies**

Rooftop-Place Suitability Analysis for Urban Air Mobility Hubs:
A GIS and Neural Network Approach

Carlos Javier Delgado Gonzalez

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

NOVA IMS
Information Management School

UNIVERSITAT JAUME·I

WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

Rooftop-Place Suitability Analysis for Urban Air Mobility Hubs:

A GIS and Neural Network Approach

Dissertation supervised by

Joel Dinis Baptista Ferreira da Silva, PhD
NOVA Information Management School
Universidade Nova de Lisboa
Lisbon, Portugal

and co-supervised by

Roberto André Pereira Henriques, PhD
NOVA Information Management School
Universidade Nova de Lisboa
Lisbon, Portugal

Carlos Granell Canut, PhD
GEOTEC
Universitat Jaume I
Castellón, Spain

February 2020

Declaration of Originality

I declare that the work described in this document is my own and not from someone else. All the assistance I have received from other people is duly acknowledged and all the sources (published or not published) are referenced.

This work has not been previously evaluated or submitted to NOVA Information Management School or elsewhere.

Lisbon, February 24, 2020

Carlos Javier Delgado Gonzalez

[the signed original has been archived by the NOVA IMS services]

Acknowledgments

First, I want to express my gratitude to the three supervisors, since without a doubt their contributions were crucial to the improvement of this thesis. I want to thank Dr. Joel Silva for his patient guidance, encouragement, and opportune support throughout the development of this research. I want to thank Professor Roberto Henriques because his office was always open to any concerns that arose in addition to his advice for enhancing the document. I would like to thank Professor Carlos Granell also for his pertinent feedback, advice, and motivation to consolidate the document in the best possible way.

Additionally, I also want to thank Professors Marco Painho and Sara Riveiro for also giving me their unconditional advice and support whenever I needed it. I want to thank my classmates, who also encouraged me to get ahead in this research, especially Itza, Maicol, Mihail, Moritz and Vicente.

I would like to state my sincere gratitude also to the Erasmus Mundus Master Program of Science in Geospatial Technologies headed by Professors Marco Painho, Christoph Brox, Christian Kray, Joaquin Huerta, and Michael Gould. Having received an Erasmus scholarship to be part of this wonderful program is one of the most rewarding experiences of my life.

Finally, I want to thank all my family, who has always believed in me. Especially to my parents Gloria and Carlos, because despite the thousands of kilometers that separate us, their constant support and words of encouragement were vital to give my best and thus achieve such desired goal.

“The application of GIS is only limited by the imagination of those who use it”

Jack Dangermond

Rooftop-Place Suitability Analysis for Urban Air Mobility Hubs: A GIS and Neural Network Approach

Abstract

Nowadays, constant overpopulation and urban expansion in cities worldwide have led to several transport-related challenges. Traffic congestion, long commuting, parking difficulties, automobile dependence, high infrastructure maintenance costs, poor public transportation, and loss of public space are some of the problems that afflict major metropolitan areas. Trying to provide a solution for the future inner-city transportation, several companies have worked in recent years to design aircraft prototypes that base their technology on current UAVs. Therefore, vehicles with electrical Vertical Take-Off and Landing (eVTOL) technology are rapidly emerging so that they can be included in the Urban Air Mobility (UAM) system. For this to become a reality, space agencies, governments and academics are generating concepts and recommendations to be considered a safe means of transportation for citizens. However, one of the most relevant points for this future implementation is the suitable location of the potential UAM hubs within the metropolitan areas. Since although UAM vehicles can take advantage of infrastructure such as roofs of buildings to clear and land, several criteria must be considered to find the ideal location.

As a solution, this thesis seeks to carry out an integral rooftop-place suitability analysis by involving both the essential variables of the urban ecosystem and the adequate rooftop surfaces for UAM operability. The study area selected for this research is Manhattan (New York, U.S), which is the most densely populated metropolitan area of one of the megacities in the world. The applied methodology has an unsupervised-data-driving and GIS-based approach, which is covered in three sections. The first part is responsible for analyzing the suitability of place when evaluating spatial patterns given by the application of Self-Organizing Maps on the urban ecosystem variables attached to the city census blocks. The second part is based on the development of an algorithm in Python for both the evaluation of the flatness of the roof surfaces and the definition of the UAM platform type suitable for its settlement. The final stage performs a combined analysis of the suitability indexes generated for the development of UAM hubs. Results reflect that 16% of the roofs in the study area have high integral suitability for the development of UAM hubs, where UAVs platforms and Vertistops (small size platforms) are the types that can be the most settled in Manhattan.

The reproducibility self-assessment of this research when considering Nüst *et al.* [45] criteria (https://osf.io/j97zp/) is: 2, 1, 2, 1, 1 (input data, preprocessing, methods, computational environment, results). GitHub repository code is available in https://github.com/carlosjdelgadonovaims/rooftop-place_suitability_analysis_for_Urban_Air_Mobility_hubs

Keywords

Artificial Neural Network
Catchment Area
Census Block
Cluster
Driving Distance
Electrical Vertical Take-off and Landing
Flatness
Geographical SOM
K-means
Light Detection and Ranging
Machine Learning
Neuron
Parallel Processing
Points-Of-Interest
Python
Rooftop
Self-Organizing Maps
Suitability Analysis
U-Matrix
Unmanned Aerial Vehicles
Urban Air Mobility
Vertihub
Vertiport
Vertistop

Acronyms

AHP	Analytic Hierarchy Process
ANN	Artificial Neural Network
ATC	Air Traffic Control
BMU	Best Matching Unit
EASA	European Aviation Safety Agency
eVTOL	Electrical Vertical Take-off and Landing
FAA	Federal Aviation Administration
FTP	Transfer Protocol Files
GDAL	Geospatial Data Abstraction Library
GeoSOM	Geographical Self-Organizing Maps
GIS	Geographical Information Systems
LAS	LASer format for point clouds
LIDAR	Light Detection and Ranging
MOD	Mobility On-Demand
NASA	National Aeronautics and Space Administration
OSM	Open Street Maps
PAV	Personal Air Vehicles
PM	Particulate Matter
POI	Points-Of-Interest
RANSAC	Random Sample Consensus
SOM	Self-Organizing Maps
SQL	Structured Query Language
UAM	Urban Air Mobility
UAS	Unmanned Aerial Systems
UAV	Unmanned Aerial Vehicles

Index of the Text

Acknowledgments .... **iv**

Abstract .... **vi**

Index of Tables .... **xii**

Index of Figures .... **xiii**

1. Introduction **1**

1.1 Problem statement and motivation 1

1.2 Research Questions 3

1.3 Objectives 3

1.4 Assumptions 3

2. Literature Review **4**

2.1 Urban Air Mobility 4

2.1.1 Special Factors for Urban Air Mobility 5

2.1.2 Urban Air Mobility platforms 6

2.2 Review of Place Selection Methods for UAM platforms 7

2.2.1 Place Selection Approach 7

2.2.2 Rooftop Shape Identification Approach 8

2.3 Self-Organizing Maps: SOM and GeoSOM 8

3. Methodology **11**

3.1 Data and Study Area 12

3.1.1 Study Area 12

3.1.2 Data Description 13

3.1.3 Data preprocessing 17

3.2 Place Suitability Analysis 18

3.2.1 Proximity Features Extraction 18

3.2.2 Clustering algorithms: SOM, GeoSOM, k-means 19

3.2.3 Classification of Place Suitability Levels 22

3.3 Rooftop-Flatness Assessment 23

3.3.1 Rooftop Elevation Statistics Calculation 23

3.3.2 Rooftop Flat Surfaces Extraction 24

3.3.3 Rooftop Suitability Categorization 25

3.4 Combined Categorization of the Suitability Indexes for UAM Platforms...26

3.5 Hardware and Software........27

4. Results and Discussion 29

4.1 Site Suitability Analysis........29

4.1.1 Driving Distance Calculation........ 29

4.1.2 SOM Clusters 31

4.1.3 GeoSOM Clusters 34

4.1.4 Visual Comparison of Cluster Results........ 37

4.1.5 GeoSOM Clusters Description........ 38

4.1.6 Place Suitability Levels Using GeoSOM Clusters........ 44

4.2 Rooftop Flatness Assessment Results 46

4.2.1 Rooftop Statistics Results 46

4.2.2 Rooftop Flatness Assessment Results........ 47

4.2.3 Rooftop Suitability Analysis 52

4.3 Final suitability categorization for UAM Platforms 55

4.4 Algorithm Performance Considerations 60

5. Limitations and Future Work 61

6. Conclusions........ 62

7. Bibliographic References 64

Annex 1........ 71

Annex 2........ 76

Annex 3........ 77

Annex 4........ 80

Annex 5........ 87

Index of Tables

Table 1. Environmental and socio-economic variables description............................13
Table 2. POIs and Rooftop Assessment inputs description.......................................14
Table 3. Area ranges proposed for the categorization of UAM platforms.................25
Table 4. Main software and libraries used for the place suitability analysis.............27
Table 5. Main software and libraries used in the rooftop suitability analysis...........28
Table 6. Example of new proximity attributes calculated by using Driving Distance algorithm for a specific census block...31
Table 7. Best input parameters configuration selected for SOM...............................32
Table 8. Summary table showing the distribution of census blocks in each place suitability index per GeoSOM clusters..46
Table 9. Table of results for the second part of the algorithm for the flatness rooftop assessment...48
Table 10. Total potentiality of the rooftops by counting the usable surfaces available 52
Table 11. Summary table showing the distribution of rooftops in each roof suitability index...55
Table 12. Distribution of the rooftops in the final categorization of suitability for UAM platforms...57

Annex 1

Table A- 1. Different parameter settings used for the sensitivity analysis of the SOM algorithm execution...71

Annex 2

Table B- 1. Distribution of integral suitability per UAM platform type...................76
Table B- 2. Distribution of integral suitability per GeoSOM clusters.......................76

Index of Figures

Figure 1. General Methodology 11
Figure 2. Study Area 12
Figure 3. Environmental variables geo-visualization 15
Figure 4. Socio-economic variables geo-visualization 15
Figure 5. POI variables geo-visualization 16
Figure 6. LIDAR tiles used in the rooftop flatness assessment 17
Figure 7. Example of the proximity features extraction process. (a) Selection of closest node to each POI and census block. (b) Selection of all possible nodes within 2km Driving Distance. (c) Selection of the shortest path to each POI type from the census block node 19
Figure 8. Example U-Matrix 5x5 neurons. (a) U-Matrix distances visualization (b) U-Matrix hints visualization 20
Figure 9. Strategy for the visualization of the initial clusters through cluster projection per neuron from a U-Matrix base-line 21
Figure 10. Scheme for the categorization of Place Suitability Levels for each of the n clusters calculated 23
Figure 11. Main steps for rooftop flat surfaces extraction 24
Figure 12. Example of attributes calculated by the algorithm in each of the main roof surfaces 26
Figure 13. Coverage area obtained after executed Driving Distance algorithm at a distance of 2km for a specific block 29
Figure 14. Visualization of the reachable POIs obtained after using Driving Distance algorithm 30
Figure 15. (a) SOM U-Matrix 10 x 15 obtained with best parameter configuration, (b) Reflected training pattern proportions from the 10 base-line clusters 32
Figure 16. SOM Box-Histo for the 10 base-line clusters 33
Figure 17. U-Matrix generated for the new 8 SOM clusters 33
Figure 18. Box-Histo of the normalized variables for each of the 8 SOM clusters 34
Figure 19. Manhattan census blocks clustered by SOM algorithm 34
Figure 20. (a) GeoSOM U-Matrix 10 x 15 obtained with best parameter configuration, (b) Reflected training pattern proportions from the 10 base-line clusters 35
Figure 21. GeoSOM Box-Histo for the 10 base-line clusters 35
Figure 22. U-Matrix generated for the new 8 GeoSOM clusters 35
Figure 23. Box-Histo of the normalized variables for each of the 8 SOM clusters 36
Figure 24. Manhattan census blocks clustered by GeoSOM algorithm 36
Figure 25. Map comparison from clusters generated using SOM, GeoSOM, and k-means 37

Figure 26. Bar-charts non-normalized variables (a) population density (b) population 40
Figure 27. Bar-charts non-normalized variables (a) Percentage of workers that commute more than 60 minutes one-way, (b) Area km^2 41
Figure 28. Bar-charts non-normalized variables (a) Road-traffic and air-traffic noise, (b) Air-quality 41
Figure 29. Bar-charts non-normalized variables (a) Median household incomes, (b) Medium Gross Rent 41
Figure 30. Comparative bar-charts for the number of POI present in each GeoSOM cluster 42
Figure 31. Comparative box-plots for the minimum distance to POI in each GeoSOM cluster 43
Figure 32. Place suitability map for Manhattan 44
Figure 33. Census blocks proportions of place suitability levels per GeoSOM clusters 45
Figure 34. Distribution of census blocks in each place suitability index per GeoSOM clusters 45
Figure 35. Maps of rooftops height statistics 47
Figure 36. Roof 1 (a) Lidar image with main surface, (b) satellite image, (c) 3D scene 49
Figure 37. Roof 4 (a) Lidar image with main surface, (b) Satellite image, (c) 3D scene 49
Figure 38. Roof 7 (a) Lidar image with main surface, (b) Satellite image, (c) 3D scene 49
Figure 39. Roof 2 (a) Lidar image with main surface, (b) Satellite image, (c) 3D scene 49
Figure 40. Roof 5 (a) Lidar image with main surface, (b) Satellite image, (c) 3D Scene 50
Figure 41. Map distribution for UAM Platforms typology using the main surface level 51
Figure 42. Waffle-chart for UAM typology distribution at different surface levels ...51
Figure 43. Suitability Index map for UAV platforms 53
Figure 44. Suitability index map for Vertistops 53
Figure 45. Suitability index map for Vertiports 54
Figure 46. Suitability index map for Vertihubs 54
Figure 47. Distribution of the suitability index levels for each UAM platform type.55
Figure 48. Combined suitability taking the criteria of place and flatness of the main surface on the roofs 56
Figure 49. Total of rooftops distributed for each combined suitability class....57
Figure 50. Proportions of combined suitability levels per UAM hub types....58
Figure 51. Proportions of combined suitability levels per GeoSOM cluster 59
Figure 52. UAM types distribution per GeoSOM clusters 59

Figure 53. Operational mode when running subprocess library for the python algorithm 60

Annex 1

Figure A- 1. Component Planes from SOM algorithm part 1 72
Figure A- 2. Component Planes from SOM algorithm part 2 73
Figure A- 3. Component Planes from GeoSOM algorithm part 1 74
Figure A- 4. Component Planes from GeoSOM algorithm part 2 75

1. Introduction

1.1 Problem statement and motivation

The growing overpopulation and urbanization in metropolitan cities in the world have generated several transport-related challenges in the field of urban mobility [49]. Furthermore, the proliferating private car dependency and high levels of spatial aggregation of economic activities are leading to significant traffic congestions [16, 47]. Additionally, recent Mobility On-Demand (MOD) schemes such as ride-sharing and car-sharing as those implemented by Uber and Lyft, have contributed to increasing levels of traffic density on some urban areas while trying to solve limited parking lots issues [16, 18]. Consequently, inner-city commute time has been meaningfully affected, where commuters in cities like Chicago, New York, and Los Angeles, on average, lose 97 hours a year due to traffic jams [29]. Several alternatives are proposed to relieve traffic congestions such as non-motorized mobility and promoting the use of public transportation. Nevertheless, these options require new infrastructures which are limited by the lack of space in densely-populated urban areas [19, 48]. In this way, all the above alternatives are summarized in solutions focused on a two-dimensional matrix [9].

As a novel solution, Electrical Vertical Take-off and Landing technology (eVTOL) aircraft are being designed and tested to become a feasible means of transport to inner-city commuting [9]. The new approach converges on a path that contemplates the three-dimensional space, which had previously been explored by the development of Unmanned Aerial Vehicles (UAV) [43, 44]. New elements are then added to the concept of Urban Air Mobility (UAM), which seeks to optimize inner-city commuting-times by avoiding existing traditional impedances generated on road networks [48]. UAM scope contemplates not only the transport of passengers but also freight modalities to open the door for multiple purposes [44]. Besides, the features and specifications of the new eVTOL vehicles are quite promising for their sustainability in an urban environment. Concepts issued by the manufacturers and consultancies show notable differences between an eVTOL aircraft and a traditional helicopter. eVTOL vehicles will be four times quieter, twice safer and ten times less expensive to build than traditional helicopters [48]. While several companies and investors work on the development of new, efficient, and comfortable prototypes of eVTOL vehicles, leading space agencies around the world generate regulatory policies for the use of these vehicles in UAM system [48]. National Aeronautics and Space Administration (NASA), European Aviation Safety Agency (EASA), and Federal Aviation Administration (FAA) are some of the entities that are currently declaring concepts for the proper implementation of UAM [40, 55, 64]. Process certification, battery technology, vehicle efficiency, performance and reliability, air traffic control, cost and affordability, safety, noise, emissions, pilot training, and infrastructure in cities summarize the challenges to overcome towards UAM implementation [27]. Subsequently, in recent years the interest of academic and scientific communities in different fields has aroused to contribute to the implementation of UAM. Geospatial technologies have been involved in this contribution, especially when creating 3D geofences for the design of inner-city flight route networks [25] and finding suitable locations for UAM infrastructures [11, 15]. The latter excels in the complexity it handles by involving a synergy of different urban, social, economic and environmental concepts, which play an important role in defining potential UAM hubs locations. Notwithstanding, the fact that UAM concepts are addressed to be applied in densely populated urban areas for the improvement of transport quality leads to add an evaluation of the infrastructure present in the city. Therefore, the lack of sufficient space in metropolitan areas implies taking into account the typology in the UAM platform designs and sizes for its construction on the rooftops of buildings.

Although not much work has been developed on this subject, some academics have contributed by applying and evaluating geo-scientific approaches in search of this ideal location. A suitability analysis was generated for the cities of Los Angeles (United States) and Münich (Germany) under an expert knowledge approach [15]. Therefore, after applying an Analytic Hierarchy Process (AHP), the consensus of experts allowed us to evaluate the importance of the social-economic and environmental variables involved [15]. The resulting map reflects regions of suitability where UAM ground structures can be located. However, an exact location was not calculated by not considering physical availability structures for UAM platforms. In contrast, another approach to the location of suitable UAV sites for landing and clearing was performed when considering the physical structure of the roofs. The study was based basically on the classification rooftop images generated both from Light Detection and Ranging (LIDAR) data and using satellite images [11]. Machine-learning classification techniques were used to distinguish different roof shapes in training samples from three cities that were manually labeled. Results of the classification allow separating seven types of roof shapes where those that are flat are the most feasible for UAV landing and take-off [11]. Despite good results, this approach does not include urban socio-economic and environmental aspects, and also it is not extending the scope to evaluate the total rooftops use according to area specifications to build different UAM types.

In this order of ideas, this research seeks to establish an unsupervised-data-driven and GIS-based methodology to find out potential locations for the development of UAM hubs in Manhattan, New York City (U.S). The proposed methodological process aims to increase the level of detail in the location of potential UAM hubs by considering both a place suitability analysis given by the urban ecosystem variables and rooftop suitability analysis when evaluating physical characteristics of flatness and area available to delimit UAM platforms types.

1.2 Research Questions

This thesis formulated the following research questions:

- *Where are the most suitable places to develop UAM hubs in the study area?*

- *What is the potential suitability of the rooftops in the study area for the development of different UAM platforms types?*

- *Is it possible to increase the level of detail in the location of suitable places to develop UAM hubs in the study area?*

1.3 Objectives

In order to achieve the formulated research questions, this thesis proposed the following specific objectives:

- Maximize the information provided by input variables for the place suitability analysis when extracting proximity features from urban Points of Interest (POI) through routing algorithms.

- Design and implement a place suitability analysis when applying Self-Organizing Maps to identify clusters with similar urban ecosystem patterns.

- Develop an algorithm to estimate the suitability of the roofs when evaluating the main existing flat surfaces and the potential UAM platform type that can be settled on them.

- Examine the results obtained by matching both proposed approaches.

1.4 Assumptions

The following assumptions were assumed for the development of this thesis:

- Up until now, a UAM system has not been officially implemented in any city in the world. Therefore no UAM structure has been materialized at present.

- The feasibility of the development of UAM hubs is based on considerations issued by space agencies, consultancies, manufacturers and academics who are exploring the future of UAM as an inner-city transport system.

- The material from which the roofs are constructed was not considered.

2. Literature Review

2.1 Urban Air Mobility

Urban Air Mobility (UAM) is a *“new”* concept that seeks to gather all those air passengers and cargo-carrying transport systems within metropolitan areas by means of both manned and unmanned aircraft [44]. Nevertheless, it is a term that has been the evolution of several projects and ideas developed before. Concepts such as Personal Air Vehicles (PAV) and On-Demand Mobility (ODM) were mentioned by National Aeronautics and Space Administration (NASA) in 2006 [15, 40, 41]. Around 2007, thanks to the ODM idea, the concept of air-taxis was already emerging as a response to the increase in traffic congestion on the roads [27, 40]. NASA and Federal Aviation Administration (FAA) even began to formulate feasibility studies for the air-taxi market and factors related to Air Traffic Control (ATC) [40]. Its applicability has depended on the technological advances achieved especially in the field of Unmanned Aerial Systems (UAS). Today the reliability given by Unmanned Aerial Vehicles (UAVs) has been the reason for its proliferation in multiple purposes such as hobbyists, cargo-services, emergency care, defense, etc. Nevertheless, regulatory considerations for both UAM and UAS are being designed to improve the ATC [59].
Regarding the passenger transport in the UAM scheme, there are several academics, industries, and governments that are strengthening and promoting the sustainability of this inner-city and intra-city transport system as a new alternative of mobility [44]. Cutting-edge technological advances in the field of electronics and aeronautics have led to the development of designs and prototypes based on electrical Vertical Take-Off and Landing (eVTOL) technology. Several prototypes of eVTOL aircraft are being designed and tested in order to guarantee the four key realms of the vertical mobility ecosystem: *aircraft systems, certification and law, social acceptance and infrastructure* [48]. Ehang, Volocopter, Airbus Vahana, and SureFly have been the pioneers developing eVTOL aircraft prototypes since 2015 and testing them with successful flights in 2017 and 2018 [1, 2, 48]. Up until the development of this research, the most recent test recorded by Volocopter was at Helsinki airport in August 2019 [10, 64]. Another important manufacturer of eVTOL aircraft is the German startup Lilium. They have designed a Jet-type air-vehicle called LiliumJet which was first tested in May 2019. Its design is also based to fulfill with EASA and FAA standards [35, 46]. Amazon with Prime Air [30] design and Boeing giant cargo drone are exploring and testing new designs for the delivery of goods of different sizes. This adds to the many other uses that UAVs have nowadays. In summary, there is evidence of a variety of companies working hard to develop prototypes that meet the requirements of a vertical mobility ecosystem.

2.1.1 Special Factors for Urban Air Mobility

2.1.1.1 Socio-economic factors

One of the realm keys of the UAM is social acceptance towards the project [28, 48]. Therefore the community must understand the benefits and problems that can cause the implementation of a new mode of transport in the city. UAM tries to minimize this impact by considering that this new mobility model does not physically replace existing mobility structures [9]. In addition, some of the existing infrastructures will serve as a pivot for the realization of future inner-city mobility [9, 27]. Thus, the location of hospitals, metro stations, bus stops, trade centers, green areas and other Points-of-interest (POI) will work to establish the appropriate geospatial environment for future UAM infrastructures. Population must also know what are the mechanisms that will guarantee a safe and reliable means of transport [48]. Furthermore, the population needs to be a participant in the definition of routes and hubs. 3D geofences have been simulated by Hildemann et *al.* [25] reflecting possible conditioned areas for air-taxis transit. The delimitation of areas with high population density and job density are relevant to find the communities that can benefit from UAM [15]. In this way, not only potential users of cities should be considered in the studies, but also all citizens who participate in the inner-city mobility in general [27, 48]. Sustainability studies of the UAM show favorable results especially for the passenger market [48] however, affordability will play an important role at the beginning of the implementation of UAM system. Therefore, spatially identifying variables related to income, job density, and mode of transport used to get job place could support the location of UAM infrastructure [15, 48]. Additionally, it is relevant to understand the cost of both monetary and time travel of commuters using similar transport systems. One of the most viable market schemes is the operation of air taxis under an on-demand approach [43]. Nevertheless, there might be other schemes which interest investors. Airport shuttle, air-ambulance, police, company shuttle and office-to-office travels, cargo-delivery are some of the potential UAM markets [43]. Another interesting approach was established by Fadhil 2018 [15] by involving office rental prices as an estimation for business trips. Potential demand for UAM could be located where rental prices are higher, leading to probable settlements of UAM structures [15].

2.1.1.2 Environmental factors

Noise pollution is one of the most worrisome variables when implementing eVTOL vehicles [9, 27, 48, 56]. Although according to the concepts, eVTOL aircraft generate less noise than helicopters, there will be an increase in noise levels in certain parts of the city [48]. Nevertheless, manufacturers and experts estimate that noise levels may be similar to those that a half-size truck can emit [27]. An eVTOL aircraft 90 meters from the ground can emit about 63dB measured at ground level [27, 48]. This could be considered "acceptable", but there are other implications that should be studied.

Therefore, this leads to recognizing the relevance of the noise variable in geospatial analysis for UAM infrastructure settlement. An important idea to mitigate noise was designed by Ancliff *et al.* [4] when trying to enhance paths for eVTOL aircraft when considering areas with high existing noise. In this way, highways and main roads of the city could function as noise absorption zones caused by the propulsion of the vehicles [4]. Consequently, areas with high emissions of particulate matter (PM) will indicate the presence of hot-spots where the development of UAM systems by implementing eVTOL technology would improve the urban air quality [43, 52]. Other environmental factors may be related to the presence of birds [56] and weather conditions that would limit the traffic of UAM vehicles.

2.1.2 Urban Air Mobility platforms

Keeping a successful business model for UAM requires an adequate infrastructure for the operation of eVTOL vehicles [9, 48]. Not only landing and take-off platforms should be considered, but battery charging infrastructures, ATC for surveillance and maintenance areas should be considered [9, 60]. Thus, the objective is to find strategic places for the development of both ground-based and rooftop structures for landing and take-off [9]. Notwithstanding, adverse conditions in highly populated cities can make it difficult to analyze, especially due to the availability of space [60]. Several concepts have been stated to classify the type of infrastructure necessary for the definition of the UAM infrastructure network. Uber Elevate [27] refers to Vertiports and Vertistops as the main types of infrastructure needed for eVTOL operations. Vertiports are considered as areas with multiple landing and clearance platforms for eVTOL vehicles [27, 43, 48]. In addition, they must have enough space to establish facilities that support maintenance, recharging, and staff [27, 48]. Uber Elevate [27] estimates Vertiports would have a maximum capacity of 12 eVTOL aircraft taking as an example the current heliports in New York. On the other hand, Vertistops are considered as a platform with a single pad for the landing and clearing of an eVTOL [48]. According to Uber Elevate [27] the advantage of these platforms is given by the fast embarkation and disembarkation of passengers but without the presence of complex support facilities. Fadhil [15] designed another category of a platform called Vertihub, which is considered as the biggest. These platforms could also have facilities for repair and maintenance, in addition, they would serve as parking for eVTOL vehicles [15]. Regarding UAVs there are no fixed specifications for the size of landing pads for traditional UAVs; however, portable landing pad designs range from three square meters depending on the drone size.

2.2 Review of Place Selection Methods for UAM platforms

Although several studies and investigations related to site selection analysis or place suitability analysis have been developed, only a few have been applied to find out the location of UAM platforms specifically. However, a bibliographic review has identified two approaches, the first one given by the urban environment conditions of the place and the second one under physical roof characteristics as potential landing and clearance spots of UAM vehicles.

2.2.1 Place Selection Approach

Regarding UAM ground infrastructure, Fadhil [15] developed his research by implementing a suitability analysis under an Analytic Hierarchy Process (AHP) methodology with a Delphi analysis. Therefore, his analysis involved an expert knowledge approach to generate suitability maps for UAM infrastructure in the cities of Los Angeles and Münich [15]. Even though the generated maps allow to identify regions with different levels of suitability, the exact location where UAM platforms can be established is not defined [15]. Similar studies applied not specifically to UAM location stations but to nodes of transport networks have also been included in this review. Multi-Criteria and Single-Criterion models were analyzed by Viera [63] in the solution of Hub Logistic Problems. Within the Multi-Criteria models analyzed by Viera [63], methods such as AHP, Fuzzy sets, Weighted Sum, Topsis, Genetic algorithms among others are applied. Apart from expert knowledge approach methods such as those already mentioned, data mining techniques have been applied to find patterns in regions of urban areas that can later be categorized [39]. Strategic site locations have been carried out using Self-Organizing Maps (Chapter 2.3) to find co-location patterns when using road networks on location-based services approach [66].

Spatial accessibility analysis is another important component in site location. Therefore, the evaluation of travel distances between points of interest directly influences urban planning [34]. Calculation of Euclidean distance, walking distance, bicycle distance, and *Driving Distance* are some of the methods that Geographical Information Systems (GIS) applies to determine characteristics of proximity and optimal route [34]. *Driving Distance* has been used to show disparities and deficiencies in the inner-city public transport, which is a starting point for urban planning and city design [34, 51]. Smart cities look for transport nodes to have rapid accessibility to citizens by increasing connectivity in travel modes [8, 14, 38]. The design of transport nodes and urban route planning have been developed by applying *Driving Distance* algorithms directly from databases. Although some GIS applications embed routing tools, database add-ons such as *pgRouting* for *PostgreSQL* databases run robust routing analysis when using Open Source road networks [17]. Thus, the quality of road networks influences the speed and accuracy of the results. Open Street Maps

(OSM) and contributing groups such as *Geofabrik* offer free sufficient data from road networks with node attributes, edges, cost and other impedances in the road system. *Driving Distance* algorithm from *pgRouting* library usually calculates distances when implementing the *Dijkstra* algorithm, which allows generating a realistic simulation of the situation [36]. Algorithms have not only been used to calculate optimal routes but also to consider coverage areas or catchment areas based on distances [20].

2.2.2 Rooftop Shape Identification Approach

An interesting approach was the one developed by Castagno *et al.* [11] when implementing a methodology for the detection of suitable roofs so that UAV can land in case of emergency. Although this approach was not properly designed for eVTOL aircraft, it takes advantage of the use of high-resolution satellite images and 3D point clouds (LIDAR) to perform a rooftop shape classification [11]. Castagno *et al.* [11] methodology, takes rooftop LIDAR data which is transformed into RGB raster files by tripling the elevation information. Subsequently, the rasters are the input for a Convolutional Neural Network and Random Forest (Machine-Learning approach) algorithms that classify the images according to the shape features. However, the labeling of training samples is done manually [11]. Castagno's research leads to that image classification performed with LIDAR data generates better results than using satellite images, but a combination of the two approaches increases accuracy [11]. Despite obtaining a good classification of the roof shape, suitability for UAV landing pads is limited to finding flat roofs. There are other approaches to identify roof planes, nevertheless these are not aimed at assessing the flatness for landing and clearance of UAM platforms, but breaking down the planes that make up the roof. Specific algorithms such as Region-Growing and Random Sample Consensus (RANSAC) have been implemented by Albano [3] and Chen *et al.* [12]. Clustering techniques have also been used for plane roof segmentation, therefore *k-means* and *fussy k-means* have been applied for roof reconstruction [54]. Cross-line Element Growth is also a technique has been provided by Wu *et al.* [65] for fast and accurate delineation of planes under airborne LIDAR data.

2.3 Self-Organizing Maps: SOM and GeoSOM

Self-Organizing Maps (SOM) is one of the most recognized types of Artificial Neural Network (ANN) created by Kohonen which is designed for the extraction and visualization of patterns employing an unsupervised learning process [5, 31]. Therefore, SOM is considered as a Machine-Learning method that provides solutions to problems that are modeled under a data-driven approach [33]. Analysis of patterns and structures are achieved after SOM performs a data reduction task by projecting high-dimensional data to a lower space that is usually two-dimensional or three-dimensional [22, 42]. The fact that the output space resulting from data reduction preserves topological relationships makes SOM a more powerful method compared to others *i.e k-means*; in addition to allowing it to be applied for different purposes such as mining data, data visualization, clustering, and classification [5, 6, 32]. The essential idea of SOM is that the output space that is usually a two-dimensional grid composed of units called neurons, can match the input patterns presented in the input space (training patterns) for the establishment of the Best Matching Unit (BMU). The BMU is the result of an iterative process where randomly an input pattern is presented to each of the SOM neurons, then distances are calculated, and the closest one is assigned as BMU [22]. Formula (1) explains the process to find the BMU, where x is the input vector, $\| . \|$ it is generally Euclidean distance, m_c corresponds to the SOM neuron and m_i to the BMU neuron.

$$\|x - m_c\| = min_i\{\|x - m_i\|\} \qquad (1)$$

After BMU is found all SOM neurons are updated and moved closer to training patterns in each iteration. The SOM update is given by the formula (2), where $\alpha(t)$ is the learning rate in specific t time and $h_{ci}(t)$ is the neighborhood function around the BMU unit c.

$$m_i = m_i(t) + \alpha(t)h_{ci}(t)(x - m_i) \qquad (2)$$

There are several neighborhood functions between them: *Bubble, Gaussian* and *Cutgass* [22]. These neighborhood functions include another parameter called the neighborhood radius, which along to the learning rate decreases in each SOM update. Therefore, SOM is increasingly adjusting to training patterns by narrowing BMUs and their topological neighbors [22]. The SOM training process is carried out in two phases, unfolding and fine-tuning. Some of the SOM parameters are configured depending on the problem being evaluated; however, Henriques [22] explains some special considerations before the algorithm is executed. SOM quality can be evaluated through two errors, quantization error and topographical error [22, 32]. The first one is more oriented to measure the adaptive capacity of the neural network, while the second one assesses the topology preservation of the SOM [22].

On the other hand, GeoSOM is a variation of the SOM algorithm considering the natural geographical component of training patterns [22, 23]. In this way, the GeoSOM concept affects the BMU search for a certain training pattern by considering only geographically close neurons [7, 23]. Therefore, the entire selection process for the winning unit selection is carried out in two steps: Definition of the geographic neighborhood and final search when considering remaining multidimensional components [7, 24]. BMU search is performed when the parameter *"geographical tolerance"* *k* is set to zero (*k=0*), which forces the BMU to be that unit that is geographically closest [24]. The fact that *k* is equal to zero describes a different scenario than when *k* increases its value since the search radius is amplified, and when *k* reaches the map size, a basic SOM then executed [7, 24]. Thus, GeoSOM outputs will have similar behavior to SOM training but only considering the spatial coordinates, which leads to each neuron working as a *"low-pass filter"* for other non-geospatial variables [23]. Several visualizations modes will then allow verifying that the GeoSOM approach yields the creation of clusters composed of geographically contiguous areas by forcing units that are close in the output space to be close in the input space [7, 24]. It is remarkable to mention that GeoSOM algorithm not only evaluates spatial homogeneity when considering spatial autocorrelation but also heterogeneous areas that although geographically may be close, non-geographic attributes might depict low correlation [24]. Furthermore, GeoSOM clusters can be used for different purposes in many thematic fields, such as urban and environmental planning [24].

3. Methodology

This chapter of the thesis introduces a description of the study area and the data used to carry out the investigation. Besides, to explain each of the main processes performed to carry out a rooftop-place suitability analysis for the development of UAM platforms. The overall methodology is separated into two sections; the first one is focused on establishing the place suitability analysis, while the second one seeks to assess the suitability of the rooftops. Once the results of both sections have been obtained, a combined suitability analysis is executed (*Figure 1*).

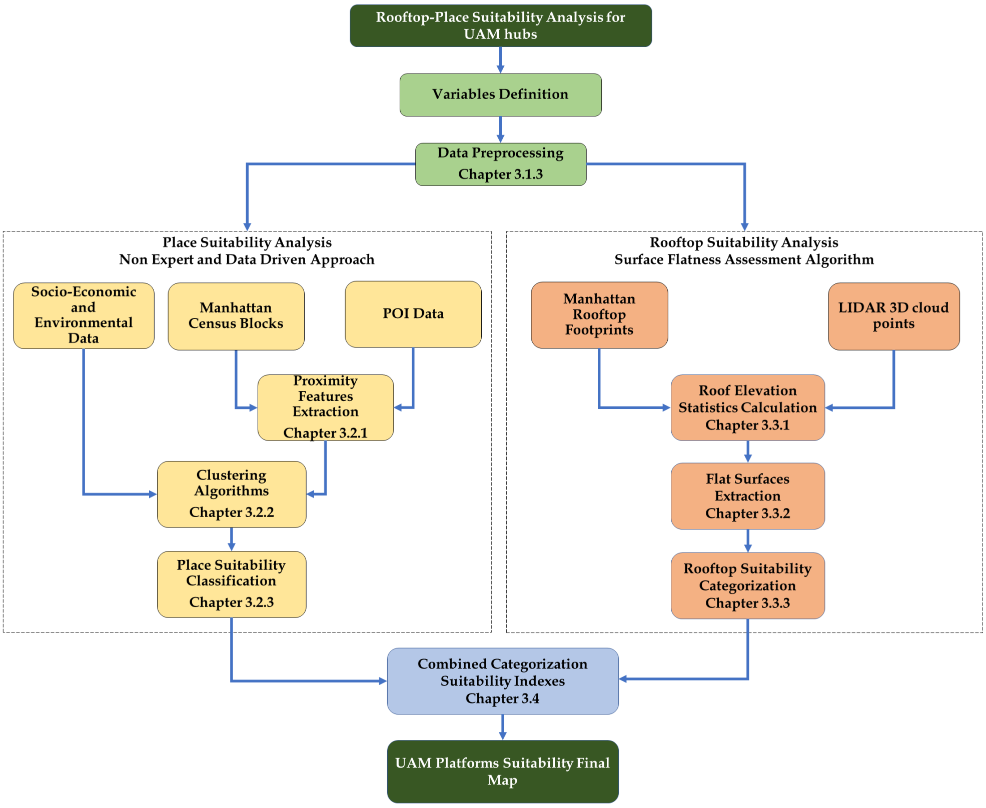

Rooftop-Place Suitability Analysis for UAM hubs
Variables Definition
Data Preprocessing Chapter 3.1.3
Place Suitability Analysis
Non Expert and Data Driven Approach
Socio-Economic and Environmental Data
Manhattan Census Blocks
POI Data
Proximity Features Extraction Chapter 3.2.1
Clustering Algorithms Chapter 3.2.2
Place Suitability Classification Chapter 3.2.3
Rooftop Suitability Analysis
Surface Flatness Assessment Algorithm
Manhattan Rooftop Footprints
LIDAR 3D cloud points
Roof Elevation Statistics Calculation Chapter 3.3.1
Flat Surfaces Extraction Chapter 3.3.2
Rooftop Suitability Categorization Chapter 3.3.3
Combined Categorization Suitability Indexes Chapter 3.4
UAM Platforms Suitability Final Map

Figure 1. General Methodology

3.1 Data and Study Area

3.1.1 Study Area

The research study area focuses on the metropolitan district (borough) of Manhattan, in New York City, northeast of the United States. The borough includes the Manhattan island and other additional islands such as Roosevelt, Randall, Governor, among others. It has approximately 1.6 million inhabitants and an area close to 59.1 km^2; therefore, it is the most densely populated district in the entire city. It concentrates the majority of economic and financial activities of the city, in addition to having multiple historical and tourist sites.

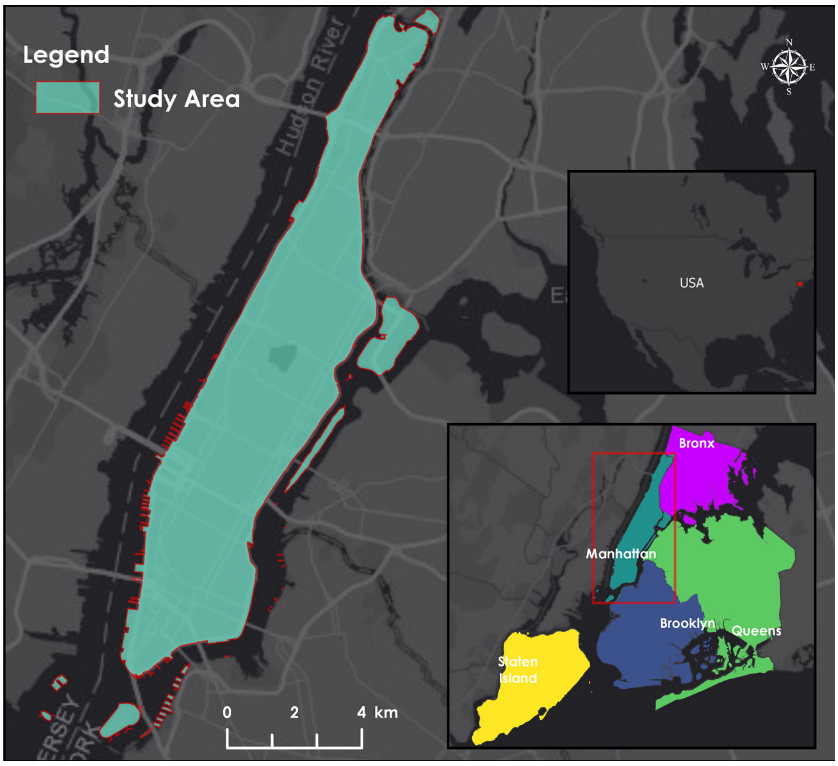


Figure 2. Study Area

3.1.2 Data Description

Most of the data used in the research try to align with the social, economic and environmental factors described in Chapter 2.1.1 for the ideal location of UAM platforms. *Table 1* and *Table 2* describes the main variables and data involved in both place suitability analysis and rooftop suitability analysis. Maps of the environmental variables are shown in *Figure 3*, socio-economic variables in *Figure 4* and Points of Interest (POI) in *Figure 5*. In addition, *Figure 6* shows the grid of the LIDAR files (tiles) available for the rooftop flatness assessment algorithm. Due to the high density of 3D point clouds, only a small section is depicted.

Type/Analysis	Variables	Description	Source
Environmental variables for Place Suitability Analysis	Air-traffic and road traffic noise	Format: Raster Units: Decibels (dB)	U.S. Department of Transportation 2017. National Transport Noise Map
	Air-quality (annual average mcg per cubic meter of fine particulate matter)	Format: Table per community district. Units: mcg PM 2.5	NYC Department of Health and Mental Hygiene, Environmental and Health Data Portal. 2013. 3rd Edition
Socio-economic variables for Place Suitability Analysis	Population	Format: Shapefile per Census block. Units: Number of people	NYC Open Data, 2010. Census Block Statistics.
	Population density	Format: Shapefile per Census block. Units: Number of people/km^2	NYC Open Data, 2010. Census Block Statistics.
	Commute 60 Minutes Or More One Way	Format: Table per census tract. Units: % of workers	Census Bureau, American Community Survey. 2018 table B08012
	Median household incomes	Format: Table per census tract. Units: $US	Census Bureau, American Community Survey. 2018 table B08013
	Median gross rent	Format: Table per census track. Units: $US	Census Bureau, American Community Survey. 2016 table B25064
	Census blocks	Format: Shapefile per Census block. (Polygon)	NYC Open Data 2010. Census Block Statistics.

Table 1. Environmental and socio-economic variables description

Purpose	Variables	Description	Source
Point of Interest for Place Suitability Analysis	Hospitals	Format: Shapefile (Polygon)	Open Street Maps 2018.
	Health centers	Format: Shapefile (Point)	NYC Open Data 2011.
	Parks	Format: Shapefile (Polygon)	Open Street Maps 2018.
	Graveyards	Format: Shapefile (Polygon)	Open Street Maps 2018.
	Schools/Universities	Format: Shapefile (Polygon)	Open Street Maps 2018.
	Touristic places	Format: Shapefile (Point)	NYC Open Data 2018.
	Public safety	Format: Shapefile (Point)	NYC Open Data 2017.
	Government facilities	Format: Shapefile (Point)	NYC Open Data 2017.
	Embassies	Format: Shapefile (Point)	Open Street Maps 2018.
	Parking lots	Format: Shapefile (Polygon)	NYC Open Data 2018.
	Malls	Format: Shapefile (Point)	Open Street Maps 2018.
	Subway stations	Format: Shapefile (Point)	NYC Open Data 2019.
	Bus stops	Format: Shapefile (Point)	NYC Open Data 2018.
Rooftop flatness assessment inputs	Rooftop building footprint	Format: Shapefile (Polygon)	NYC Open Data 2018.
	Airborne LIDAR 3D cloud points	Format: LAS (LASer) Tiles	USGS New York City. Woolpert Organization from GIS.NY.GOV 2014.

Table 2. POIs and Rooftop Assessment inputs description

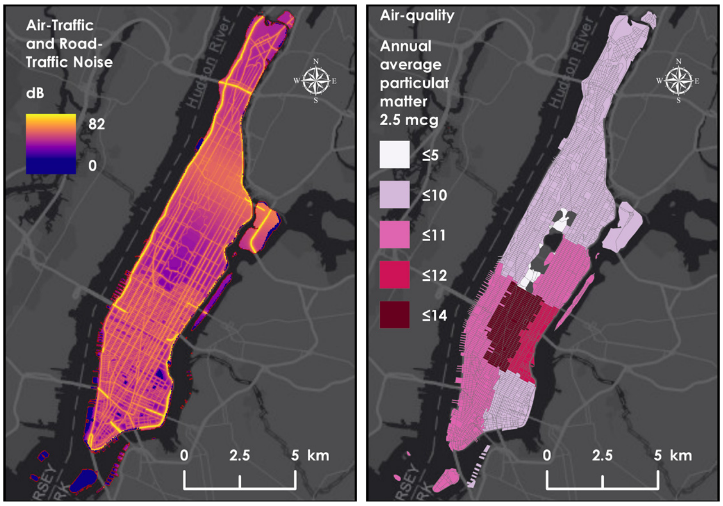

Air-Traffic and Road-Traffic Noise
dB
82
0
Hudson River
0 2.5 5 km
Air-quality
Annual average particulat matter 2.5 mcg
≤5
≤10
≤11
≤12
≤14
Hudson River
0 2.5 5 km

Figure 3. Environmental variables geo-visualization

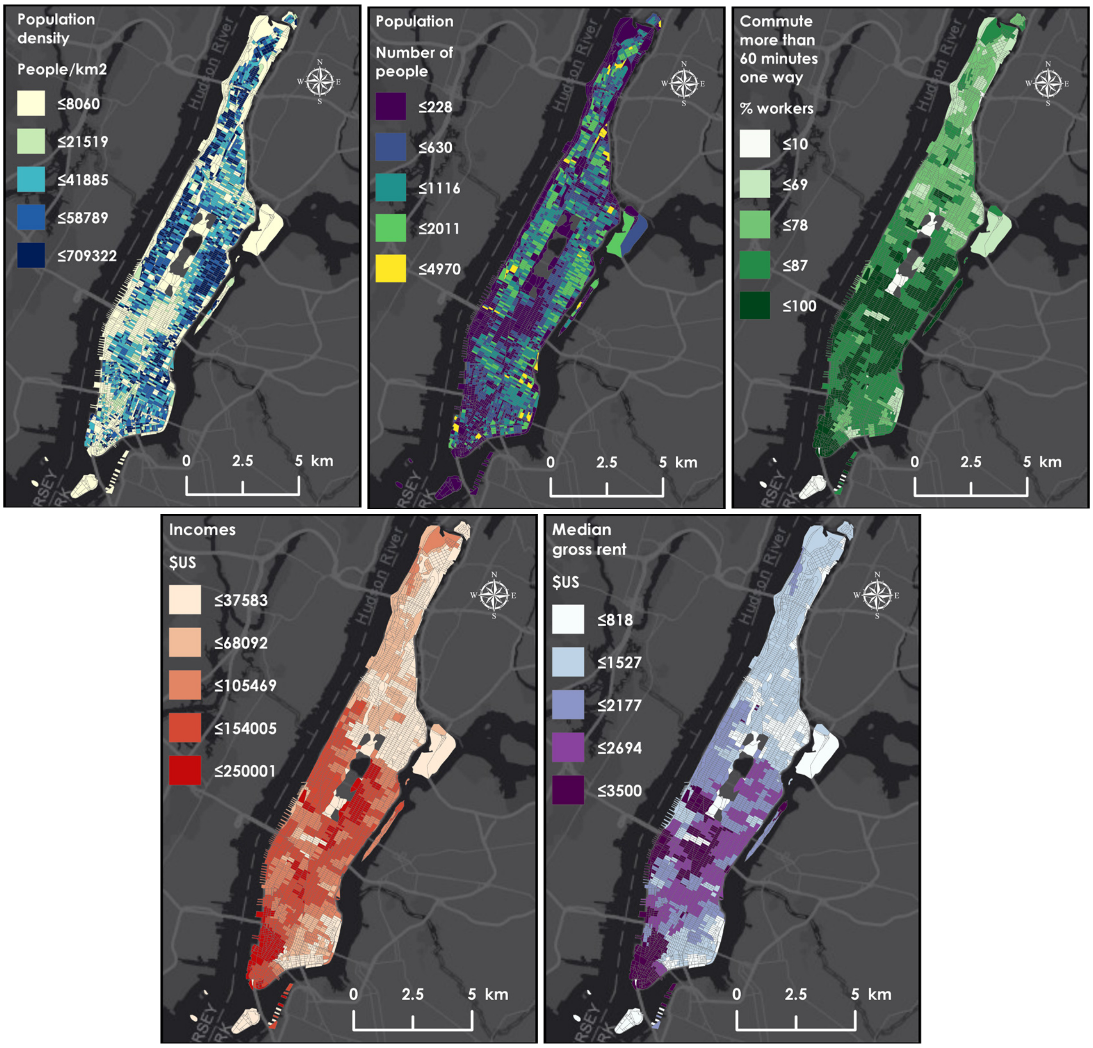


Figure 4. Socio-economic variables geo-visualization

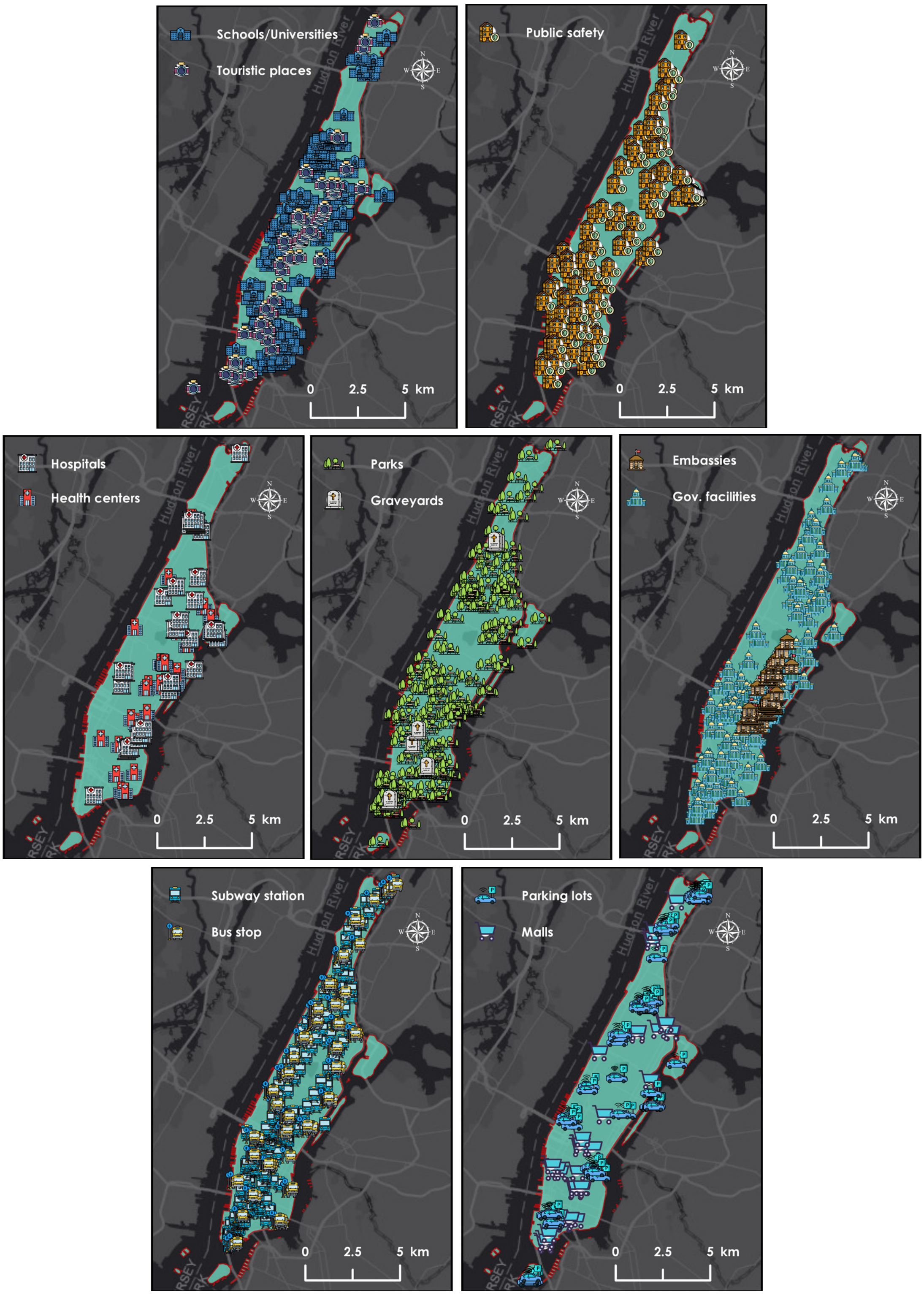


Figure 5. POI variables geo-visualization

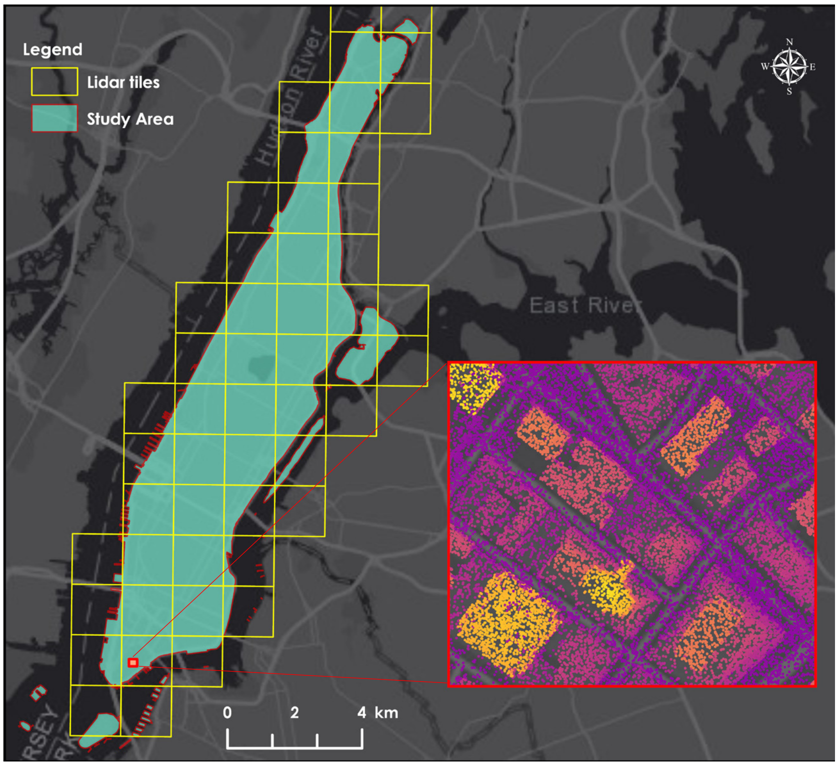


Figure 6. LIDAR tiles used in the rooftop flatness assessment

3.1.3 Data preprocessing

Preprocessing for the place suitability analysis consisted of joining all the information coming from socio-economic and environmental variables to the census blocks since this is the minimum unit of information for this analysis. Therefore, all alphanumeric information from tables was joined to census block identifiers. Besides, air-traffic and road-traffic noise information was also associated with the census blocks by averaging noise values through geoprocessing tools. Polygon-type POIs were transformed to point geometry upon finding the centroid; however, polygons with a significant extent such as parks were transformed when densifying the polygon with random-placed points through GIS tools. Before the execution of the SOM, GeoSOM and *k-means* algorithms (Chapter 3.2.2), data transformation was implemented due to the variability of measurement units for each of the input data involved in the analysis. Among the main transformation methods, Min-Max and Z-score normalizations were tested for the evaluation of spatial patterns from *GeoSOM* suite software. Furthermore, verification of null values and the potential presence of outliers was assessed. However, due to the nature of some variables, especially those generated after the proximity feature extraction process (Chapter 3.2.1), null values and possible outliers were preserved throughout the analysis when considering their information was relevant for pattern identification. Regarding the suitability analysis for roofs, LIDAR data was downloaded through a Transfer Protocol Files (FTP), and organized in a folder for easy handling. LIDAR tiles were examined individually to verify the existence of the information, and in some cases, it was necessary to apply coordinate system transformation for proper geolocation. Outliers verification for elevation values was also performed before the calculation of elevation statistics for each of the rooftops (Chapter 3.3.1).

3.2 Place Suitability Analysis

3.2.1 Proximity Features Extraction

Driving Distance algorithm from *pgRouting* library (Chapter 3.5) is calculated for proximity feature extraction, which allows expansion and maximization of the information available in the Manhattan census blocks. Therefore, vector geographic layers with dimension 0 (point type) for each of the POI categories are stored in *PostgreSQL* database with *PostGIS* extension (Chapter 3.5). A routable Manhattan road network is downloaded from Open Street Maps (OSM) through *Geofabrick* repository. Then by using *osm2pgrouting* (Chapter 3.5), the road network can be stored in the PostgreSQL database. This network is characterized by having information about the source nodes, destination nodes, edges, and cost. Census block centroids are also stored in the PostgreSQL database. Once the information is organized, the first step is to calculate the closest node to each element for the different POI types and census blocks by using Euclidean distance (*Figure 7-a*). In this way, new tables of closets nodes are generated for each geographical layer.

Subsequently, *Driving Distance* algorithm is executed by assuming a maximum distance of 2km from each closest node of the census blocks. The result is a new table with all possible nodes and edges that are within 2km away for each census block (*Figure 7-b*). Since all the information is properly related and referenced in the database, SQL queries allow us to obtain the number of POI elements that are associated with the nodes within 2km. Moreover, it allows knowing what is the shortest distance to reach each of the POI type nodes (*Figure 7-c*). SQL routines lead associating these results as two new columns per POI types; which are the number of reachable POIs and minimum distance to reach each POI type. Thus, by using *Driving Distance* we are adding real proximity information to the census blocks through the POI geographic location.

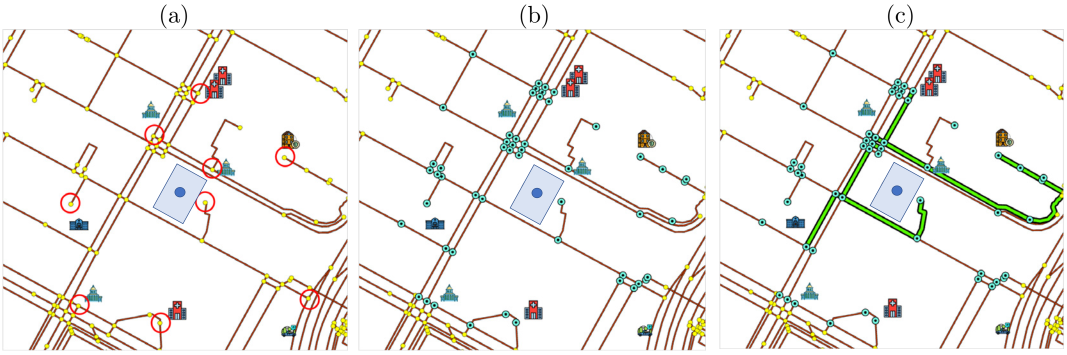


Figure 7. Example of the proximity features extraction process. (a) Selection of closest node to each POI and census block. (b) Selection of all possible nodes within 2km *Driving Distance*. (c) Selection of the shortest path to each POI type from the census block node

3.2.2 Clustering algorithms: SOM, GeoSOM, k-means

Prior to SOM Execution, it is necessary to apply a normalization of the variables due to they have different measurement units. Min-Max normalization technic was selected for this purpose. Therefore, the new attribute values for the census blocks (training patterns) will be between 0 and 1. For SOM training algorithm from *GeoSOM* software, it is necessary to configure the input parameters requested by the tool. It is important to note that the quality of the results is directly related to the parameters selected for SOM [22, 37, 62]. Although theoretically, there is no better configuration for the initialization of SOM parameters [22], special considerations from the literature review were studied for parameter selection. Subsequently, a sensitivity analysis after running the algorithm several times contributes to the detection of the minimum possible error of both quantization and topographical. Then this can helps in the best configuration selection. Regarding SOM structure parameters, most attempts were made using a hexagonal topology because it offers a connection to its six neighbors and generates smoother maps [22]. Different input space sizes were examined considering that the larger it is, patterns and grouping structures are clearer to define [22, 58].

On the other hand, within the training parameters, different values for the number of iterations, learning rate, and neighborhood radius were tested in each SOM attempt. These variations were established considering the two phases of SOM training, unfolding and fine-tuning. The learning rate and neighborhood radio values are usually smaller in the tuning phase than in the unfolding. In contrast, the number of iterations in the unfolding phase is usually less than in the tuning phase. These considerations are raised because in the unfolding phase all neurons are scattered throughout the input space, while in the tuning process, the neurons are located in areas where training patterns are highly concentrated [22]. Other training parameters such as learning function and neighborhood function were selected according to literary review. The *Inverse* function was selected for the first one by allowing greater network mobility in the training phase and achieving rapid adjustments [22], while Gauss function was selected for the second according to showing a smoother granularity in the resulting map [22]. Once the selected configuration shows satisfactory results, the output space is analyzed by means of the U-Matrix, which provides information about the distance between neurons [26, 62] (*Figure 8-a*). The grayscale palette color shows high distances in dark tones, and low in the light ones. Hints visualization provides information about the number of training patterns associated with each neuron (*Figure 8-b*).

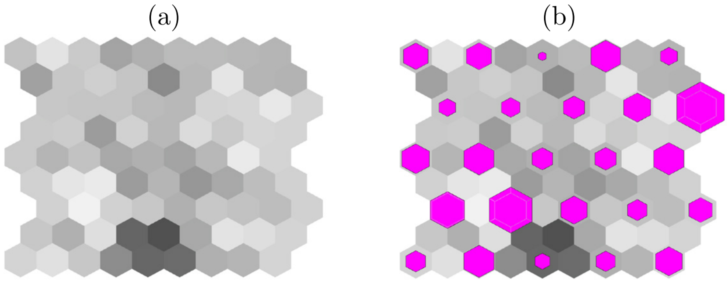


Figure 8. Example U-Matrix 5x5 neurons. (a) U-Matrix distances visualization (b) U-Matrix hints visualization

To simplify the visualization of the patterns in the output space, a new SOM is trained with the same size in X as the initial one but with only one neuron in Y. *Figure 9* shows that for the previous example, the resulting U-Matrix size is 5x5, so the new U-Matrix will have a size of 5x1. This new U-Matrix is taken as a base-line for the cluster projection per neuron in the U-Matrix 5x5, where pie-charts depicts the proportions of associated clusters in each neuron (*Figure 9*). Visualization of the patterns and structures through Box-Histo diagrams and component planes allow observing potential similarities and differences present in certain variables for each of the generated clusters. Because this thesis is aimed at taking the initial clusters only as a guide, new clusters are created from *GeoSOM* software by identifying areas and clusters of the U-Matrix that may tend to reflect similar patterns.

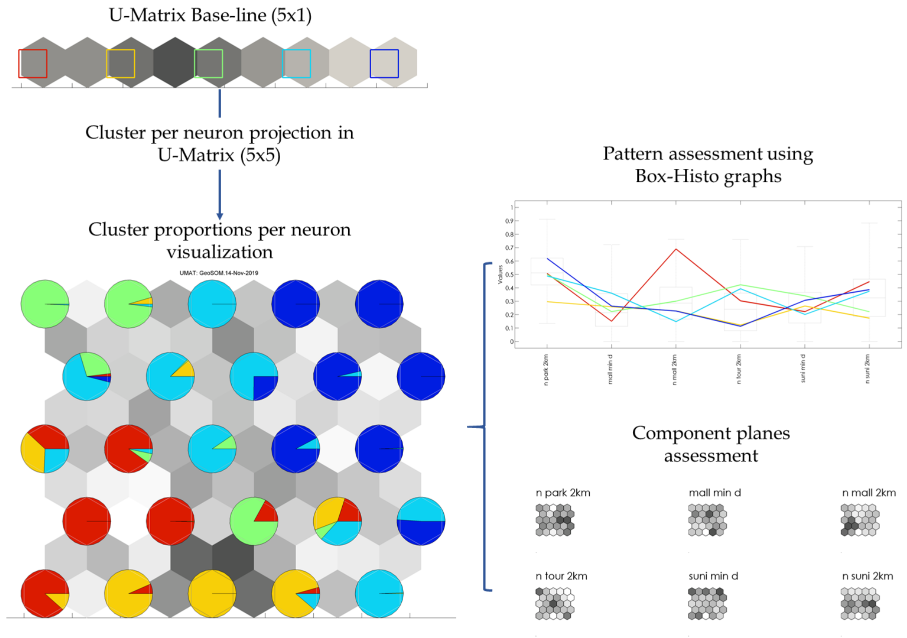


Figure 9. Strategy for the visualization of the initial clusters through cluster projection per neuron from a U-Matrix base-line

Regarding the GeoSOM algorithm, this is trained from *GeoSOM* suite with the best configuration of parameters found in SOM. That is because the only difference between SOM and GeoSOM is that the latter considers the relationship of spatial contiguity offered by the nature of the geographic data. The methodological process described is again executed for the results of GeoSOM, thus generating a base-line U-Matrix that will project cluster per neuron in the initially generated U-Matrix. Pattern evaluation is performed for GeoSOM through Box-Histo and component planes.

Although SOM is a powerful clustering technique when finding in its training process the best matching prototype in addition to updating topological neighbors on the map [5], the goal of running *k-means* is to have a point of comparison with an unsupervised clustering algorithm other than SOM. Therefore, *k-means* is executed by using the same initially normalized values, besides to take into account the same number of clusters selected for SOM. Subsequent visual inspection of the distribution of the spatialized clusters will allow the selection of one of these algorithms. The selected algorithm will be used to perform a cluster analysis of the non-standardized data. In this way, a description of the main characteristics of each cluster can be carried out when evaluating the collective behavior of the non-normalized variables. The defined clusters will be the main input to continue with the classification of place suitability levels.

3.2.3 Classification of Place Suitability Levels

While this thesis is aimed at developing a non-expert knowledge and unsupervised data-driven based methodology, suitability levels will be calculated from the results obtained by the clustering algorithm selected. Furthermore, the analysis ensures suitability levels are established within census blocks that have similar patterns. Consequently, the literature review allows an interpretation of the importance of the environmental, socio-economic and proximity variables involved in the study. High air/road traffic noise and PM values can influence the design of navigation routes for air-taxis [4, 9, 27, 48, 56], as well as defining pilot areas to improve air quality [27, 48]. Socio-economic variables such as medium household incomes and gross rent with high values could indicate the location of potential inhabitants who can afford air-taxi services for business trips [15, 27, 48]. In addition, demographic variables and workers' data could contribute to finding areas where UAM vehicles would facilitate not only public transport but also cargo and delivery activities [9, 28, 44]. Proximity variables calculated by the *Driving Distance* algorithm provide detailed information about the number of accessible POIs for each of the census blocks. Therefore, high values for these variables would indicate possible regions near nodes of recreation, health, transportation, public safety, and citizen care. In contrast, low values for attributes that describe the minimum distance to reach POIs show greater importance in accessing the mentioned nodes.

When considering these statements, this thesis proposes then to establish per cluster a statistical Jenks distribution in each of the variables. Jenks or *"natural breaks"* distribution allows establishing classifications by optimizing the differences between them [13]. Therefore, *Jenks* distribution is performed to establish three different class ranges per variable in each cluster. By bearing in mind the importance of the described variables, a score is assigned to each of the classes in each variable. This score is built in a range from 1 to 3. Where 3 will be assigned to classes of greater importance, 2 medium importance, and 1 low importance. *Figure 10* allows us to observe the processing scheme about the generation of *Jenks* classifications given per variable in each cluster. Subsequently, the summation of the scores of the variables per cluster is performed in order to have totalized values of all scores in each cluster. Then these totalized scores are classified again by implementing *Jenks* distribution to keep the same class ranges previously established. Thus, the final suitability categories are established in each of the clusters.

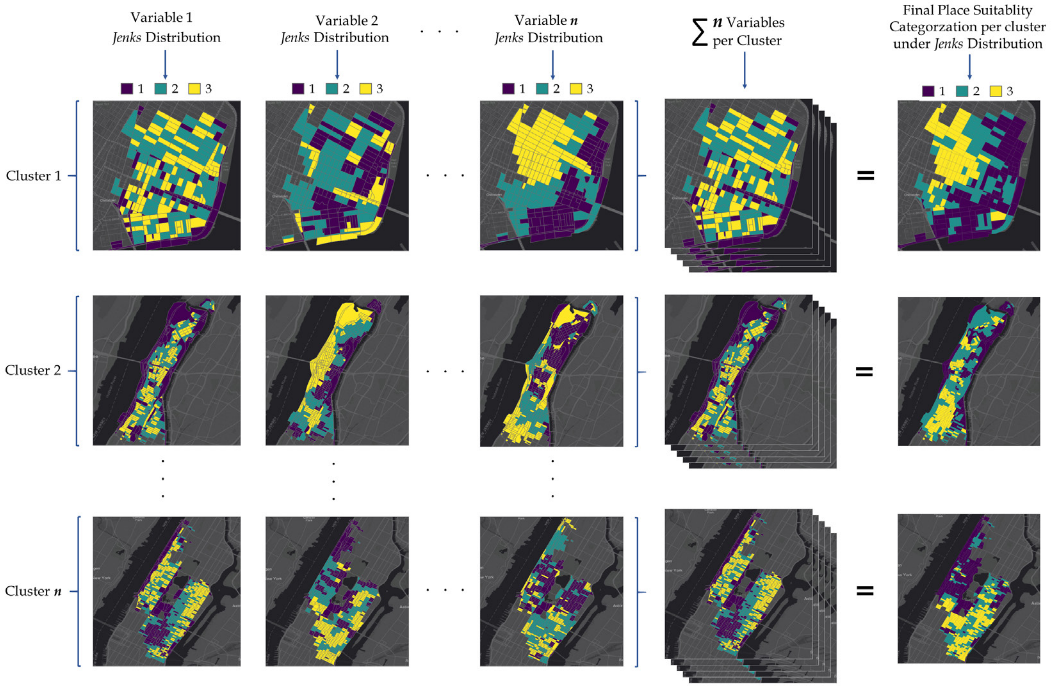


Figure 10. Scheme for the categorization of Place Suitability Levels for each of the n clusters calculated

3.3 Rooftop-Flatness Assessment

This section of the thesis is aimed at explaining the methodology applied to the evaluation of roof-flatness, which is developed in two phases. The first one seeks to estimate the basic roof elevation statistics. The second stage is responsible for carrying out the flatness evaluation to then determine suitability levels to build UAM platforms.

3.3.1 Rooftop Elevation Statistics Calculation

The first step in the calculation of roof elevation statistics is to crop the tiles of LIDAR 3D cloud points with the rooftops footprint vector layer. The algorithm designed in Python takes advantage of the existing functions and methods in the *ArcPy* library for the LIDAR management data. In this way, the algorithm performs a routine to extract the 3D cloud of points in each roof of Manhattan. Due to the possibility that a roof can geographically share information from more than one LIDAR tile, the algorithm considers these situations by extracting the 3D points by sections and then annexing them. Therefore, the completeness of the information per roof is guaranteed. Taking as reference the work done by Castagno *et al.* [11] in the roof LIDAR data pre-processing, elimination of potential outliers was considered in the algorithm. These outliers can generate noise in the analysis when including topographic surface points.

Once the new LIDAR generated files are clean, roof elevation data is stored as *pandas* data frames to accelerate data processing. Thus, the algorithm calculates main statistics per roof such as mean elevation, maximum elevation, and standard deviation of the heights. This last statistic is of great importance since it works as a proxy in the identification of almost completely flat roofs by having values very close to zero. Then a geo-visualization of the statistics described yields to examine the spatial distribution and their patterns.

3.3.2 Rooftop Flat Surfaces Extraction

Once the cropped LIDAR files are obtained, the first stage of the algorithm is responsible for generating a raster file of the elevations for each roof. *ArcPy* LAS methods allow the rasterization of the 3D points where each pixel will take the corresponding elevation value. Pixels are generated with a size of 30 cm to increase the level of image detail. This step is based on the methodology applied by Castagno *et al.* [11] in the generation of LIDAR images for roof shapes identification. Having these images for each of the roofs, the next step is to perform a segmentation process to identify surfaces with similar height. This process is carried out by simulating GDAL polygonization function, where neighboring pixels that share the same value are grouped as a polygon. Nevertheless, this process is coded under *ArcPy* functions, where surfaces with the same height value are generated and stored as compact polygons. These polygons allow us to estimate the area of continuous surfaces with equal height based on the grouping of pixels at high resolution for each roof (*Figure 11*).

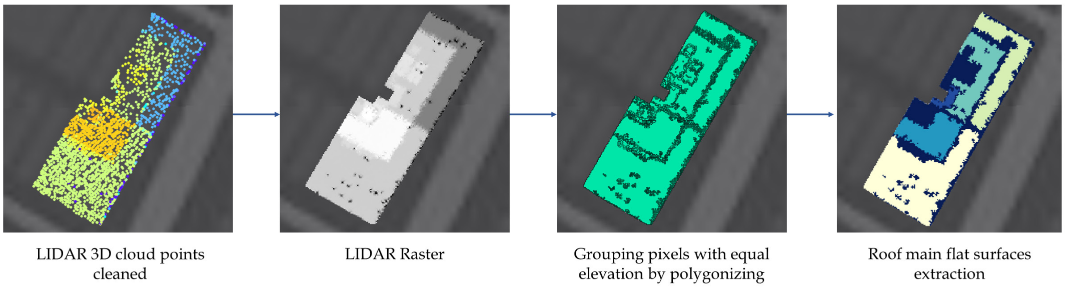

LIDAR 3D cloud points cleaned
LIDAR Raster
Grouping pixels with equal elevation by polygonizing
Roof main flat surfaces extraction

Figure 11. Main steps for rooftop flat surfaces extraction

Although other methods for precision segmentation and roof reconstruction have been proposed when working with LIDAR points directly [3, 12], LIDAR image segmentation simplifies the calculations necessary for this study by implementing GIS integrated tools. The algorithm orders these areas to select the five largest and then calculate the percentage they represent within the entire roof area (*Figure 12*). These areas are then classified according to the size necessary for the construction of different UAM platforms. Sizes for eVTOL aircraft proposed by Fadhil, Seeley, and Uber Elevate [15, 27, 53] were taken as a reference for the definition of size ranges for each of the possible UAM hubs type. However, a new category was proposed for the landing and clearance of UAVs in order to maximize the potential for rooftop use (*Table 3*).

Potential UAM Hubs Type	**Area Range (m^2)**
Vertihubs	> 3400
Vertiports	1000 - 3400
Vertistops	160 - 1000
UAV vehicles	3 - 160

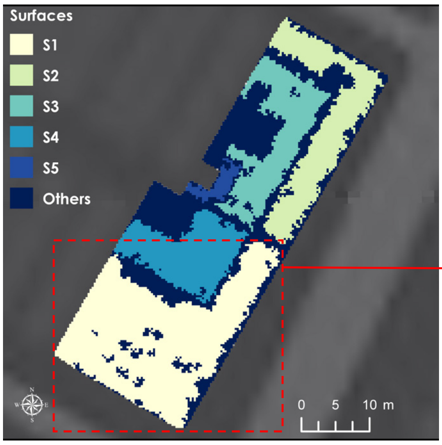
Table 3. Area ranges proposed for the categorization of UAM platforms

3.3.3 Rooftop Suitability Categorization

Because these polygons can have different forms, the designed algorithm evaluates the compactness of each of them by means of Elongation method or *Length-Width* proposed by Harris [21] and described by the formula (3). W_{MBB} refers to the width of the minimum bounding-box and L_{MBB} to the length of the minimum bounding-box. To calculate the ratio, the maximum measure of the polygon must be in the denominator.

$$LW = \frac{W_{MBB}}{L_{MBB}} \qquad (3)$$

Ratio values close to 1 indicate greater compactness than those close to 0. Then, the algorithm saves this value per polygon in each roof. As a second evaluation metric for the detection of polygons with rectangular shapes and inclined orientation, the algorithm also includes the percentage of area covered by the polygon within the bounding-box region. Metric elongation is then categorized into quartiles for the definition of suitability ranges.



	S1	S2	S3	S4	S5
Area (m^2)	382	166	141	133	17
% Total Rooftop	32	13.9	11.8	11.2	1.4
Elevation (m)	61	47	61	69	64
Elongation	0.7	0.5	0.4	0.7	0.5
% within Bbox	0.5	0.2	0.4	0.6	0.3

Figure 12. Example of attributes calculated by the algorithm in each of the main roof surfaces

3.4 Combined Categorization of the Suitability Indexes for UAM Platforms

The last stage in the proposed methodology is the generation of the final suitability for the location of UAM platforms. Therefore the process consists of combining the place suitability calculated from the definition of clusters and rooftop suitability given by the flatness assessment. Consequently, the spatial overlapping of both geographical layers (census blocks and roofs) will allow establishing amalgamate suitability categories. This combined suitability leads to both spatially and statistically to verify the identification of total suitability levels in each of the potential UAM platform types which can be developed in the different clusters. Thus, an estimate of the number of rooftops with high integral suitability for UAM platforms can be found, in addition to geographically knowing where they lie in each of the clusters. In the same way, the outputs let knowing which roofs definitely do not show integral suitability for the development of the UAM hubs. Furthermore, joined suitability levels can also be examined by relating them to the features given by the non-standardized urban ecosystem variables in each of the clusters. The overlapping process is made considering that no ambiguities are generated in the combination of the categories by knowing in advance that entities in the rooftop building footprint layer used in the analysis can only be inside a single census block.

3.5 Hardware and Software

All the data preprocessing and processes related to the place suitability analysis were developed and executed from a laptop with a processor Intel i7-9750H CPU 2.60GHz, 2592 MHz and 16 GB RAM. *Table 4* provides a brief description and purpose of the main software and libraries used.

Software/Library	Description/Purpose
PostgreSQL v.10.10	Open-source relational database management system for the information involved in the place suitability analysis.
PostGIS v.2.5.3 r17699	It supports the management of the geographical component of the stored layers in PostgreSQL.
pgRouting	Extension library for PostGIS / PostgreSQL geospatial databases to provide geospatial routing functionalities. Specifically used for Driving Distance calculations.
Osm2pgrouting v.2.1	Library used to parse OpenStreetMap (OSM) road network information from Geofabrick and allow it to be routable.
QGIS v.3.8.2	Desktop GIS used for visualization of geographic information stored in PostgreSQL.
GeoSOM Suite v.201	An open-source application developed by NOVA University to develop Self-Organizing-Maps by considering the spatial properties of geographic information. Therefore used for the development of SOM and GeoSOM clusters in the place suitability analysis.
Spyder IDE v.3.3.3 (Anaconda) – Python v3.7.3	Scientific Python Development Environment for editing the code used by creating the k-means algorithm.
Scikit-learn v.0.22.1	Python library to code the k-means clustering algorithm.
ArcGIS Pro v.2.2.0	Desktop GIS used in geoprocessing models design for the calculation of place suitability levels. Besides, it was used for mapping processing.
R v. 3.6.1 – Rstudio -ggplot2	R programing language for statistical computing was used under Rstudio for creating all the statistical graphics by using ggplot2 library.

Table 4. Main software and libraries used for the place suitability analysis

Regarding the rooftop suitability analysis, the development of the algorithm and analysis of results was carried out in the same laptop described above. However, its execution was performed in a distributed environment with five machines. Each of these machines had a processor Intel i5-4570 CPU 3201 MHz and 8GB RAM. *Table 5* briefly describes the software and libraries used.

Software/Library	Description/Purpose
PyCharm IDE	Python integrated development environment (IDE) for editing the code used to perform rooftop flatness assessment.
ArcPy	Python site package from ArcGIS used to apply LIDAR and geoprocessing functions within the algorithm.
Subprocess	Python library used to perform the parallel processing and processing-time optimization
Pandas	Python library used to manage and operate the attribute tables in an optimized way.
Os, Sys, Datetime	Python libraries used to manage directory files, folders and processing times.
ArcGIS Pro v.2.2.0	Desktop GIS used in geoprocessing models for the final rooftop-place suitability levels. Besides, it was used for mapping processing.
R v. 3.6.1 – Rstudio -ggplot2, lidR	R programing language for statistical computing was used under Rstuido for creating all the statistical graphics by using ggplot2. LidR was used for 3d Lidar visualizations.

Table 5. Main software and libraries used in the rooftop suitability analysis

4. Results and Discussion

The results presented below are part of the main outputs of the processes applied when following the methodology presented in the previous chapter. Thus, each section in this chapter depicts a description, analysis, and discussion about the importance of the results in the final identification of the adequate locations of UAM platforms.

4.1 Site Suitability Analysis

4.1.1 Driving Distance Calculation

Following the methodological process described in Chapter 3, after all the information related to census blocks, POIs, and road networks is pre-processed and stored in *PostgreSQL* database, the algorithm developed for the calculation of *Driving Distance* is executed. Therefore, the algorithm is responsible for calculating all the nodes and edges that are reachable at a maximum *Driving Distance* of 2 km for each of the 3.787 census blocks in Manhattan. As an example, *Figure 13* allows us to observe the coverage area given by the nodes and edges reached only for a specific Manhattan block.

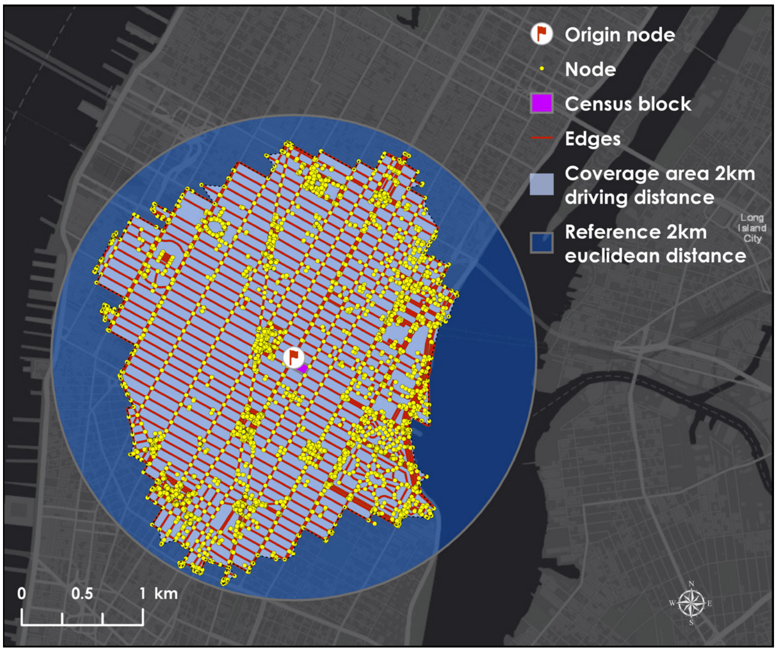

Origin node
Node
Census block
Edges
Coverage area 2km driving distance
Reference 2km euclidean distance
Long Island City
0
0.5
1 km
N
W
E
S

Figure 13. Coverage area obtained after executed *Driving Distance* algorithm at a distance of 2km for a specific block

This information is essential for the algorithm to determine both the number of POI reachable in that coverage area and the minimum distance needed to reach each POI type. Although this calculation can be complex when considering a large number of nodes (32.761) and blocks present in Manhattan, *PostgreSQL* and *pgAdmin* extension showed good performance when fully executed in less than 2 hours. *Figure 14* shows the different POIs which can be reached at a distance of 2km.

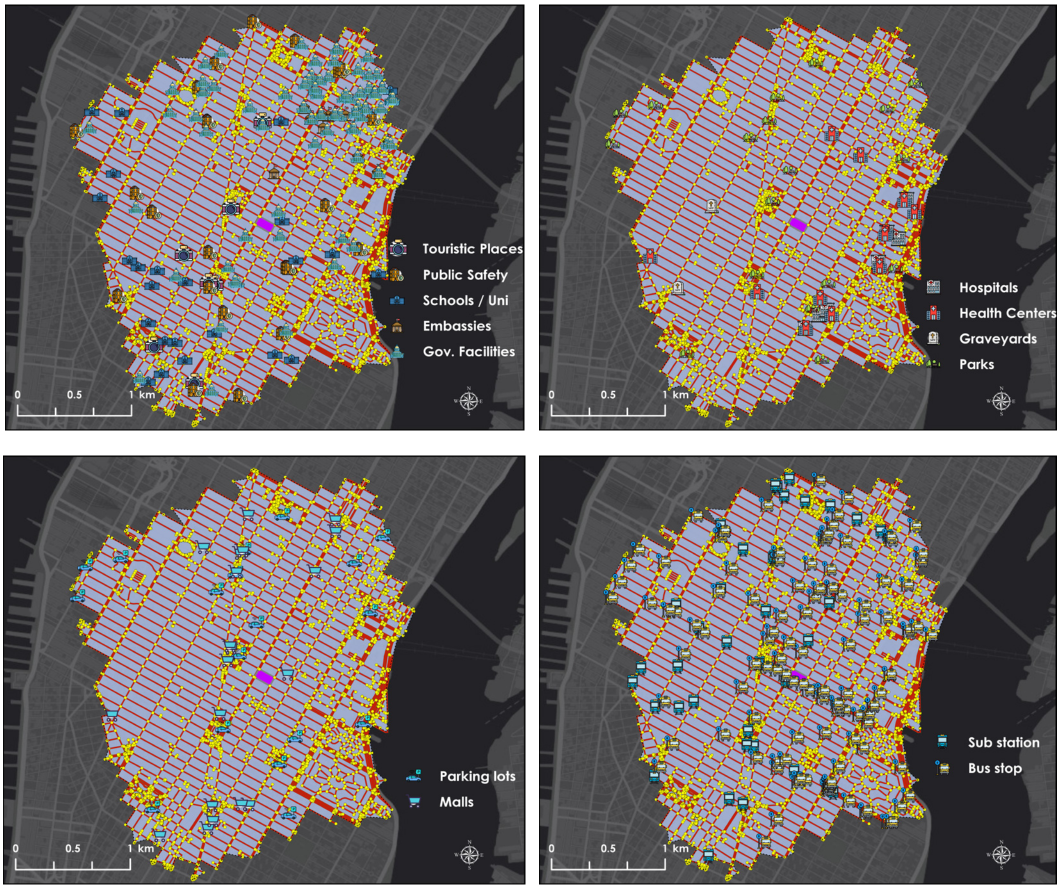


Figure 14. Visualization of the reachable POIs obtained after using *Driving Distance* algorithm

Consequently, the algorithm generates two new attributes in the census block table for each type of existing POI. Due to 13 POI categories are being considered in the analysis, 26 new attributes are added to the census block layer. *Table 6* depicts the values of the new proximity attributes obtained for each of the POIs, considering the previous example with a single census block. Undoubtedly, these new attributes, together with the environmental and socio-economic variables, contribute to strengthening the inputs of the subsequent analyzes.

	Proximity Attributes	
POI Types	**Number of Elements**	**Minimum Distance (m)**
Hospitals	7	934
Health Centers	11	811
Graveyards	2	871
Parks	19	233
Touristic Places	7	336
Public Safety	18	624
Schools / Uni	22	228
Embassies	11	402
Gov. Facilities	65	257
Parking Lots	18	77
Malls	16	228
Subway Stations	29	94
Bus Stops	103	83

Table 6. Example of new proximity attributes calculated by using *Driving Distance* algorithm for a specific census block

4.1.2 SOM Clusters

Once the geographical layer of Manhattan census blocks is appended with the socio-economic, environmental information and proximity attributes; clustering methods can be executed. Therefore, SOM is the first method that was executed from *GeoSOM* software. SOM was executed several times with different input parameter settings to perform a sensitivity analysis (*Table A- 1 – Annex 1*). This analysis allows us to select the best parameter configuration when evaluating both the quantization error and the topographical error present after execution. *Table 7* allows us to observe the final configuration of the selected parameters for SOM. With the aim to get a base-line U-Matrix for cluster definition, SOM is executed again with the same configuration but with a 10 x 1 size matrix, which means a variation only in the value of Y for the initialization of the map. Considering that the resulting U-Matrix will have ten neurons, from *GeoSOM* suite a cluster per neuron is established by default. Then these clusters are reflected in the initial U-Matrix (10 x15). *Figure 15* shows the proportions of training patterns (census blocks) belonging to each cluster given by the base-line.

Nevertheless, due to this is only taken as a reference for subsequent adjustment, Box-Histo (*Figure 16*) and component planes (*Figure A- 1, Figure A- 2 - Annex 1*) also contribute to the analysis of base-line clusters' behavior. The combined analysis allows us to verify potential similarities between clusters both by normalized values (Box-Histo) and by distances (U-Matrix and component planes). Consequently, new boundaries are drawn from *GeoSOM* suite, resulting in 8 new clusters (*Figure 17*).

SOM Parameters	**Value**
Lattice	Hexagonal
Size X-Axis	15
Size Y-Axis	10
Normalization type	Range (Min-Max)
Map Train	Sequential Train
Iterations (Rough)	10000
Radio (Rough)	5
Alpha (Rough)	0.3
Iterations (Fine-tune)	20000
Radio (Fine-tune)	3
Alpha (Fine-tune)	0.1
Q. Error	0.6000947
T. Error	0.0158437

Table 7. Best input parameters configuration selected for SOM

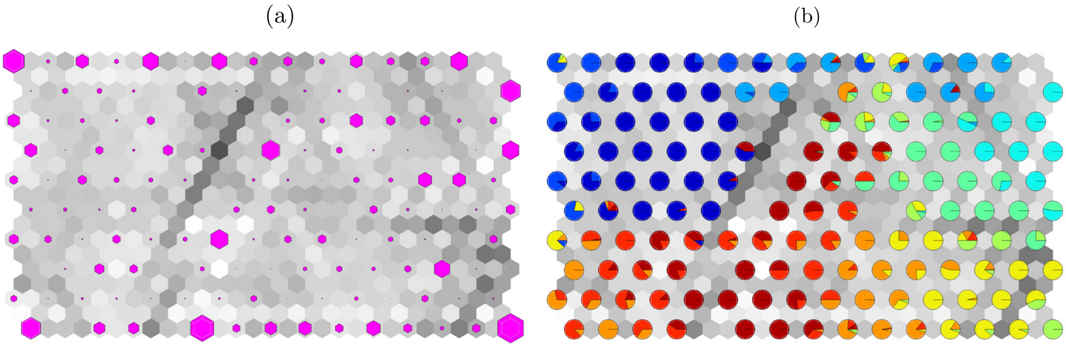


Figure 15. (a) SOM U-Matrix 10 x 15 obtained with best parameter configuration, (b) Reflected training pattern proportions from the 10 base-line clusters

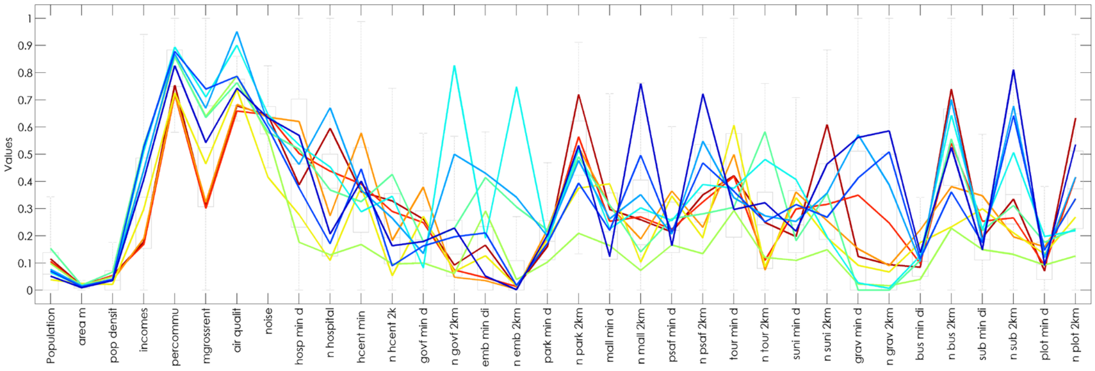


Figure 16. SOM Box-Histo for the 10 base-line clusters

A new Box-Histo is generated to verify the average behavior of the normalized variables in the 8 clusters raised (*Figure 18*). This new Box-Histo reflects certain similarities with the baseline. However, it associates more patterns that tend to be the same in fewer clusters. In addition, the clusters obtained reflect a set of different behaviors that characterize the uniqueness of the patterns present in them. To complement the description of the generated clusters, they were visualized on a map to observe their spatial distribution (*Figure 19*). Therefore, the overview of the Map with the SOM clusters allows us to observe the existence of clusters that tend to represent compact and non-dispersed groups (Clusters 2, 4, 5, 6, and 8). On the contrary, Clusters 1, 3, and 7 show greater dispersion throughout the study area, especially Cluster 1. Nevertheless, it is important to note that although Cluster 8 is mostly represented by a compact grouping of blocks, near the Downtown (South of the study area) shows a small grouping of census blocks with the same characteristics.

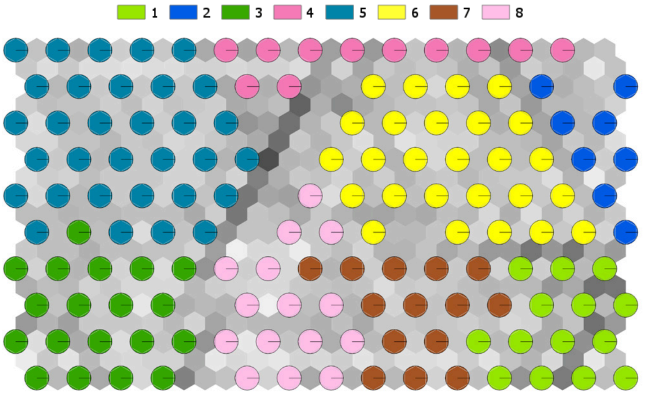


Figure 17. U-Matrix generated for the new 8 SOM clusters

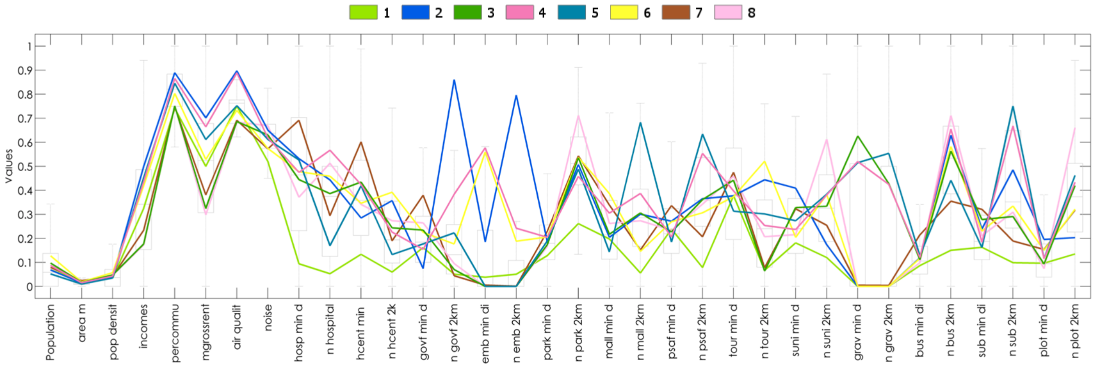


Figure 18. Box-Histo of the normalized variables for each of the 8 SOM clusters

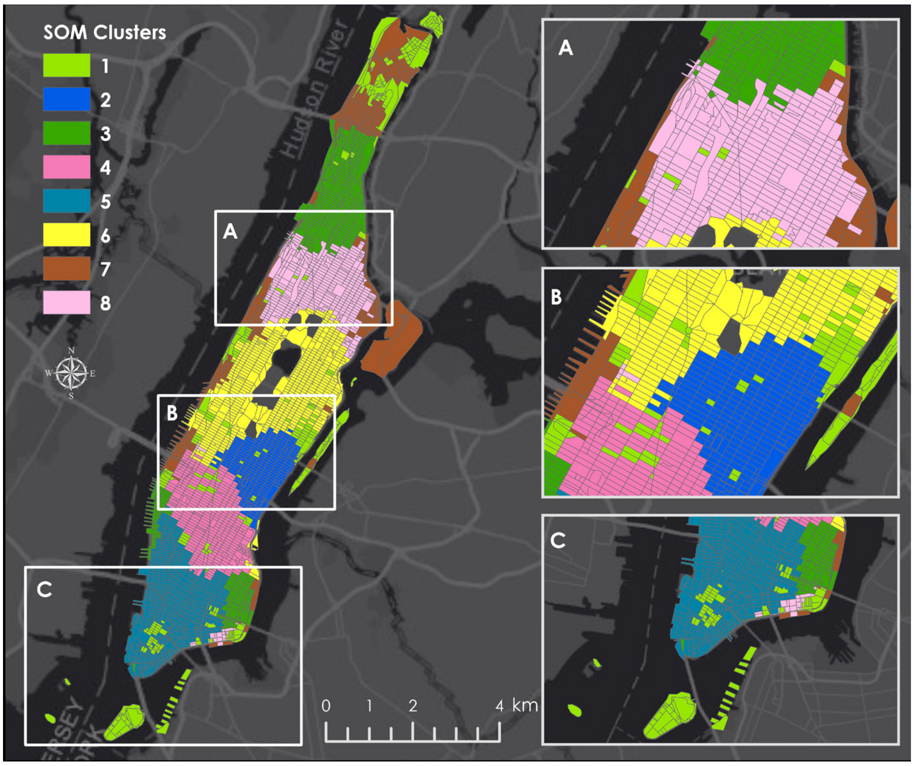


Figure 19. Manhattan census blocks clustered by SOM algorithm

4.1.3 GeoSOM Clusters

After running SOM, GeoSOM is executed using the best input parameters configuration. In the same way, a new GeoSOM base-line with a U-Matrix size 10x1 is generated in order to project a cluster per neuron to then obtain 10 base-line clusters (*Figure 20*). Proportions of the training patterns in each cluster are analyzed together with the Box-Histo (*Figure 21*) and the component planes (*Figure A- 3, Figure A- 4 - Annex 1*). The similarity in the average behavior of normalized variables and distances between neurons indicates that some baseline clusters may be showing very similar patterns. Thus, from *GeoSOM* a new design of the clusters is carried out and the steps performed in SOM are repeated. Eight new clusters are generated on the U-Matrix (*Figure 22*) where the average variability of each of the clusters is displayed in Box-Histo (*Figure 23*). Similar to SOM, some of the clusters can converge to similarities in some variables; however, in others, they can show totally different.

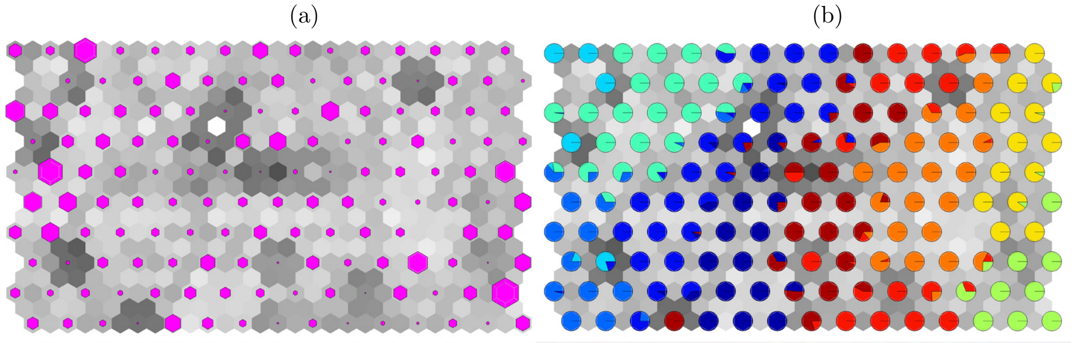


Figure 20. (a) GeoSOM U-Matrix 10 x 15 obtained with best parameter configuration, (b) Reflected training pattern proportions from the 10 base-line clusters

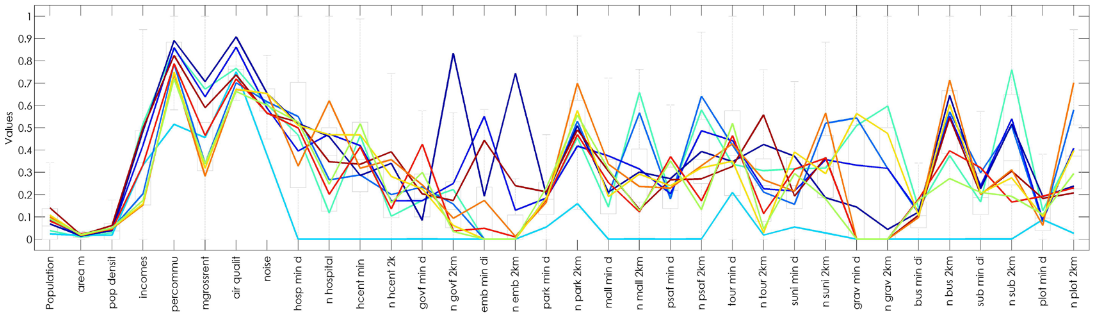


Figure 21. GeoSOM Box-Histo for the 10 base-line clusters

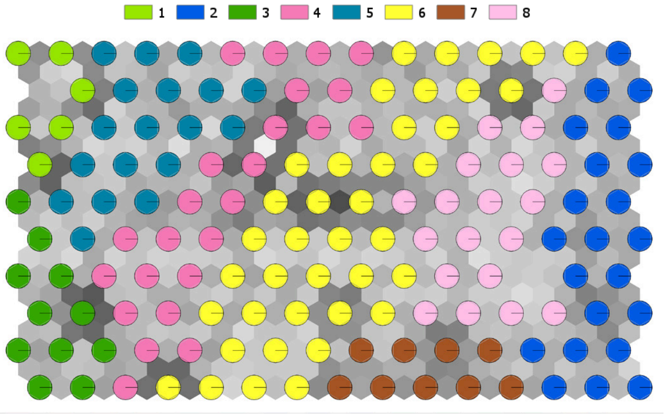


Figure 22. U-Matrix generated for the new 8 GeoSOM clusters

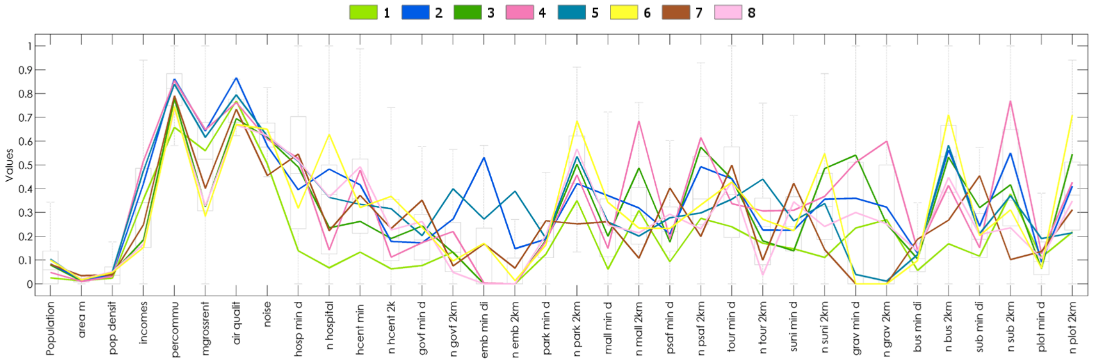


Figure 23. Box-Histo of the normalized variables for each of the 8 SOM clusters

As a next step, a visual analysis of the behavior of clusters generated with GeoSOM is carried out. Clearly and as expected by the nature of GeoSOM, the spatial adjacency as a topological relationship plays an important role in defining clusters. Most clusters show a clear tendency to form a single compact group, with certain exceptions. Just a few census blocks from Clusters 1, 2, 3, 4 and 5 are lightly far from their corresponding main group (*Figure 24*).

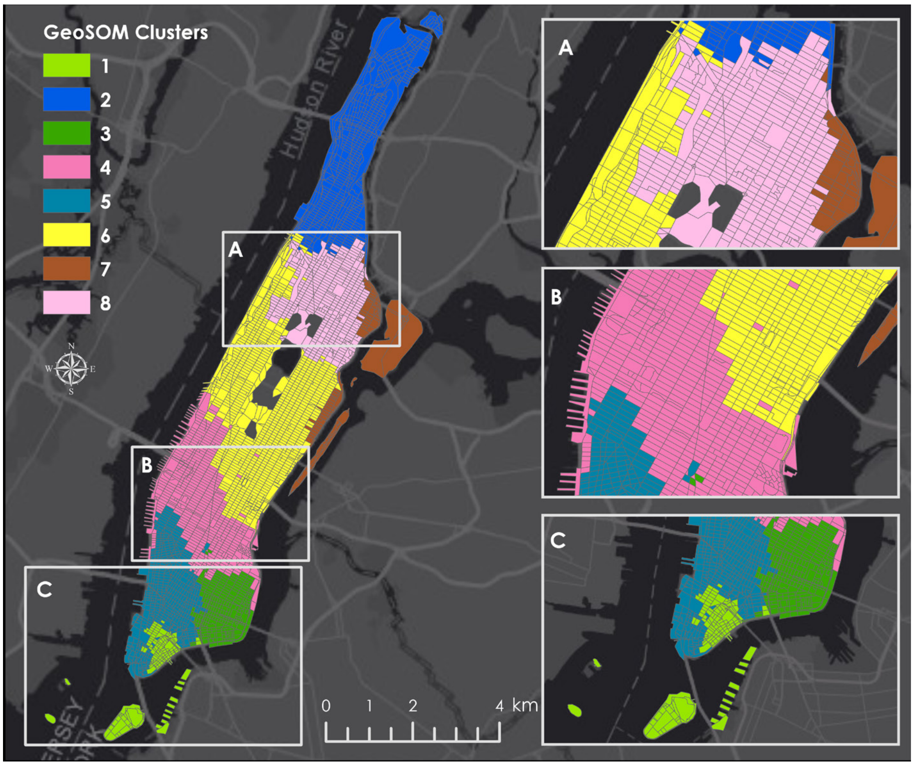


Figure 24. Manhattan census blocks clustered by GeoSOM algorithm

4.1.4 Visual Comparison of Cluster Results

Once the results have been obtained from SOM and GeoSOM, this thesis seeks to evaluate the quality of the information acquired in the variables by applying a different grouping algorithm. Therefore *k-means*, as mentioned in the methodology, was the algorithm selected to generate clusters based solely on the multi-dimensional distances given by the selected variables. *k-means* was executed to obtain the same number of clusters as in the other algorithms. It is interesting to note that although there are some dispersions of census blocks for some of the groups formed, most of them are concentrated in large blocks. Before addressing some of the main differences found between the algorithms, it is important to mention that colors and numbers were nominally set for ease of interpretation (*Figure 25*). Therefore, it is not possible to assert that SOM Cluster 1 is equal to *k-means* Cluster 1 even when they can be in a similar geographical position. With this in mind, we can observe that the present cluster patterns generated using SOM and *k-means* show a very similar spatial distribution. Greater differences are presented in concentrations given by Clusters 2 and 7. Concerning GeoSOM we can confirm that it is the grouping method that generates more compact clusters when considering contiguity in training patterns. Nevertheless, several of the clusters generated with GeoSOM show a spatial distribution slightly similar to those obtained in SOM and *k-means*, especially in the Downtown of Manhattan. Thus, the fact that the results of SOM and *k-means* reflect similar patterns in their clusters is evidence that the variables included in the analysis are informative enough to the context of the problem.

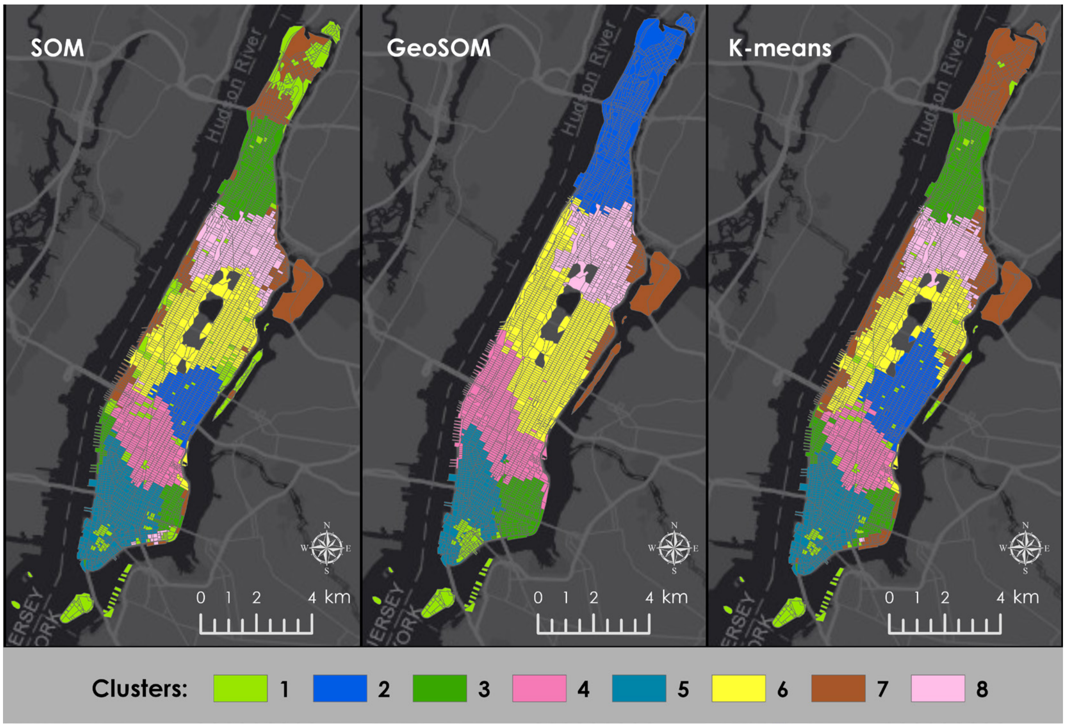


Figure 25. Map comparison from clusters generated using SOM, GeoSOM, and *k-means*

To continue with the methodology proposed in this investigation, it is necessary to select one of the proposed methods. GeoSOM, being the only one of the methods which adds the geographic contiguity of training patterns as a grouping factor, tries to guarantees that spatially the grouped census blocks keep greater compactness. Furthermore, GeoSOM results are selected since they depict more in sync with the definition of the first law of geography proposed by Tobler, *"everything is related to everything else, but near things are more related than distant things"* [57]. However, it is important to include in the analysis that although GeoSOM results show more compact homogeneity of training patterns when considering spatial variables, heterogeneity of non-geospatial variables can occur [24]. Subsequently, a deeper analysis of the clusters when reversing the normalization of the variables will allow generating more exhaustive statements.

4.1.5 GeoSOM Clusters Description

This section aims to analyze and break down the main patterns associated with each of the GeoSOM clusters when considering the non-normalized values of each variable. Since the analysis involves several variables, statistical graphs were grouped according to socio-economic and environmental variables (*Figure 26*, *Figure 27*, *Figure 28*, *Figure 29*), the number of associated POIs (*Figure 30*), and minimum distance to reach each POI (*Figure 31*).

- Cluster 1:

It is composed of blocks located in southern Manhattan, specifically in the financial district, in addition to the three main islands (Governors, Liberty and Ellis) along with some ports on the side of Brooklyn. It is characterized by being the cluster with the least number of POIs; however, it can have quick access to almost all types of POIs except for embassies. Despite its size, it concentrates high incomes and gross rent; this probably because it takes part in the city's financial district. On the other hand, it is the cluster with fewer workers. Deducing that it is probably a sector where there are quite a few entrepreneurs. Concerning air quality, it is one of the clusters with the highest concentrations of PM, although the noise levels are not as high compared to other clusters. In summary, this cluster may reflect the potential for the development of UAM platforms where eVTOL vehicles are related to commercial and business activities. Although it could also be used for the air-taxis services by covering the tourist sites that are in the mentioned islands.

- Cluster 2:

Located at the Upper Manhattan and bordering the Bronx, it is characterized by being the second most extensive. It is the second group with the lowest incomes. Compared to the other clusters, it has one of the highest noise levels, but one of the lowest in PM concentrations. Possibly related when being the cluster with the largest number of green areas (parks and graveyards). However, it has a considerable number of parking lots, bus stations and health centers. Taking this into consideration, the cluster does not reflect a high interaction of the variables to find out a population that will afford air-taxi services, except for very specific activities such as access to the few tourist sites or health centers.

- Cluster 3:

Located in the south of Manhattan and very close to the financial center, it stands out because it is one with the lowest population density, although it has a large percentage of workers. It is also characterized by showing very low levels of income and gross rent. It is a cluster where on average, the metro stations are quite distant. However, as transport nodes, it has several bus stations and parking lots. With these descriptions, it can be deduced that this area does not offer attractive features to massively build UAM platforms. These could be dedicated to the specific use of health centers and government entities present in this group.

- Cluster 4:

Located in Midtown Manhattan. It has access to most POIs. In addition, there is a significant percentage of workers; household incomes and gross rent rates are considerably high. On the other hand, it depicts the highest PM level among all the clusters, which is consistent with its high noise level. Without a doubt, it is a cluster that offers a priority scenario for the construction of UAM platforms to improve air quality. Additionally, it shows that it is a cluster with a mixture of services that can be offered, air-taxis, air-ambulances, air-security, among others.

- Cluster 5:

The grouping is located in Downtown Manhattan. One of the most remarkable characteristics is that although it has a very low population density, it shows the highest levels of both incomes and gross rent. Low presence of health centers and poor access to hospitals predominate. Also, it is a cluster that has a moderate number of tourist sites and a fair presence of transport nodes. Clearly, this cluster brings together the necessary properties to distinguish the population that can most afford UAM services.

- Cluster 6:

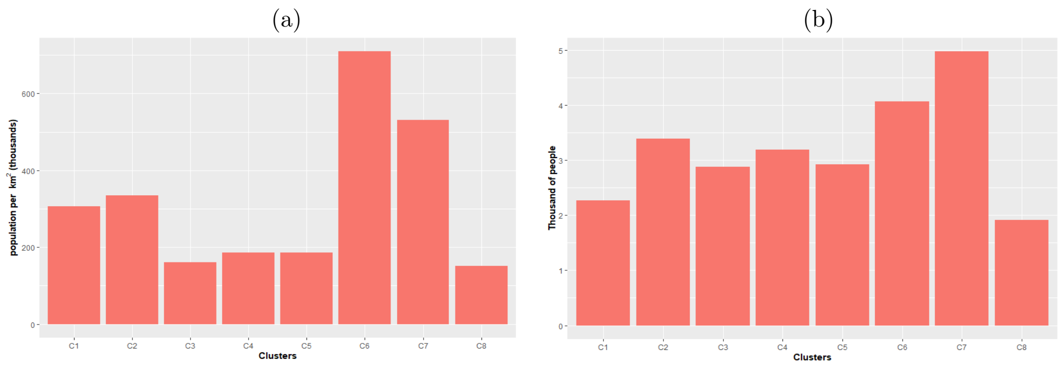
It has the largest extension and is located in Midtown Manhattan, covering the entire region near Central Park. It highlights due to brings together all embassies, in addition to having quick access to government and security entities. It also stands out for concentrates the majority of schools and universities. As expected, easy access to green areas. However, it has high levels of noise and PM. It is also a cluster where incomes and gross rent are moderately high. It aggregates more transport nodes, although not many parking lots. Considering these descriptions, it is a cluster that can provide platforms for several types of UAM services. Nevertheless, by collecting a large number of government entities, some areas could be used exclusively for diplomatic transport and security control.

- Cluster 7:

Located in the Upper east side of Manhattan, it groups blocks very close to the East River coast and involves the Randalls and Roosevelt Islands. It gathers the largest number of inhabitants; however, they have moderately low incomes and gross rents. It also shows to be the group with the lowest levels of noise. Besides, it has the largest number of hospital centers, but not very easy access to metro stations. This cluster may not be a priority for the development of services such as air-taxis, but the development of air-ambulances.

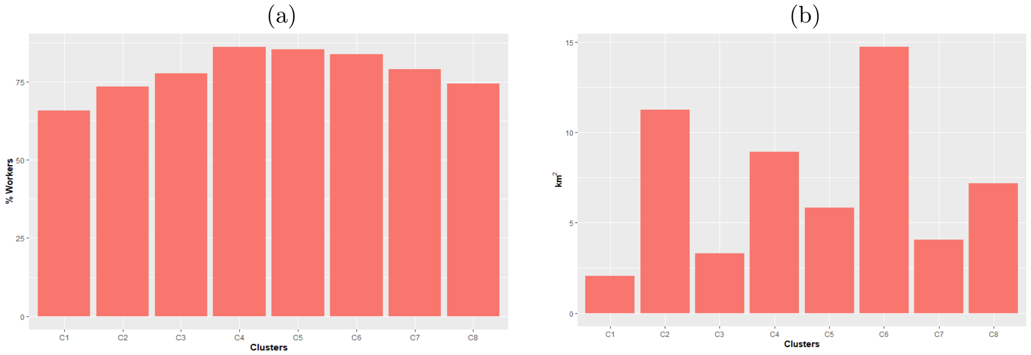
- Cluster 8:

Located at the north of the Central Park in Upper Manhattan. It has the highest noise levels and moderate PM levels. Although it is a group with low-incomes levels, it has a considerable percentage of workers. It groups the largest number of parking lots and has easy access to bus stations. It also shows having considerable number of health centers, schools, and universities. Therefore, although this does not show too many attractive features to build several UAM platforms, it shows favorable environmental conditions for the transit of eVTOL vehicles.


(a)

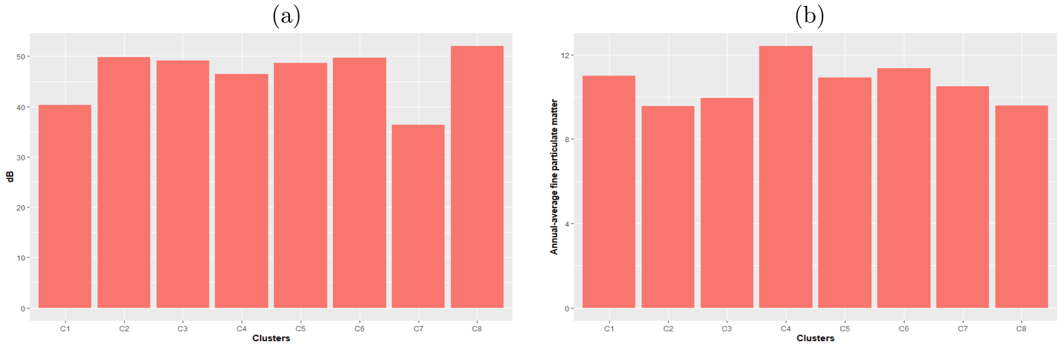
(b)

Figure 26. Bar-charts non-normalized variables (a) population density (b) population


(a)

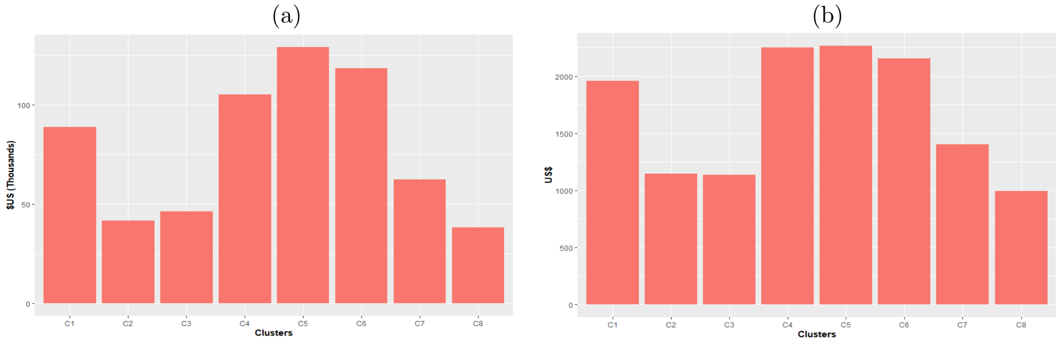
(b)

Figure 27. Bar-charts non-normalized variables (a) Percentage of workers that commute more than 60 minutes one-way, (b) Area km^2


(a)

(b)

Figure 28. Bar-charts non-normalized variables (a) Road-traffic and air-traffic noise, (b) Air-quality


(a)

(b)

Figure 29. Bar-charts non-normalized variables (a) Median household incomes, (b) Medium Gross Rent

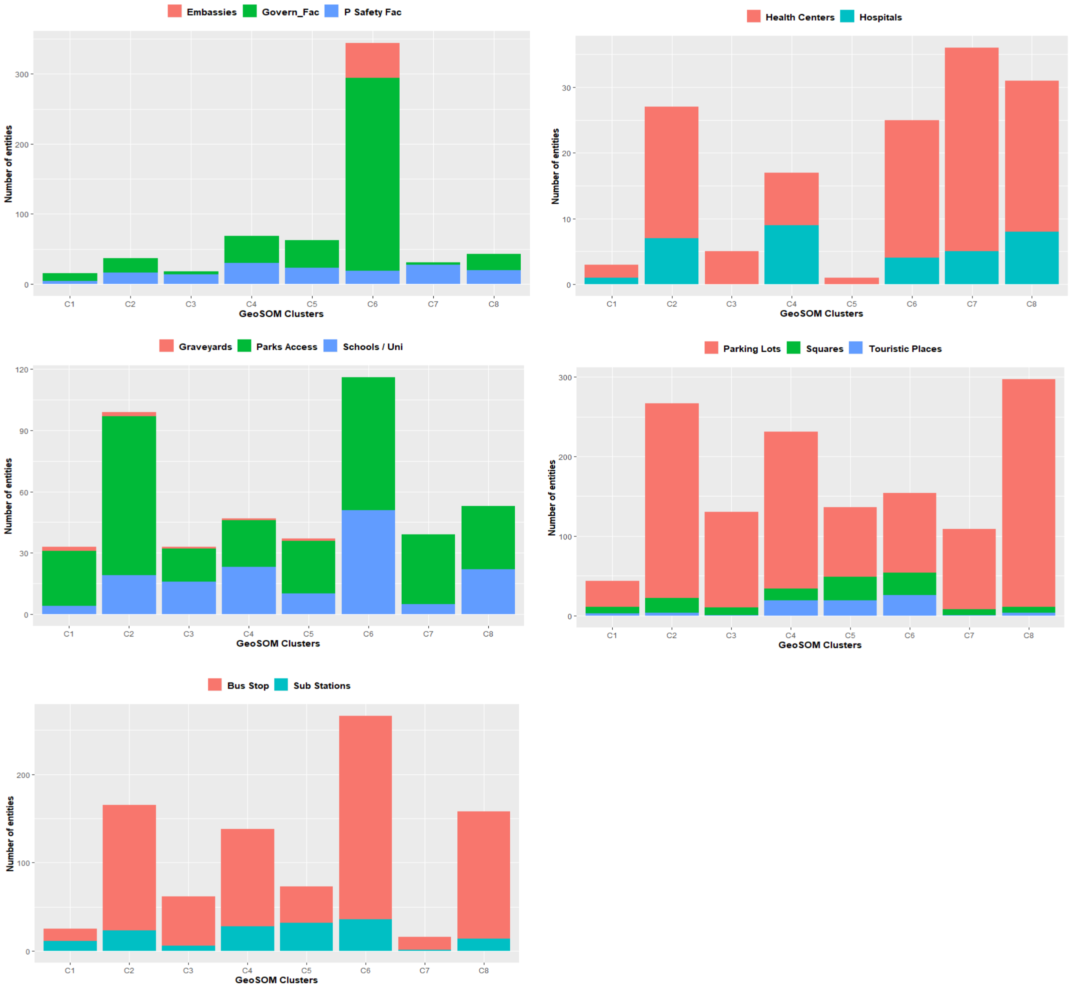


Figure 30. Comparative bar-charts for the number of POI present in each GeoSOM cluster

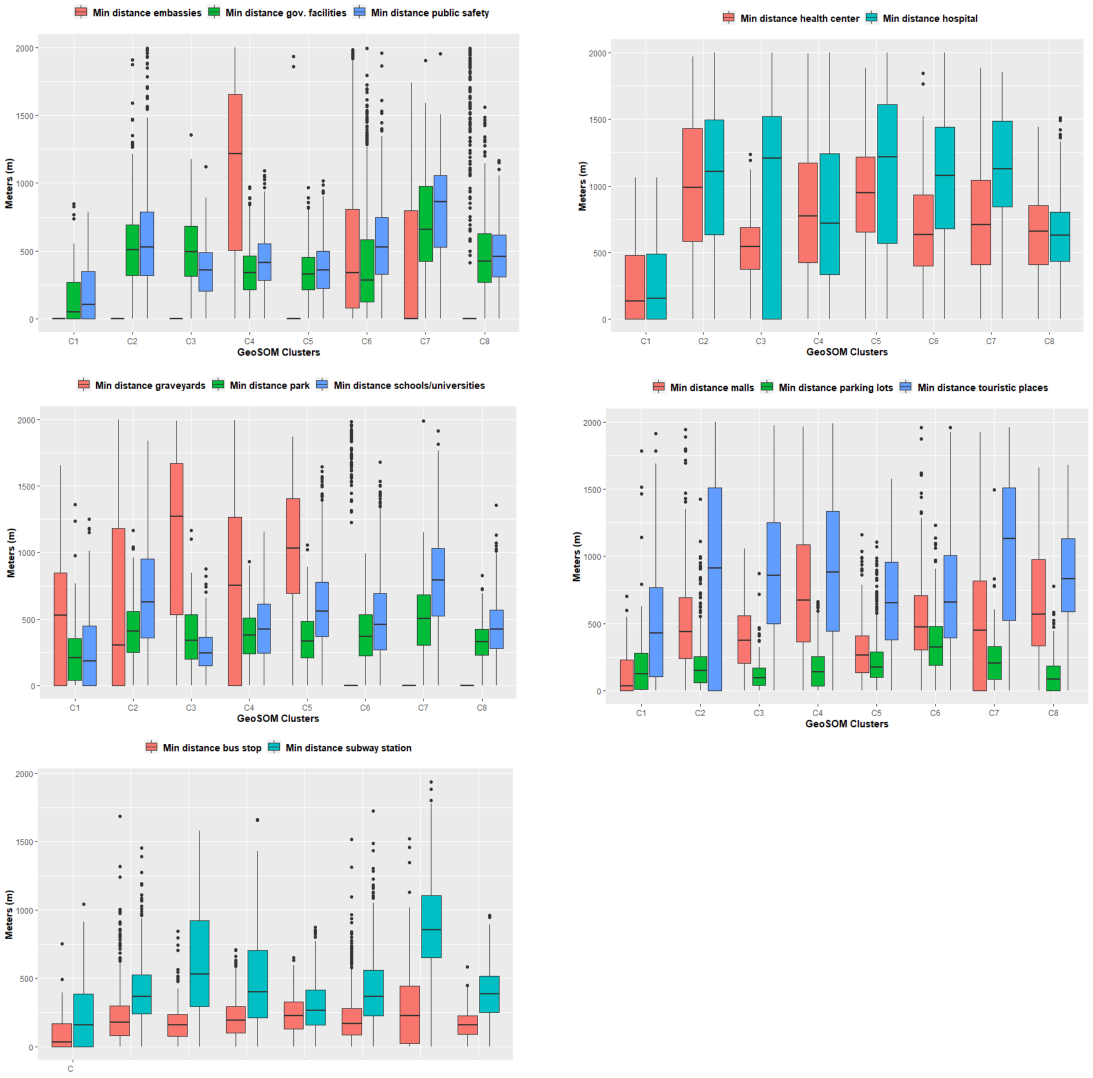


Figure 31. Comparative box-plots for the minimum distance to POI in each GeoSOM cluster

4.1.6 Place Suitability Levels Using GeoSOM Clusters.

Once the clusters are defined, the next step in the proposed methodology is the categorization of the suitability of the census blocks within each of the clusters. All the variables for each cluster were categorized into three groups under *Jenks* statistical distribution by implementing a geoprocessing model. The model allowed the scoring of each class in a range from 1 to 3. Classes established for socio-economic, environmental variables and number of POIs receive high scores as long as they have high values. In contrast, classes of variables related to minimum distance receive high scores as distances decrease. After each of the variables is ranked by cluster, the sum of these scores gives us the final grade. The final scores are classified into three classes performing *Jenks* distribution again. The resulting map allows appreciating the spatial distribution of suitability levels in each cluster (*Figure 32*).

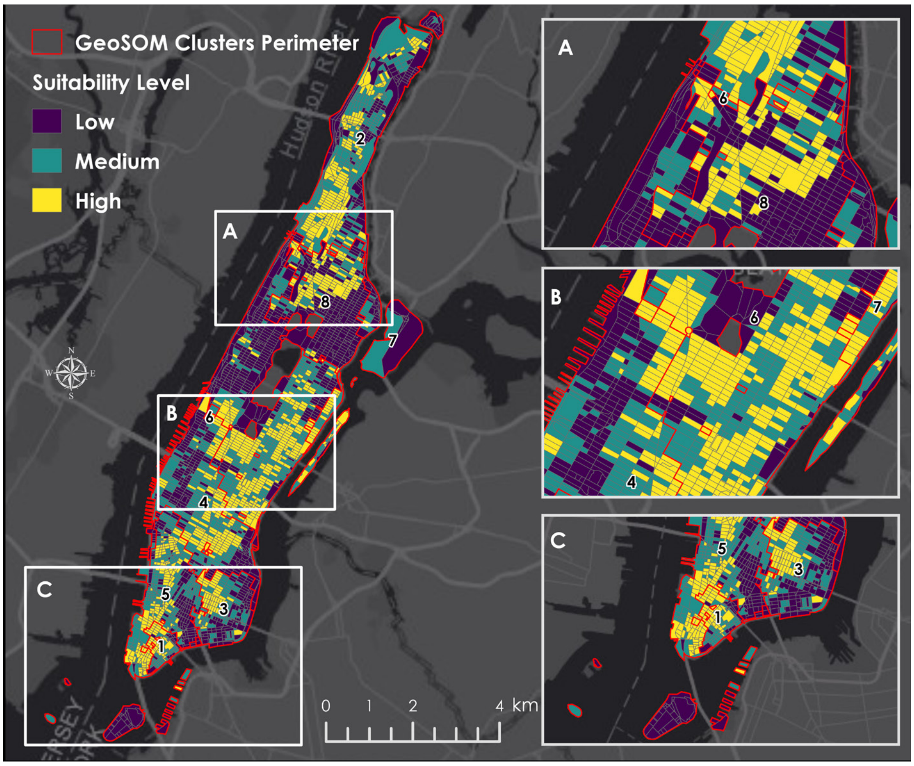


Figure 32. Place suitability map for Manhattan

Most of the clusters with high and medium place suitability are located in the south of the study area (Downtown and Midtown) and north of Central Park in Upper Manhattan. On the contrary, low suitabilities indexes predominate in Midtown Manhattan and the islands that are south of the study area. Remarkably, the result of the process guarantees the existence of census blocks for each of the established suitability categories (*Figure 33*). Cluster 6, although it collects a large number of census blocks with low suitability, is the cluster that has the most blocks with high and medium suitability levels (*Figure 34*). Proportionally Clusters 2 and 5 show higher block accumulations at high and medium levels than at low levels. Cluster 1 and 7 show very similar distributions, while Cluster 8 shows a significant number of blocks with low suitability. In an overview of the place suitability for Manhattan, 29% (1092 blocks) of the census blocks have a High index, 34% a Medium level (1290 blocks), and the remaining 37% (1405 blocks) with a Low suitability index (*Table 8*).

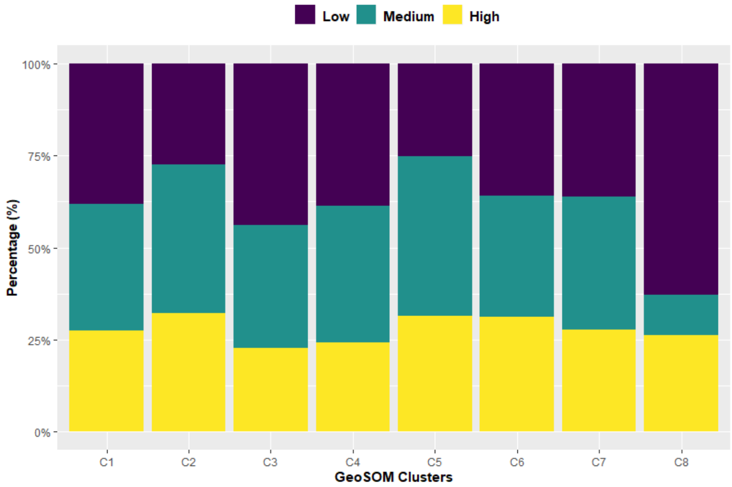


Figure 33. Census blocks proportions of place suitability levels per GeoSOM clusters

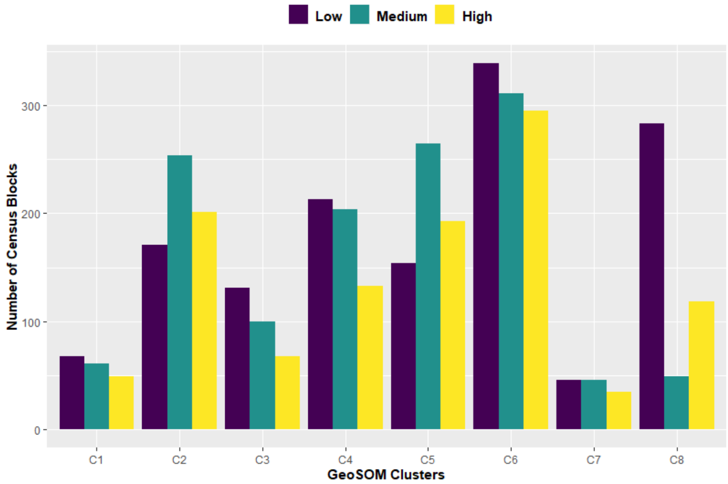


Figure 34. Distribution of census blocks in each place suitability index per GeoSOM clusters

	Number of census blocks per Place Suitability Index				
Clusters	**High**	**Medium**	**Low**	**Total blocks**	**% per Cluster**
C1	49	61	68	178	4.7
C2	201	254	171	626	16.5
C3	68	100	131	299	7.9
C4	133	204	213	550	14.5
C5	193	265	154	612	16.2
C6	295	311	339	945	25.0
C7	35	46	46	127	3.4
C8	118	49	283	450	11.9
Total blocks	1092	1290	1405	3787	100.0
% per Index	28.8	34.1	37.1	100.0	

Table 8. Summary table showing the distribution of census blocks in each place suitability index per GeoSOM clusters

4.2 Rooftop Flatness Assessment Results

This section explains the main results obtained after executed both parts of the designed algorithm. Finally, the results of the rooftops suitability levels are stated according to the type of platform in which they work.

4.2.1 Rooftop Statistics Results

After executing the first part of the algorithm described in the methodology for the flatness evaluation, the main output is the generation of the general statistics for the roof heights of each Manhattan building. *Figure 35* allows us to observe the spatial distribution of the average heights, maximum heights, and standard deviation of the heights per rooftop. By making a visual inspection, the highest heights are showed in the southern area of Manhattan (Downtown) along with some areas in the south of Central Park. In contrast, low heights are concentrated in the northern, northeast and lower Manhattan.

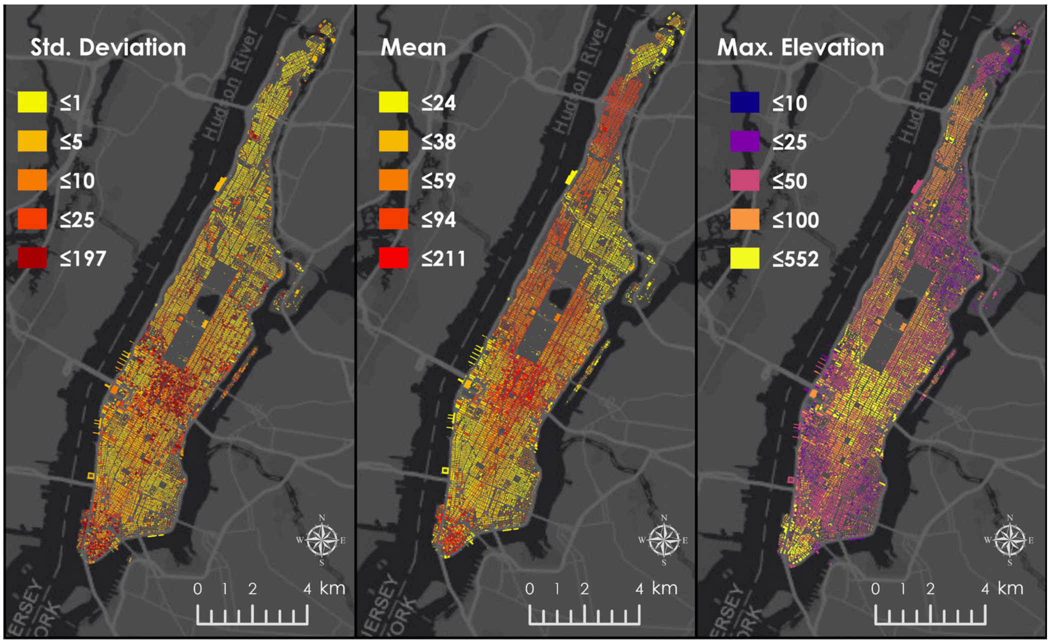


Figure 35. Maps of rooftops height statistics

A close relationship between maximum heights and high levels of standard deviation can be evidenced, which was to be expected. It is interesting the patterns depicted in the northern and Midtown Manhattan, as the rooftops have high average heights and low standard deviations. While in some areas of southern Manhattan, the standard deviation of the roofs is high but the average height is relatively low. As mentioned earlier in the methodology, the importance of standard deviations close to zero is that these could work as a good proxy in the location of possible flat rooftops. However, this does not fully guarantee that rooftops with medium and high standard deviations do not work for the development of UAM platforms. Therefore, an in-depth analysis is executed in the following section to find useful areas in all the roofs of the study area.

4.2.2 Rooftop Flatness Assessment Results

The second part of the designed algorithm allows having robust and complete information about the suitable surface size for landing and take-off of UAV and eVTOL aircraft. *Table 9* represents a sample of ten rooftops with their attributes and values calculated after the execution of the algorithm. For instance, Roof 1 indicates that its main flat surface has an area of 6753.7 m^2 at the height of 54 m, which occupies 40.4% of the entire rooftop area; its elongation (compactness) factor is very close to 1 (0.99) by occupying 72.5% of the total bounding-box area. Undoubtedly, it is a surface that offers great potential to work as a landing and clearing platform for vehicles within the UAM system. A visual inspection of a satellite image and its 3D representation helps us to verify this information (*Figure 36*). The main Roof 4 surface is also characterized by having a considerable area (3.393 m^2) and a moderately high elongation factor (0.7), which indicates that it can also be a good place to build UAM hubs (*Figure 37*). On the contrary, the main surface for Roof 7 indicates that its area is about 69.7 m^2 occupying 8.9% of the entire roof area with an elevation of 23 m high; also, the compactness indicator is relatively low (0.4) (*Figure 38*). In other words, this surface does not offer adequate features for the development of eVTOL platforms, but it could work as a UAV platform for landing and takeoff. The largest areas on Roofs 2 and 5 only occupy about 33% of the total roofs, having moderate elongation values (~0.6). These characteristics show that the roof might be divided on different surfaces, and some of them could work for some types of UAM platforms. *Figure 39* and *Figure 40* support verifying these characteristics.

		ID Rooftop									
	Flat Surface	**id_1**	**id_2**	**id_3**	**id_4**	**id_5**	**id_6**	**id_7**	**id_8**	**id_9**	**id_10**
Area (m^2)	**Surf. 1**	6753.7	382.1	1740.5	3392.9	462.2	155.0	69.7	676.3	202.8	34.9
	Surf. 2	142.4	166.5	345.6	824.2	160.3	139.7	63.7	23.3	113.0	16.4
	Surf. 3	124.0	141.0	108.1	36.8	93.1	5.9	60.7	15.7	55.3	6.2
	Surf. 4	119.7	133.8	40.5	36.7	34.3	2.5	59.7	1.8	7.1	3.4
	Surf. 5	24.7	17.7	40.2	12.2	33.7	2.0	32.8	1.3	5.0	1.8
% of the total roof area	**Surf. 1**	40.4	32.0	32.6	71.7	33.9	45.5	8.9	83.8	33.4	45.1
	Surf. 2	2.4	14.0	6.5	17.4	11.7	41.0	8.1	2.9	18.6	21.2
	Surf. 3	2.2	11.8	2.0	0.8	6.8	1.7	7.8	2.0	9.1	8.0
	Surf. 4	2.0	11.2	0.8	0.8	2.5	0.7	7.6	0.2	1.2	4.4
	Surf. 5	1.5	1.5	0.8	0.3	2.5	0.6	7.4	0.2	0.8	2.3
Elevation (m)	**Surf. 1**	54.0	61.0	11.0	29.0	95.0	27.0	23.0	49.0	36.0	16.0
	Surf. 2	23.0	47.0	11.0	32.0	96.0	26.0	22.0	54.0	37.0	19.0
	Surf. 3	51.0	61.0	12.0	32.0	95.0	30.0	23.0	52.0	36.0	20.0
	Surf. 4	51.0	69.0	12.0	36.0	105.0	27.0	23.0	50.0	38.0	20.0
	Surf. 5	50.0	64.0	12.0	33.0	95.0	29.0	20.0	52.0	37.0	16.0
Compactness Assessment (Elongation)	**Surf. 1**	0.99	0.7	0.5	0.7	0.6	0.8	0.4	0.6	0.2	0.3
	Surf. 2	0.2	0.5	0.3	0.4	0.7	1.0	1.0	0.7	0.4	0.6
	Surf. 3	0.3	0.5	0.1	0.6	0.3	0.7	0.3	0.6	0.7	0.5
	Surf. 4	0.3	0.8	0.9	0.9	0.5	0.5	0.3	0.8	0.5	0.3
	Surf. 5	0.3	0.6	0.3	0.1	0.5	0.5	0.3	0.7	0.7	0.6
% of bounding-box (ancillary)	**Surf. 1**	72.5	56.2	33.4	74.5	49.8	74.4	70.7	47.6	72.8	58.7
	Surf. 2	73.0	29.4	31.2	77.4	25.4	73.8	23.2	70.9	48.0	50.9
	Surf. 3	13.7	49.1	35.4	47.3	36.7	52.4	76.0	56.8	63.4	44.9
	Surf. 4	13.0	63.3	52.2	69.7	42.6	57.1	79.2	66.7	51.5	55.0
	Surf. 5	14.0	37.3	47.3	49.7	37.7	52.0	70.1	62.5	50.2	45.5

Table 9. Table of results for the second part of the algorithm for the flatness rooftop assessment

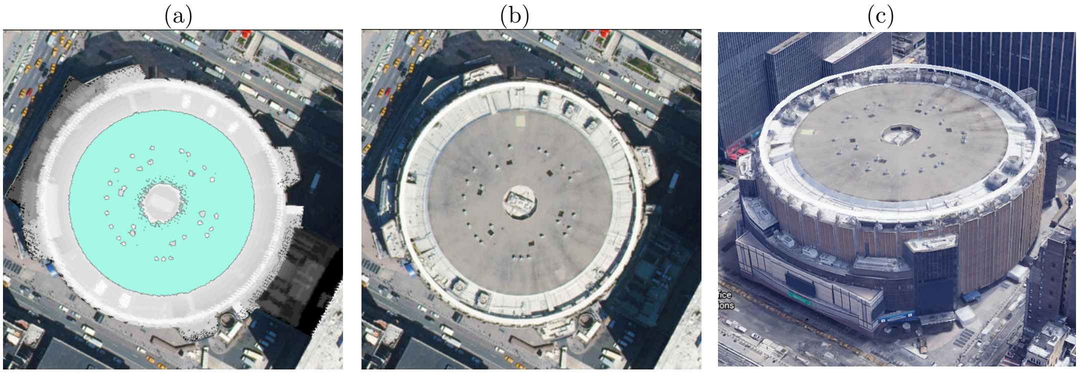


Figure 36. Roof 1 (a) Lidar image with main surface, (b) satellite image, (c) 3D scene

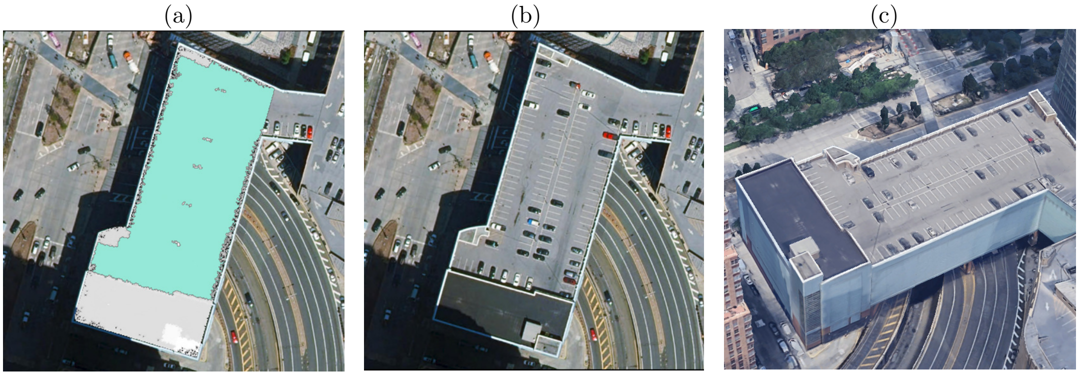


Figure 37. Roof 4 (a) Lidar image with main surface, (b) Satellite image, (c) 3D scene

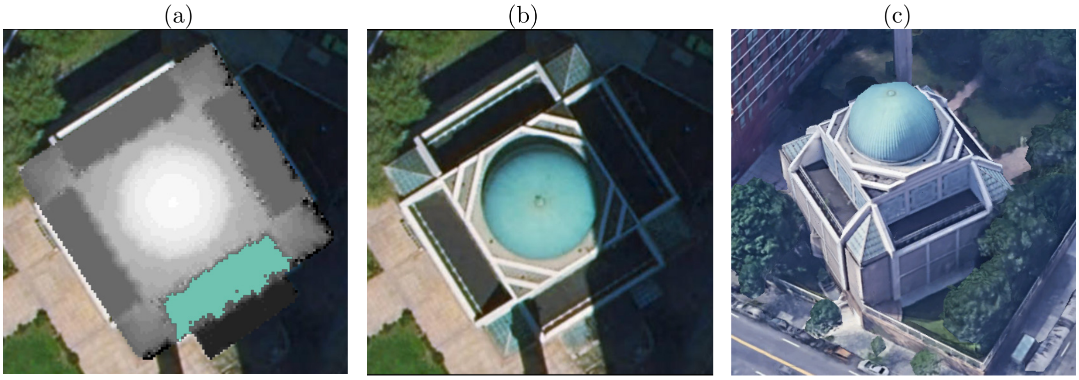


Figure 38. Roof 7 (a) Lidar image with main surface, (b) Satellite image, (c) 3D scene

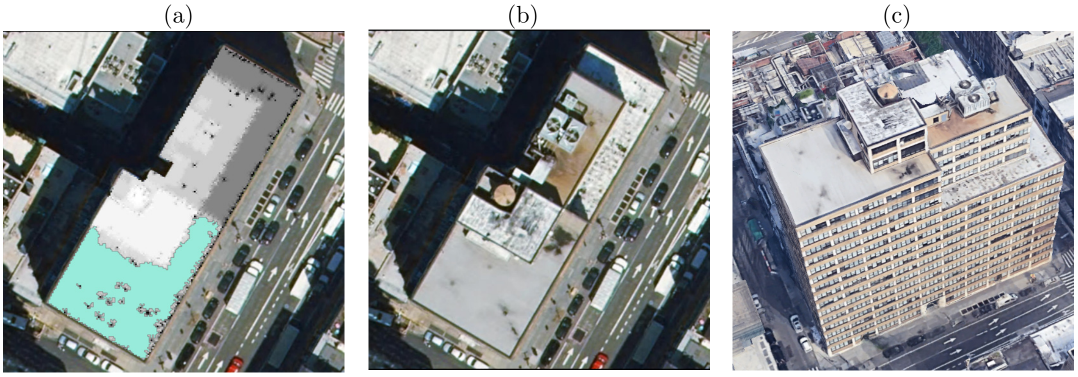


Figure 39. Roof 2 (a) Lidar image with main surface, (b) Satellite image, (c) 3D scene

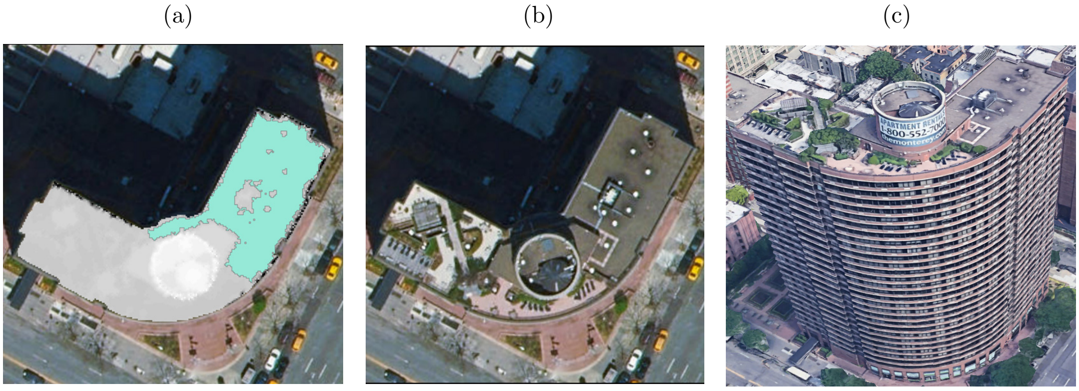


Figure 40. Roof 5 (a) Lidar image with main surface, (b) Satellite image, (c) 3D Scene

Once these attributes have been calculated for each of the 45.515 roofs evaluated, the algorithm takes the area of the main surface as a reference for the estimation of the type of platform to be built. The literature review allowed us to gather enough information to establish estimates of the area necessary for the construction of UAM platforms. Therefore, this thesis has taken as a reference to the platforms typology given by [15, 27, 53] for eVTOL aircraft and has added a special category for UAV. The map in *Figure 41* allows us to appreciate the distribution of the platform types associated with the main suitable area for each of the roofs. Therefore, only considering the main surface of roofs, 33.070 (72.7%) roofs would work for the construction UAV platforms, 11.631 (25.6%) rooftops for installation of Vertistops, 703 (1.5%) rooftops for Vertiports and 57 (0.1%) roofs for the larger Vertihubs. Only 54 roofs were disqualified as they did not have a minimum useful area for UAM platforms. As a complement, *Figure 42* allows us to visualize the distribution of the platform types available for each of the surface levels. Therefore we can deduce that as the area in the surface levels decreases, the number of available platforms also decreases, especially the larger ones such as Vertihubs and Vertiports.

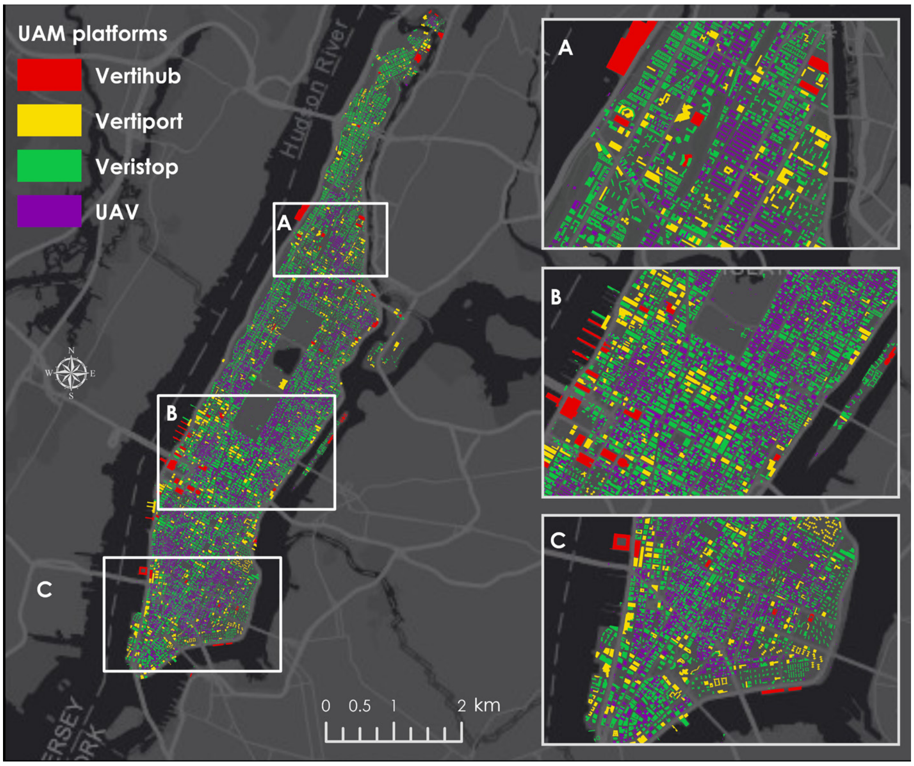


Figure 41. Map distribution for UAM Platforms typology using the main surface level

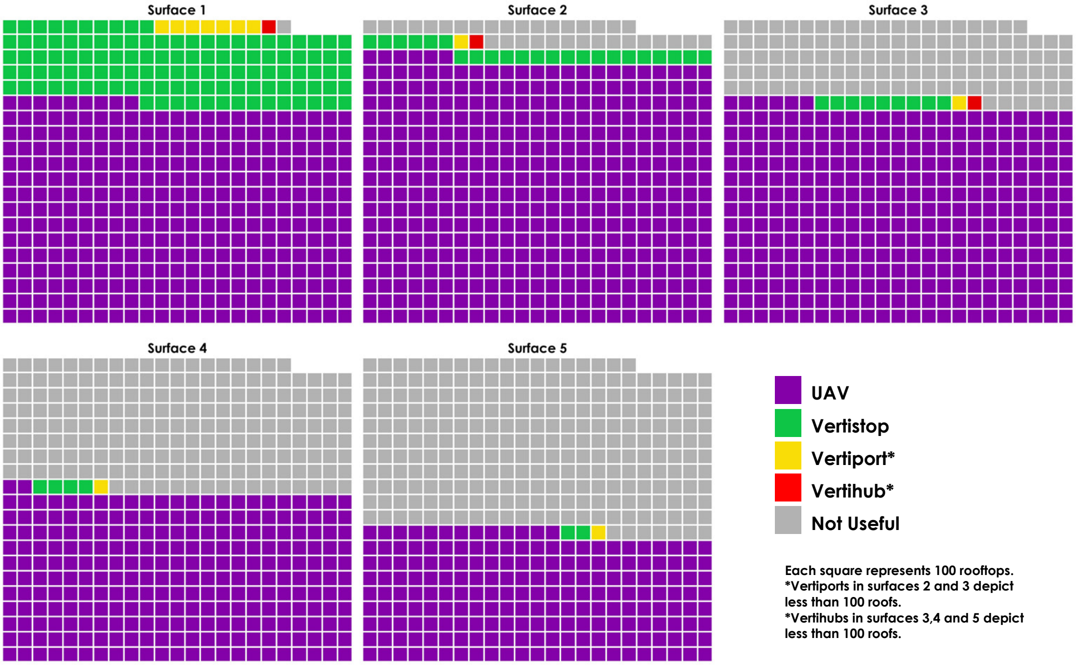


Figure 42. Waffle-chart for UAM typology distribution at different surface levels

An additional and deeper analysis leads to the evaluation of the rooftop flat-surfaces total potential for the development of UAM platforms. Thus, when considering the five main surfaces, *Table 10* shows that 19.993 (44%) roofs could generate 99.965 UAM spots. Thus, for all of Manhattan, 45.461 roofs could have the potential to generate 167.262 places for landing and clearing of vehicles in the UAM system.

Number of flat Surfaces useful for UAM Platforms	Number of Rooftops	%	Total Potential Hubs Locations
1	3294	7.2	3294
2	8443	18.6	16886
3	7807	17.2	23421
4	5924	13.0	23696
5	19993	44.0	99965
Total	45461	100	167262

Table 10. Total potentiality of the rooftops by counting the usable surfaces available

4.2.3 Rooftop Suitability Analysis

The results for the last step of this methodology was the calculation of the ranges of suitability levels for the main surface level. Due to 85% of the main surfaces of all Manhattan roofs occupy more than 50% of the area in their corresponding bounding-boxes (ancillary factor), suitability was estimated only considering the elongation (compactness factor). Subsequently, the values given by the elongation factor were categorized into quartiles for each of the platform types selected in this investigation. The maps *Figure 43*, *Figure 44*, *Figure 45* and *Figure 46* allow us to observe the spatial distribution and final suitability for each of the platform categories according to the main surface. *Figure 47* shows that independent of the number of roofs available for each typology, High and Medium/High suitability indexes are fairly distributed in each UAM platform type. Nevertheless, Medium/High and Medium/Low indexes show a greater number of roofs for each of the UAM platform typologies (*Table 11*). In general terms for all Manhattan, 16.5% (7.512) of the roofs reflect a High suitability, 36.5% (16.608) Medium/High, 44.6% (20.286) Medium/Low and the remaining 2.3% (1.055) a Low suitability. Concerning the UAM type, the UAV platforms and Vertistops have the highest number of roofs available in all suitability indexes.

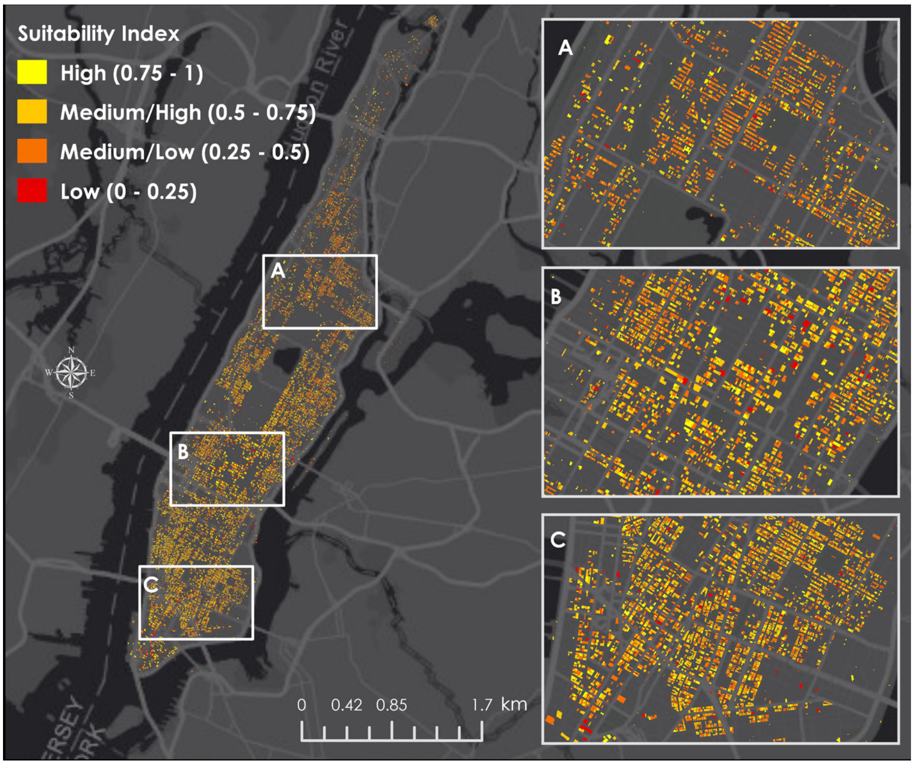


Figure 43. Suitability Index map for UAV platforms

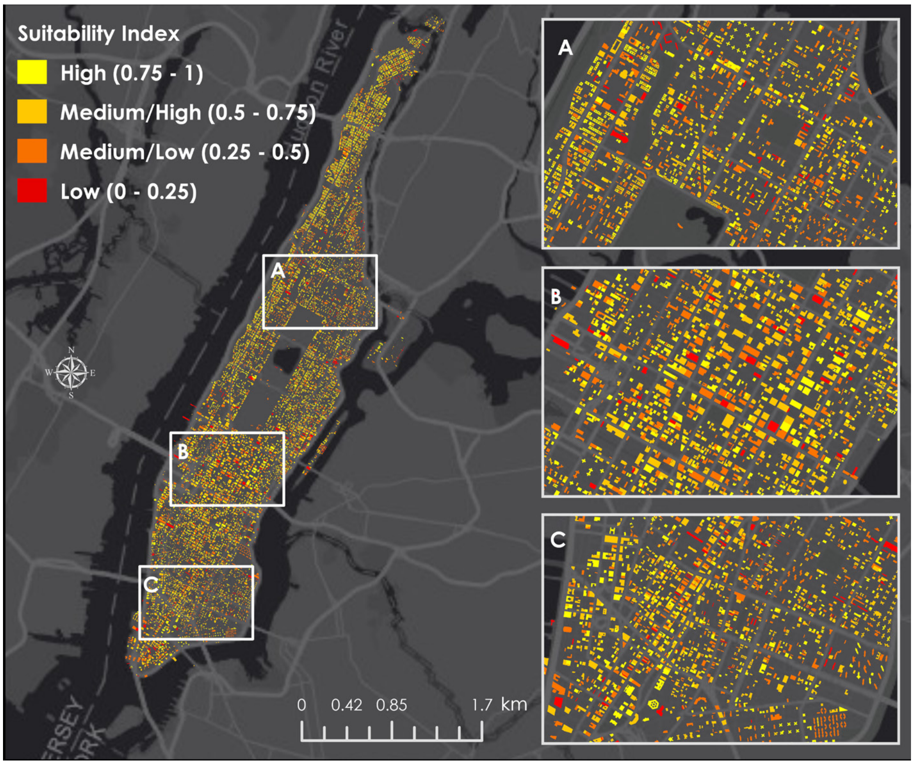


Figure 44. Suitability index map for Vertistops

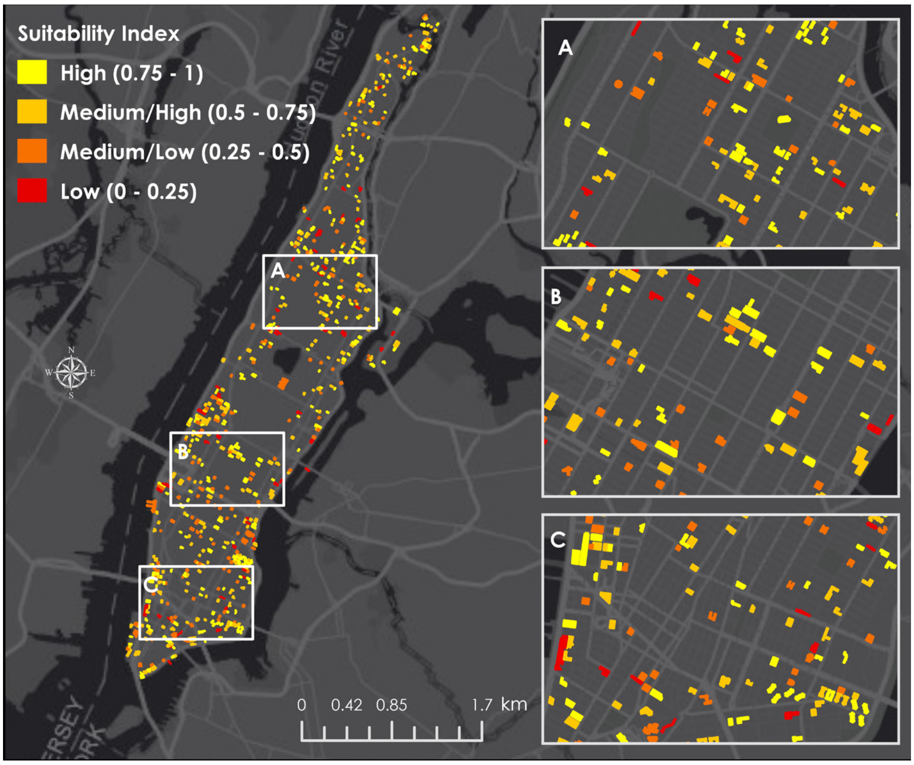


Figure 45. Suitability index map for Vertiports

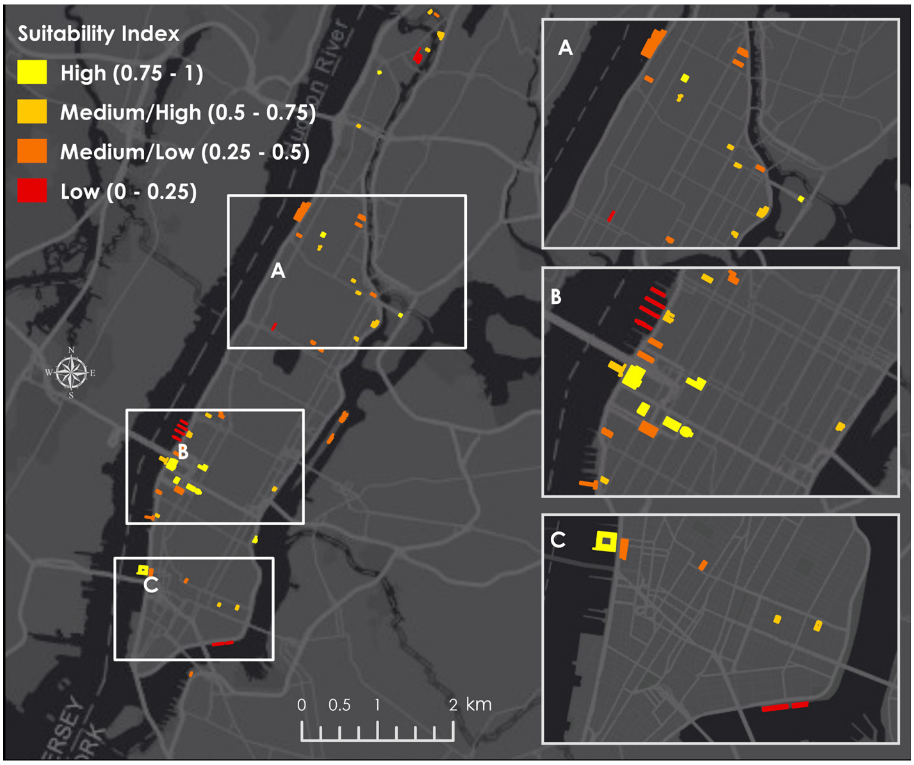


Figure 46. Suitability index map for Vertihubs

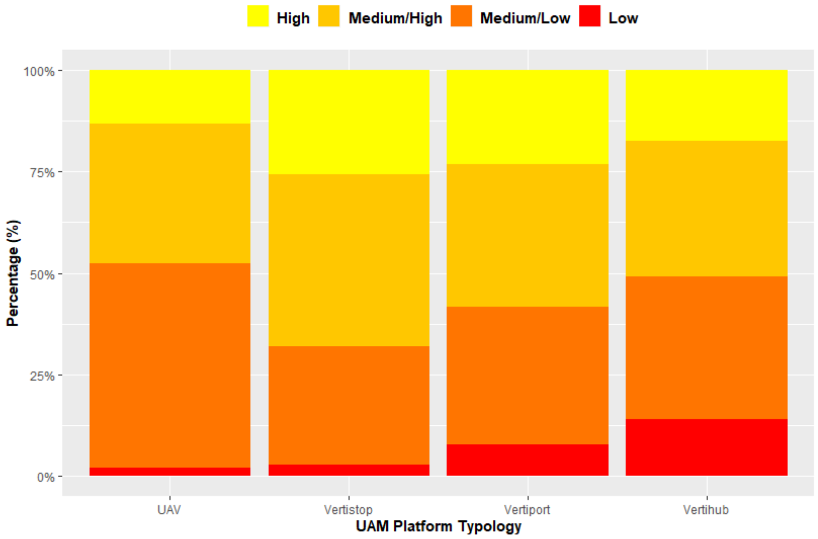


Figure 47. Distribution of the suitability index levels for each UAM platform type

UAM Hub Type	Number of rooftops per Suitability Index				Total	% per UAM Type
	High	Medium/High	Medium/Low	Low		
UAV	4357	11411	16637	665	33070	72.7
Vertistop	2982	4930	3392	327	11631	25.6
Vertiport	163	248	237	55	703	1.5
Vertihub	10	19	20	8	57	0.1
Total	7512	16608	20286	1055	45461	100
% per Index	16.5	36.5	44.6	2.3	100	

Table 11. Summary table showing the distribution of rooftops in each roof suitability index

4.3 Final suitability categorization for UAM Platforms

This section of the investigation shows the results obtained after the combination of place-suitability levels for the census blocks obtained from GeoSOM clusters and rooftop-suitability levels defined after running the rooftop-flatness assessment. The fact that the roofs of the buildings only belong to a census block guarantees that each roof will have a unique classification of total suitability. Therefore, when overlapping maps does not generate uncertainties in the results. However, 76 roofs were not overlapped with the layer of Manhattan census blocks, as most of them are in areas belonging to Central Park and some islands where there is no coverage by the layer of census blocks. So, these were not considered in the final statistics.

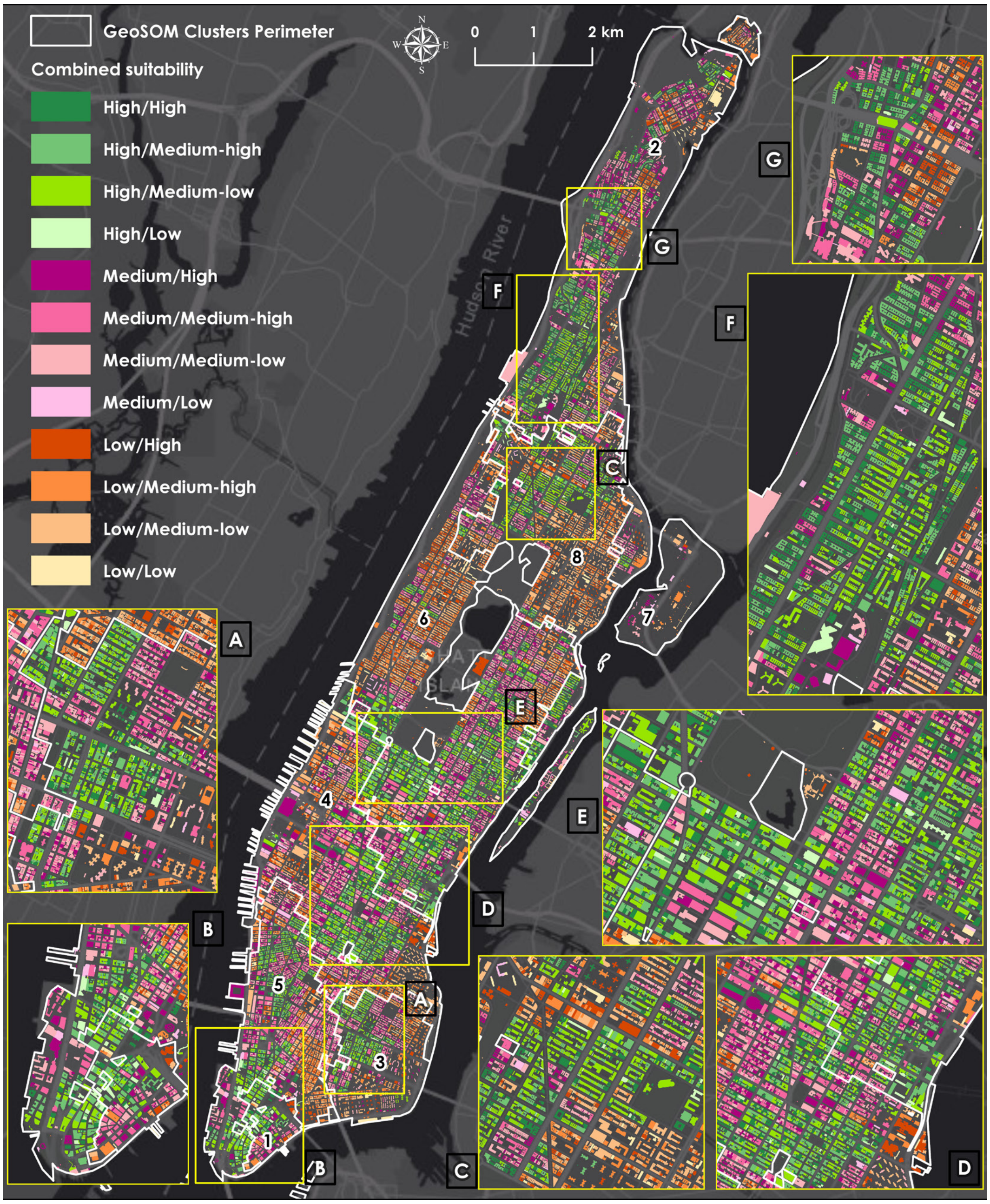


Figure 48. Combined suitability taking the criteria of place and flatness of the main surface on the roofs

Figure 48 depicts the spatial distribution of the 12 combined levels of suitability, where the first suitability category refers to place features and the second one to the rooftop flatness. Roofs with *High/High* and *High/Medium-high* combinations are geographically distributed throughout the study area but tend to be more inland Manhattan. However, near these areas, there are also roofs with low suitability of surfaces for landing and clearance of UAM vehicles but with high place suitability represented by the *High/Low* level. On the other hand, *Medium/High* and *Medium/Medium-high* categories are also scattered all over the study area but show concentrations both near the shoreline and in central areas of the borough. These areas are of great relevance since they can be taken as a second priority for the urban planning of UAM hubs. It also stands out that some roofs in the northern, Downtown, and especially in Midtown Manhattan have suitable surfaces for the settlement of UAM platforms, but urban ecosystem conditions do not show an appropriate place.

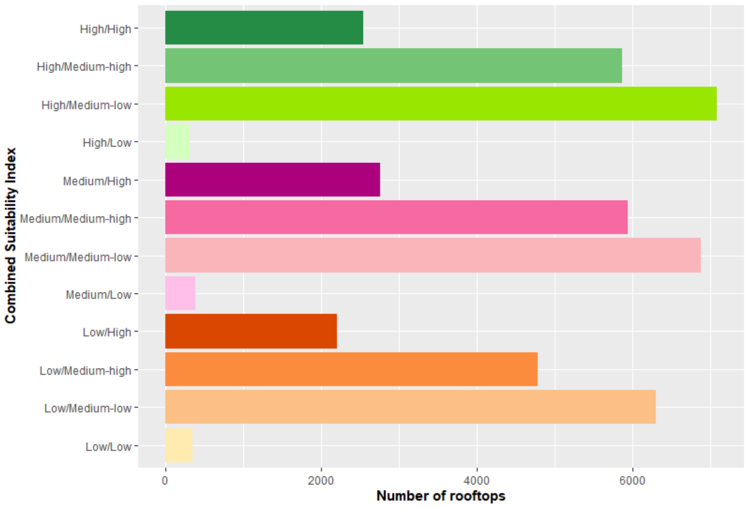


Figure 49. Total of rooftops distributed for each combined suitability class

		Rooftop Suitability Indexes					
		High	Medium-high	Medium-low	Low	Total	% Place Suit.
Place Suitability Indexes	High	2541	5864	7080	311	15796	34.8
	Medium	2752	5937	6877	385	15951	35.1
	Low	2204	4784	6302	348	13638	30.1
Total		7497	16585	20259	1044	45385	100
% Roof Suit.		16.5	36.5	44.6	2.3	100	

Table 12. Distribution of the rooftops in the final categorization of suitability for UAM platforms

It is also interesting to observe how the number of roofs is distributed very similarly in the different levels of integral suitability, particularly for the medium suitability categories (*Figure 49*). Furthermore, there are about 2.500 (5.6%) roofs with High suitability for both criteria and less than 350 (0.8%) with Low suitability (*Table 12*).

The category with the most roofs is the one given by High place suitability and Medium/Low roof suitability with 6.877 (15.6%) rooftops (*Table 12*). According to the platform type, UAVs and Vertistops show a greater number of appropriate roofs with a combined *High/High* suitability, but Vertihubs does not show any roof classified as *High/High* (*Table B- 1– Annex 2, Figure 50*).

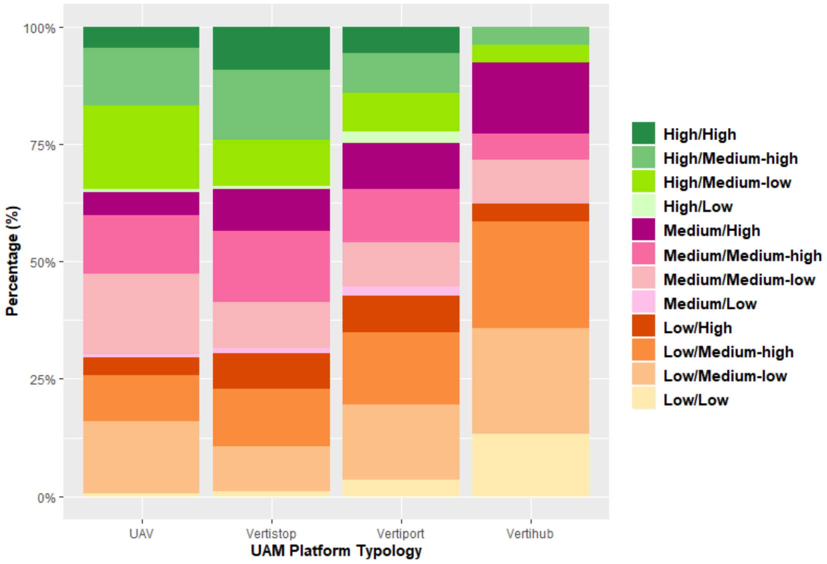


Figure 50. Proportions of combined suitability levels per UAM hub types

Regarding a cluster-oriented analysis, combined suitability ratios show that the *High/High* category is distributed throughout the GeoSOM clusters. Cluster 1 is the one having the highest proportion of this integral suitability level (*Figure 51*). However, when being larger, Cluster 5 and 6 show a greater number of roofs with this category (*Table B- 2– Annex 2*). Categories with medium suitability for prioritization in the construction of UAM hubs (*Medium/High* and *Medium/Medium-High*) show a fair distribution across all clusters, but Cluster 8 shows lower proportions for these categories (*Figure 51*). Besides, Clusters 6 and 8 are the ones that concentrate roofs with low suitability of both place and roof-surface for the development of UAM platforms (*Table B- 2– Annex 2*). As a complementary analysis, amalgamated data also allows us to observe that the type of platforms is fairly distributed for each of the GeoSOM clusters (*Figure 52*). Nonetheless, the only ones that can develop Vertihubs are Clusters 2, 4 and 7. Finally, it is also possible to connect these results with the discriminated descriptions of each cluster generated in Chapter 4.1.5. Thus, potential UAM services described can be carried out by evidencing available roof surfaces for different types of UAM hubs.

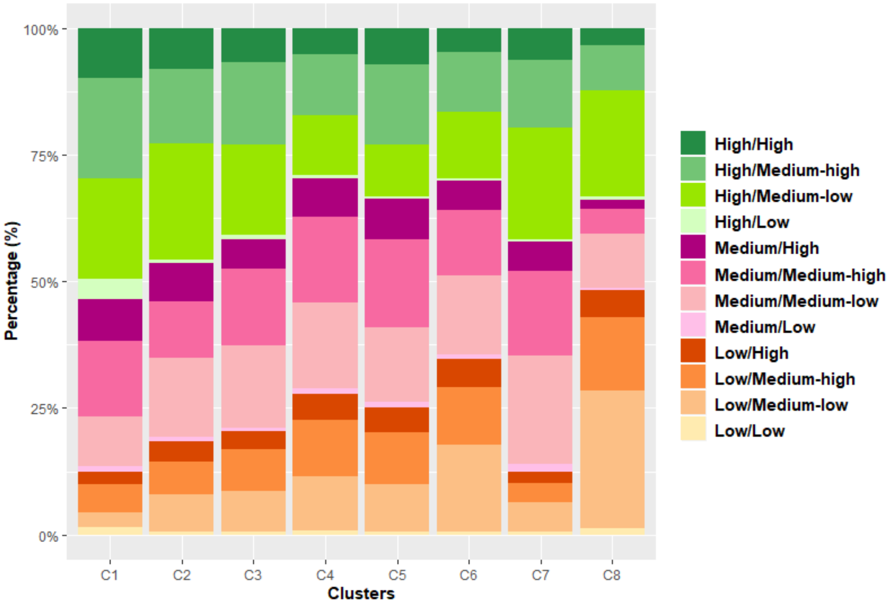


Figure 51. Proportions of combined suitability levels per GeoSOM cluster

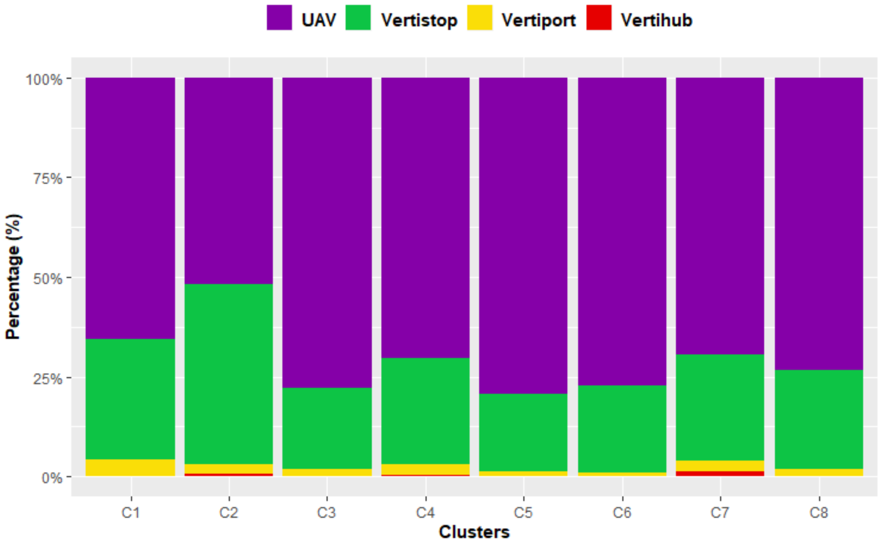


Figure 52. UAM types distribution per GeoSOM clusters

4.4 Algorithm Performance Considerations

Massive and iterative LIDAR data processing and management can be costly both in time and in computational resources [61]. Techniques for accelerating LIDAR data processing were proposed by Wu *et al.* [65] when using algorithms programmed in c++ and by applying GPU approaches [61]. Nonetheless, to improve the processing times of the algorithm programmed in Python, this research has applied parallel processing methods. Rural Agricultural Planning Unit in Colombia (UPRA) [50] has implemented this approach to accelerate geoprocessing times under *ArcPy* library. Therefore, the Python library *subprocess* helps to run several applications/processes at the same time under the condition that the input information can be divided. Thus, the algorithm can be parallelized as many times as RAM and CPU support it. For the machine conditions described previously, five processes were parallelized where each one was responsible for processing ~1800 roofs. To further speed up the processing time, five machines were configured so that each one processed ~9000 roofs using the parallel processing approach (*Figure 53*). Time estimations indicate that for the first part of the algorithm ~19.5 hours was necessary to process ~9000 roofs without using any optimization method; while using five parallel processes, ~3.5 hours was necessary. It addition, the designed algorithm tries to keep in RAM memory the majority of intermediate processes as a method for reducing time, which is a very good practice.

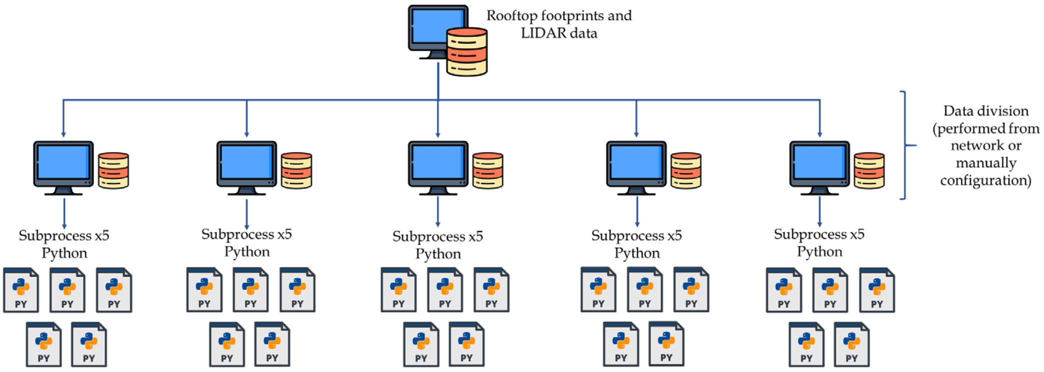


Figure 53. Operational mode when running *subprocess* library for the python algorithm

5. Limitations and Future Work

Several limitations were found throughout the work. First, although New York is a city that has extensive information repositories, not everything is at the level of the census blocks. Therefore, one aspect to take into account for future work would be to acquire more detailed information to distinguish other patterns in the urban ecosystem. Besides, the suitability analysis could include information that was not accessible for this research, such as urban ecological transport nodes (bicycles, scooters, etc.) and variables related to weather conditions. Regarding the roof suitability analysis, it is important to mention that the flatness assessment in some cases, can be affected by several factors. Complex rooftop shapes and the presence of trees or other elements on the roofs probably lead to discrepancies in the estimated values. Moreover, the analysis may be affected by buildings that are being constructed at the time of catching the LIDAR data. Possible temporary differences between the building rooftops footprints layer and LIDAR data can influence when extracting the surfaces, since it could be including two different buildings in the same analysis or even areas that are not part of the building.

Experimenting with other accurate techniques that handle LIDAR data directly for building reconstruction could greatly improve the detail in the calculations. In subsequent investigations, the roof suitability index could be even stricter by combining the surface elongation rate with the percentage of area covered within the bounding-box. Although more classes could probably be generated for the suitability index. Besides, the material from which the roof is made could be included in the analysis. The area ranges proposed for different types of UAM platforms can be adjusted according to expert opinions and new studies. It is remarkable to mention that the algorithm can be extended as a GIS tool to deepen the analysis of results with other approaches such as the one proposed by Castagno *et al.* [11]. Despite this research was focused on finding the total integral potential of the roofs for UAM platforms, 3D geofences such as those proposed by Hildemann *et al.* [25] can be involved in upcoming work. The fact that UAM platforms have not materialized in the study area makes non-viable to validate with real data the results obtained. Notwithstanding, the results of the research can be taken as a guide for urban planners and decision-makers by prioritizing areas as a starting point for UAM hubs.

6. Conclusions

The thesis conducted is aimed at answering three research questions that in a general-purpose seek to establish a robust suitability analysis for the location of UAM hubs. The first approach proposed in the methodology essentially performs a spatial pattern analysis through GeoSOM on census blocks to find clusters with similar behaviors in urban ecosystem variables. Although GeoSOM is a powerful method used in urban planning [24], socio-economic and environmental variables were potentialized beforehand by adding proximity attributes through *Driving Distance* algorithm. GeoSOM clusters are then the input for *Jenks* statistical distribution to categorize census blocks within each cluster according to the importance of the variables by following the literature review. Suitability indexes were achieved when the importance of the variables was ranked and totalized in each census block per GeoSOM cluster. The result of applying this methodology depicts that the most suitable census blocks for UAM hubs development predominate in the Downtown and Midtown of Manhattan, besides a few blocks in the north of the study area. Thus, 29% of the census blocks in Manhattan have high suitability for the location of platforms, 34% medium, and 37% have low suitability. The resulting maps and statistics lead to answer the first research question related to the location of the most suitable areas for the development of UAM hubs.

To answer the second research question, this thesis proposed a method for the extraction and evaluation of the flat surfaces for each roof in the study area. Consequently, the algorithm developed crops the information of the airborne LIDAR data with each of the building rooftop footprints and calculates the corresponding elevation statistics without outliers for each roof. These statistics and especially the standard deviation aid as a proxy for the location of mostly flat roofs. To deepen the categorization of suitability, the second part of the algorithm is responsible for extracting the potential flat surfaces of each roof. To do so, LIDAR raster files are generated and then segmented in a vector way by grouping pixels with similar elevation. At this point, the algorithm evaluates the compactness of the five most extensive surfaces utilizing the elongation factor of polygons, which is taken to categorize the suitability of each flat surface on the roof. The resulting information from the algorithm not only allows evaluating the suitability of the rooftop surfaces but also assessing which is the feasible UAM platform type to be developed. Considering the information obtained from the main flat surface of the rooftops, only 16.5% of the roofs in Manhattan show high suitability and 36.5% Medium-High suitability. In addition, UAVs and Vertistops platforms are the types with the highest number of suitable roofs for their development with 72% and 25.6%, respectively. Furthermore, the data released by the algorithm allow us to deduce that 44% of roofs in Manhattan offer the potential to carry out different UAM platform types on the five main surfaces. Although not directly associated with the results, it is worth mentioning that the inclusion of parallel processing techniques from Python significantly improved the performance of the algorithm when processing LIDAR data massively.

The last stage of the designed methodology is aimed at answering the last research question asked. Therefore, by combining the two approaches proposed in this thesis, both urban ecosystem variables and physical information of existing rooftops are being involved in the analysis. Moreover, the combined approach was designed so that ambiguities are not generated in the final result, because the roofs used are only located in a single census block. The output of this analysis generates an amalgamated suitability map, which makes it possible to distinguish where the most suitable roofs for UAM hubs are located from an integral perspective. Results allow asserting that only 5.6% (2.541) of the roofs have the highest integral for UAM platforms development. These roofs are geographically distributed throughout the study area and become a priority for urban planning of UAM hubs. Conversely, 0.8% (365) of the rooftops in the study area depict low integral suitability to build UAM hubs. Nevertheless, decision-makers can consider roofs with the best *Medium* combined suitabilities as second options for the construction of UAM platforms. Additionally, results allow verifying that in almost all the clusters, most UAM platform types can be carried out, except Vertihubs that can only be built in three specific clusters. Lastly, when considering investigations done by Fadhil [15] and Castagno *et al.* [11], this thesis shows results that offer a better detail level of potential locations for the settlement of UAM hubs.

7. Bibliographic References

[1] AIRBUS. **2018**. *Urban Air Mobility. The sky is yours* (November 2018). Retrieved July 24, 2019 from https://www.airbus.com/newsroom/stories/urban-air-mobility-the-sky-is-yours.html.

[2] AIRBUS. **2019**. *Airbus Vahana Project* (2019). Retrieved February 5, 2020 from https://acubed.airbus.com/projects/vahana/.

[3] ALBANO, R. **2019**. Investigation on Roof Segmentation for 3D Building Reconstruction from Aerial LIDAR Point Clouds. *Applied Sciences* 9, 21, 4674. Retrieved October 10 , 2019 from DOI: https://doi.org/10.3390/app9214674.

[4] ANTCLIFF, K., MOORE, M., and GOODRICH, K., Eds. **2016**. *Silicon Valley as an Early Adopter for On-Demand Civil VTOL Operations*. NASA Langley Research Center; Hampton, VA, United States.

[5] ASGARY, A., SADEGHI, A., and LEVY, J. **2012**. Modeling the risk of structural fire incidents using a self-organizing map. *Fire Safety Journal* 49, 1–9. Retrieved September 18, 2019 from DOI: https://doi.org/10.1016/j.firesaf.2011.12.007.

[6] BAÇÃO, F., LOBO, V., and PAINHO, M. **2005**. Self-organizing Maps as Substitutes for K-Means Clustering. In *Computational science -- ICCS 2005. 5th international conference, Atlanta, GA, USA, May 22-25, 2005 : proceedings*, Vaidy S. Sunderam, Ed. Lecture Notes in Computer Science, 3514. Springer, Berlin, 476–483. Retrieved September 24, 2019 from DOI: https://doi.org/10.1007/11428862_65.

[7] BAÇÃO, F., LOBO, V., and PAINHO, M. **2005**. The self-organizing map, the Geo-SOM, and relevant variants for geosciences. *Computers & Geosciences* 31, 2, 155–163. Retrieved November 18, 2019 from DOI: https://doi.org/10.1016/j.cageo.2004.06.013.

[8] BAST, H., FUNKE, S., SANDERS, P., and SCHULTES, D. **2007**. Fast routing in road networks with transit nodes. *Science (New York, N.Y.)* 316, 5824, 566. Retrieved October 10, 2019 from DOI: https://doi.org/10.1126/science.1137521.

[9] BAUR, S., SCHICKRAM, S., HOMULENKO, A., MARTINEZ, N., and DYSKIN, A. **2018**. *Urban Air Mobility. The rise of a new mode of transportation* (2018). Retrieved September 27, 2019 from https://www.rolandberger.com/en/Publications/Passenger-drones-ready-for-take-off.html.

[10] BELLAMY, W. **2019**. *Helsinki Volocopter Flight Test Evaluates UTM for Air Taxi Operations. Air Taxi, ATM Modernization* (2019). Retrieved September 27, 2019 from https://www.aviationtoday.com/2019/09/04/helsinki-volocopter-flight-test-evaluates-utm-for-air-taxi-operations/.

[11] CASTAGNO, J. and ATKINS, E. **2018**. Roof Shape Classification from LiDAR and Satellite Image Data Fusion Using Supervised Learning. *Sensors (Basel,*

Switzerland) 18, 11. Retrieved September 3, 2019 from DOI: https://doi.org/10.3390/s18113960.

[12] CHEN, D., ZHANG, L., LI, J., and LIU, R. **2012**. Urban building roof segmentation from airborne lidar point clouds. *International Journal of Remote Sensing* 33, 20, 6497–6515. Retrieved November 28, 2019 from DOI: https://doi.org/10.1080/01431161.2012.690083.

[13] CHEN, J., YANG, S., LI, H., ZHANG, B., and LV, J. **2013**. Research on Geographical Environment Unit Division Based on the Method of Natural Breaks (Jenks). *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XL-4/W3, 47–50. Retrieved December 12, 2019 from DOI: https://doi.org/10.5194/isprsarchives-XL-4-W3-47-2013.

[14] CZIOSKA, P., MATTFELD, D., and SESTER, M. **2017**. GIS-based identification and assessment of suitable meeting point locations for ride-sharing. *Transportation Research Procedia* 22, 314–324. Retrieved October 10, 2019 from DOI: https://doi.org/10.1016/j.trpro.2017.03.038.

[15] FADHIL, D. **2018**. *A GIS-Based Analysis for Selecting Ground Infrastructure Locations for Urban Air Mobility. Master's Thesis, Munich, 2018.* Master. Technical University of Munich.

[16] FIEDLER, D., CAP, M., and CERTICKY, M. **2017**. Impact of mobility-on-demand on traffic congestion: Simulation-based study. In *IEEE ITSC 2017. 20th International Conference on Intelligent Transportation Systems : Mielparque Yokohama in Yokohama, Kanagawa, Japan, October 16-19, 2017.* IEEE, Piscataway, NJ, 1–6. Retrieved September 15, 2019 from DOI: https://doi.org/10.1109/ITSC.2017.8317830.

[17] FIRDHOUS, M., BASNAYAKE, D., KODITHUWAKKU, K., HATTHALLA, N., CHARLIN, N., and BANDARA, P. **2010**. Route Advising in a Dynamic Environment – A High-Tech Approach. Innovations in Computing Sciences and Software Engineering.

[18] FOX, A. **2019**. Uber and Lyft may be making San Francisco's traffic worse. *Science.* Retrieved October 18, 2019 from DOI: https://doi.org/10.1126/science.aax9717.

[19] FU, M., ROTHFELD, R., and ANTONIOU, C. **2019**. Exploring Preferences for Transportation Modes in an Urban Air Mobility Environment: Munich Case Study. *Transportation Research Record* 7, 036119811984385. Retrieved September 27, 2019 from DOI: https://doi.org/10.1177/0361198119843858.

[20] GIANQUINTIERI, L., CAIANI, E. G., BRAMBILLA, P., PAGLIOSA, A., VILLA, G. F., and BROVELLI, M. A. **2019**. Open Data in Health-Geomatics: Mapping and Evaluating Publicly Accessible Defibrillators. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-4/W14, 63–70. Retrieved November 10 , 2019 from DOI: https://doi.org/10.5194/isprs-archives-XLII-4-W14-63-2019.

[21] HARRIS, C. **1964**. A scientific method of districting. *Behavioral science* 9, 3, 219–225. Retrieved December 8, 2019 from DOI: https://doi.org/10.1002/bs.3830090303.

[22] HENRIQUES, R. **2010**. *Artificial Intelligence in Geospatial Analysis: applications of Self-Organizing Maps in the context of Geographic Information Science*. PhD Thesis. Instituto Superior de Estatística de Lisboa, Universidade Nova de Lisboa, Lisboa.

[23] HENRIQUES, R., BACAO, F., and LOBO, V. **2010**. Spatial Clustering with SOM and GeoSOM: Case Study of Lisbon's Metropolitan Area. In *GEOProcessing 2010. Proceedings, the Second International Conference on Advanced Geographic Information Systems, Applications, and Services : 10-16 February, 2010, St. Maarten, Netherlands Antilles*. IEEE Computer Society, Los Alamitos, Calif., 148–152. Retrieved November 29, 2019 from DOI: https://doi.org/10.1109/GEOProcessing.2010.30.

[24] HENRIQUES, R., BACAO, F., and LOBO, V. **2012**. Exploratory geospatial data analysis using the GeoSOM suite. *Computers, Environment and Urban Systems* 36, 3, 218–232. Retrieved November 24, 2019 from DOI: https://doi.org/10.1016/j.compenvurbsys.2011.11.003.

[25] HILDEMANN, M. and DELGADO, C. **2019**. *An adaptable and scalable least cost network for air-taxis in urban areas Study area: Manhattan, New York*. In: Kyriakidis, P., Hadjimitsis, D., Skarlatos, D. & Mansourian, A. (Eds.), 2019. Accepted Short Papers and Posters from the 22nd AGILE Conference on Geo-information Science. Cyprus University of Technology 17-20 June 2019, Limassol, Cyprus. ISBN 978-90-816960-9-8.

[26] HIMBERG, J. **2000**. A SOM based cluster visualization and its application for false coloring. In *International joint conference on neural networks*. IEEE, 587-592 vol.3. Retrieved December 28, 2019 from DOI: https://doi.org/10.1109/IJCNN.2000.861379.

[27] HOLDEN, J. and GOEL, N. **2018**. *Uber Elevate. Fast-Forwarding to a Future of On-Demand Urban Air Transportation*. Uber. Retrieved September 27, 2019 from https://www.uber.com/elevate.pdf.

[28] HOLMES, B., PARKER, R., STANLEY, D., MCHUGH, P., GARROW, L., and MASSON, P. **2017**. *NASA Strategic Framework for On-Demand Air Mobility. A Report for NASA Headquarters Aeronautics Research Mission Directorate* (2017).

[29] INRIX. **2019**. *Global Traffic Scorecard. INRIX Research* (2019). Retrieved September 27, 2019 from https://inrix.com/scorecard/.

[30] JUNG, S. and HYUNSU, K. **2017**. Analysis of Amazon Prime Air UAV Delivery Service. *Journal of Knowledge Information Technology and Systems* 12, 2, 253–266. Retrieved October 15, 2019 from DOI: https://doi.org/10.34163/jkits.2017.12.2.005.

[31] KASKI, S., NIKKILÄ, J., KOHONEN, T., Ed. **1998**. *Methods for interpreting a self-organized map in data analysis.*

[32] KOHONEN, T. **1982**. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 1, 59–69. Retrieved November 27, 2019 from DOI: https://doi.org/10.1007/BF00337288.

[33] KOHONEN, T. **2013**. Essentials of the self-organizing map. *Neural Networks* 37, 52–65. Retrieved November 16 , 2019 from DOI: https://doi.org/10.1016/j.neunet.2012.09.018.

[34] LI, L., DU, Q., REN, F., and MA, X. **2019**. Assessing Spatial Accessibility to Hierarchical Urban Parks by Multi-Types of Travel Distance in Shenzhen, China. *International journal of environmental research and public health* 16, 6. Retrieved October 15, 2019 from https://doi.org/10.3390/ijerph16061038.

[35] LILIUM AVIATION. **2019**. *Mission: Aircraft for everyone* (2019). Retrieved from https://lilium.com/our-beliefs.

[36] LIU, R., LIU, H., KWAK, D., XIANG, Y., BORCEA, C., NATH, B., and IFTODE, L. **2016**. Balanced traffic routing: Design, implementation, and evaluation. *Ad Hoc Networks* 37, 14–28. Retrieved September 29, 2019 from DOI: https://doi.org/10.1016/j.adhoc.2015.09.001.

[37] LOURENCO, J. **2009**. *Visualization of Clusters in Geo-referenced Data Using Three-dimentional Self-Organizing Maps*. Master Thesis. Universidade Nova de Lisboa, Lisboa.

[38] MENEGUETTE, R., DE GRANDE, R., and LOUREIRO, A. **2018**. *Intelligent Transport System in Smart Cities: Aspects and Challenges of Vehicular Networks and Cloud*. Springer International Publishing.

[39] MOHAPATRA, S. and BHUYAN, P. **2012**. Self Organizing Map of Artificial Neural Network for Defining Level of Service Criteria of Urban Streets. *IJTTE* 2, 3, 236–252. Retrieved November 21, 2019 from DOI: https://doi.org/10.7708/ijtte.2012.2(3).06.

[40] MOORE, M. **2006**. The Third Wave of Aeronautics: On-Demand Mobility. In *NASA Personal Air Transportation Technologies*. SAE Technical Paper Series. SAE International400 Commonwealth Drive, Warrendale, PA, United States. Retrieved September 22, 2019 from DOI: https://doi.org/10.4271/2006-01-2429.

[41] MOORE, M., Ed. **2006**. *NASA Personal Air Transportation Technologies*. SAE Technical Paper Series. SAE International400 Commonwealth Drive, Warrendale, PA, United States. Retrieved September 22, 2019 from DOI: https://doi.org/10.4271/2006-01-2413.

[42] MOOSAVI, V. **2017**. Contextual mapping: Visualization of high-dimensional spatial patterns in a single geo-map. *Computers, Environment and Urban Systems* 61, 1–12. Retrieved November 14, 2019 from DOI: https://doi.org/10.1016/j.compenvurbsys.2016.08.005.

[43] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. **2018**. *NASA UAM Market Study – Final Report* Retrieved September 30, 2019 from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20190001472.pdf.

[44] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. **2018**. *Urban Air Mobility Airspace Integration Concepts and Considerations.* (2018).

Retrieved July 19, 2019 from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20180005218.pdf.

[45] NÜST, D., GRANELL, C., SILERYTE, R., HOFER, B., and OSTERMANN, F. **2019**. *Reproducibility of GIScience MSc Theses - Self-assessment experiment.* OSF Retrieved February 10, 2020 from DOI: https://doi.org/10.17605/osf.io/j97zp.

[46] O'HEAR, S. **2019**. *Lilium unveils five-seater air taxi prototype after a successful maiden flight for its latest jet* (2019). Retrieved September 27, 2019 from https://techcrunch.com/2019/05/15/lilium-5-seater/.

[47] ONYENEKE, C. **2018**. Modeling the Effects of Traffic Congestion on Economic Activities - Accidents, Fatalities and Casualties. *BSI* 3, 2, 7. Retrieved September 29, 2019 from DOI: https://doi.org/10.11648/j.bsi.20180302.11.

[48] PORSCHE CONSULTING. **2018**. *The Future of Vertical Mobility. Sizing the market for passenger, inspection, and goods services until 2035* (2018). Retrieved September 27, 2019 from https://www.porsche-consulting.com/en/press/insights/detail/study-the-future-of-vertical-mobility/.

[49] RODRIGUE, J-P., COMTOIS, C., and SLACK, B. **2017**. *The geography of transport systems* (4th edition). Routledge Taylor & Francis Group, London, New York ISBN: 978-1-31-561815-9.

[50] RURAL AGRICULTURAL PLANNING UNIT. **2018**. *UPRA Geoprocessing optimized tools. Github Online Repository* (2018). Retrieved from https://github.com/UpraAnalisis.

[51] SALONEN, M. and TOIVONEN, T. **2013**. Modelling travel time in urban networks: comparable measures for private car and public transport. *Journal of Transport Geography* 31, 143–153. Retrieved August 21, 2019 from DOI: https://doi.org/10.1016/j.jtrangeo.2013.06.011.

[52] S SCHNELL, J., NAIK, V., HOROWITZ, L., PAULOT, F., GINOUX, P., ZHAO, M., and HORTON, D. **2019**. Air quality impacts from the electrification of light-duty passenger vehicles in the United States. *Atmospheric Environment* 208, 95–102. Retrieved September 21, 2019 DOI: https://doi.org/10.1016/j.atmosenv.2019.04.003.

[53] SEELEY, B. **2017**. Regional Sky Transit IV: Pocket Airpark Design Constraints. In *17th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics, Reston, Virginia, 2017. Retrieved October 11, 2019 DOI: https://doi.org/10.2514/6.2017-3440.

[54] SHAN, J. and SAMPATH, A. **2009**. Building Extraction from LiDAR Point Clouds Based on Clustering Techniques. In *Topographic laser ranging and scanning. Principles and processing*, Jie Shan and Charles K. Toth, Eds. CRC, Boca Raton, 421–444. Retrieved September 20, 2019 DOI: https://doi.org/10.1201/9781420051438.ch15.

[55] SUASNEWS. **2017**. *NASA Embraces Urban Air Mobility (UAM), Calls for Market Study* (2017). Retrieved September 26, 2019 from

https://www.suasnews.com/2017/11/nasa-embraces-urban-air-mobility-uam-calls-market-study/.

[56] THIPPHAVONG, D., APAZA, R., BARMORE, B., BATTISTE, V., BURIAN, B., DAO, Q., FEARY, M., GO, S., GOODRICH, K., HOMOLA, J., IDRIS, H., KOPARDEKAR, P., LACHTER, J., NEOGI, N., NG, H., OSEGUERA-LOHR, R., PATTERSON, M., and VERMA, S. **2018**. Urban Air Mobility Airspace Integration Concepts and Considerations. In *2018 Aviation Technology, Integration, and Operations Conference, June 25-29, 2018, Atlanta, Georgia. AIAA AVIATION Forum*. American Institute of Aeronautics and Astronautics, [Reston (VA)], 2018. Retrieved October 21, 2019 DOI: https://doi.org/10.2514/6.2018-3676.

[57] TOBLER, W. **1970**. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography* 46, 234. Retrieved November 21, 2019 DOI: https://doi.org/10.2307/143141.

[58] ULTSCH, A. and SIEMON, H., Eds. **1990**. *Kohonen's self organizing feature maps for exploratory data analysis*. Kluwer Academic press.

[59] VASCIK, P., BALAKRISHNAN, H., and HANSMAN, R., Eds. **2018**. *Assessment of Air Traffic Control for Urban Air Mobility and Unmanned Systems*.

[60] VASCIK, P. and HANSMAN, R. **2018**. Scaling Constraints for Urban Air Mobility Operations: Air Traffic Control, Ground Infrastructure, and Noise. In *2018 Aviation Technology, Integration, and Operations Conference, June 25-29, 2018, Atlanta, Georgia. AIAA AVIATION Forum*. American Institute of Aeronautics and Astronautics, [Reston (VA)], 413. Retrieved September 19, 2019 DOI: https://doi.org/10.2514/6.2018-3849.

[61] VENUGOPAL, V. and KANNAN, S. **2013**. Accelerating real-time LiDAR data processing using GPUs. In *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, Piscataway, 1168–1171. Retrieved December 5, 2019 DOI: https://doi.org/10.1109/MWSCAS.2013.6674861.

[62] VESANTO, J. **1999**. SOM-based data visualization methods. *Intelligent Data Analysis* 3, 2, 111–126. Retrieved November 17, 2019 DOI: https://doi.org/10.1016/S1088-467X(99)00013-X.

[63] VIEIRA, C. and LUNA, M. **2016**. Models and Methods for Logistics Hub Location: A Review Towards Transportation Networks Design. *Pesqui. Oper.* 36, 2, 375–397. Retrieved September 12, 2019 DOI: https://doi.org/10.1590/0101-7438.2016.036.02.0375.

[64] VOLOCOPTER. **2019**. *Pioneering the Urban Air Taxi Revolution. White Paper* (2019). Retrieved September 27, 2019 from https://press.volocopter.com/images/pdf/Volocopter-WhitePaper-1-0.pdf.

[65] WU, T., HU, X., and YE, L. **2016**. Fast and Accurate Plane Segmentation of Airborne LiDAR Point Cloud Using Cross-Line Elements. *Remote Sensing* 8, 5, 383. Retrieved October 9, 2019 DOI: https://doi.org/10.3390/rs8050383.

[66] YU, W. **2016**. Spatial co-location pattern mining for location-based services in road networks. *Expert Systems with Applications* 46, 324–335. Retrieved November 29, 2019 DOI: https://doi.org/10.1016/j.eswa.2015.10.010.

Annex 1

Id				Normalization	Iterations	Radio	Alpha	Iterations	Radio	Alpha	Q	
Attemtp	X	Y	Lattice	Type	(Rough)	(Rough)	(Rough)	(Finetune)	(Finetune)	(Finetune)	Error	T Error
1	10	1	Hex	Z_Score	10000	4	0.3	20000	2	0.1	4.62871	0.03248
2	5	5	Hex	Z_Score	2500	5	0.3	5000	3	0.1	4.3898	0.014787
3	5	5	Rec	Z_Score	2500	5	0.3	5000	3	0.1	4.36213	0.028783
4	10	5	Hex	Z_Score	2500	5	0.3	5000	3	0.1	4.12333	0.02403
5	10	5	Hex	Z_Score	8000	8	0.4	16000	4	0.2	4.0094	0.025614
6	5	10	Hex	Z_Score	3000	5	0.3	6000	3	0.1	3.99226	0.017428
7	10	10	Hex	Z_Score	6000	5	0.3	12000	3	0.1	3.61997	0.024229
8	15	10	Hex	Z_Score	10000	5	0.3	20000	3	0.1	3.36671	0.019277
9	10	15	Hex	Z_Score	10000	7	0.4	20000	4	0.2	3.36263	0.016899
10	15	10	Hex	Z_Score	10000	10	0.5	20000	5	0.2	3.32511	0.016372
11	10	1	Hex	Min-Max	10000	4	0.3	20000	2	0.1	0.91351	0.02403
12	5	5	Hex	Min-Max	2500	5	0.3	5000	3	0.1	0.861302	0.004489
13	5	5	Rec	Min-Max	2500	5	0.3	5000	3	0.1	0.846135	0.022181
14	10	5	Hex	Min-Max	2500	6	0.3	5000	3	0.1	0.760353	0.015316
15	5	10	Rec	Min-Max	3000	5	0.3	6000	3	0.1	0.757865	0.007394
16	10	5	Hex	Min-Max	8000	8	0.4	16000	4	0.2	0.757237	0.011883
17	10	10	Rec	Min-Max	6000	5	0.3	12000	3	0.1	0.664587	0.00977
18	10	15	Hex	Min-Max	10000	7	0.4	20000	4	0.2	0.61566	0.015844
19	15	10	Rec	Min-Max	10000	10	0.5	20000	5	0.2	0.605454	0.016108
20	15	10	Hex	Min-Max	10000	5	0.3	20000	3	0.1	0.600095	0.015844

Table A- 1. Different parameter settings used for the sensitivity analysis of the SOM algorithm execution

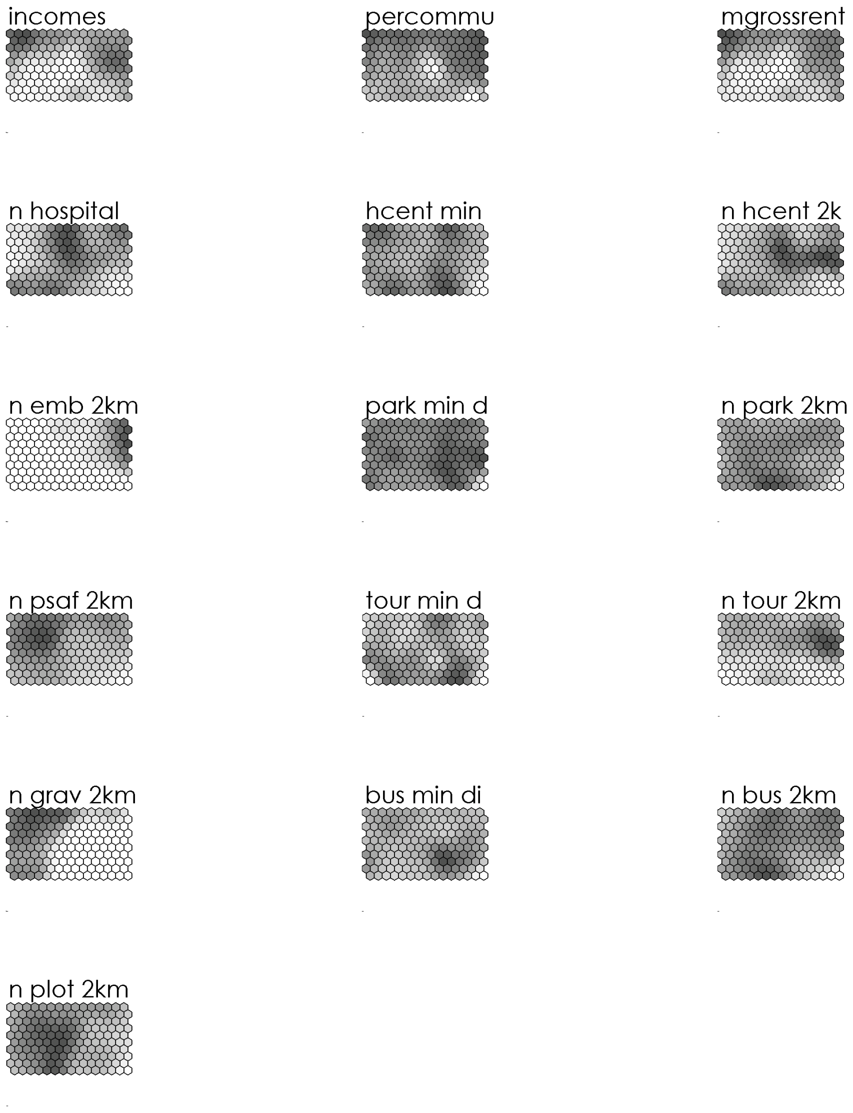


Figure A- 1. Component Planes from SOM algorithm part 1

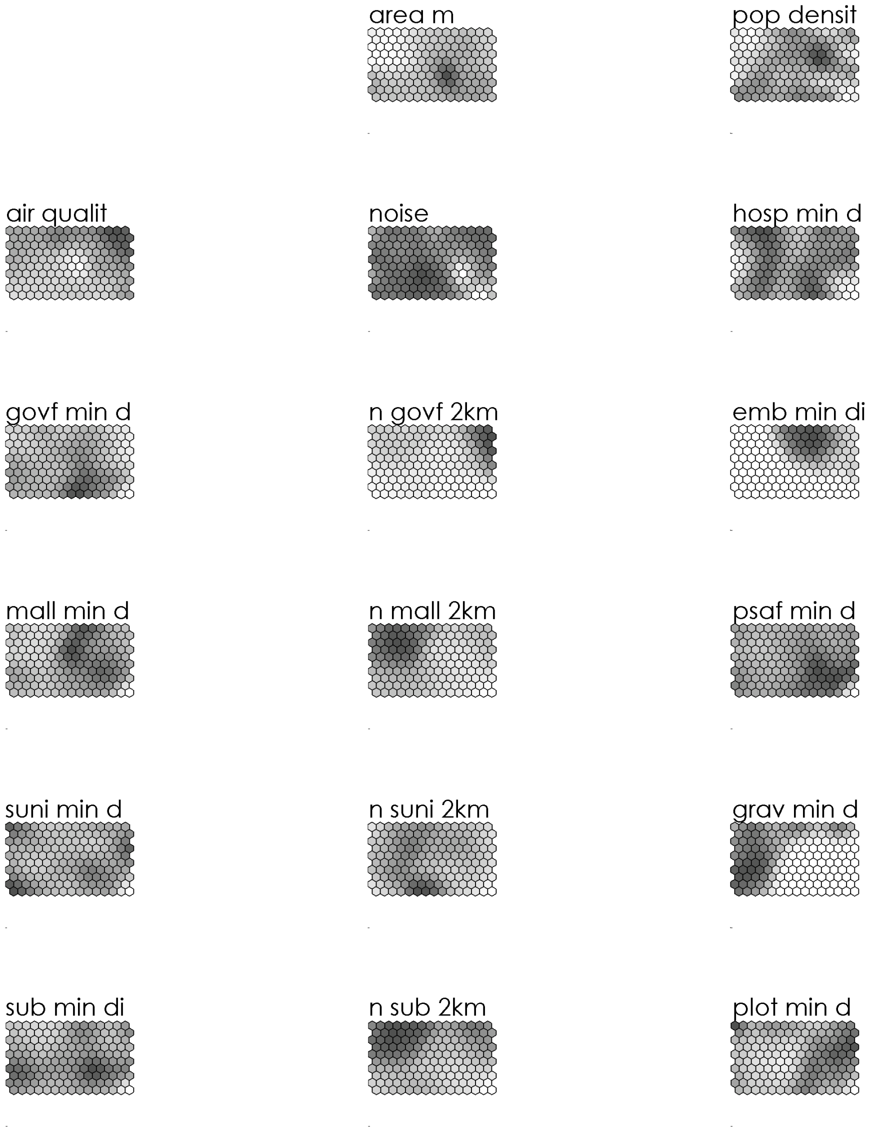


Figure A- 2. Component Planes from SOM algorithm part 2

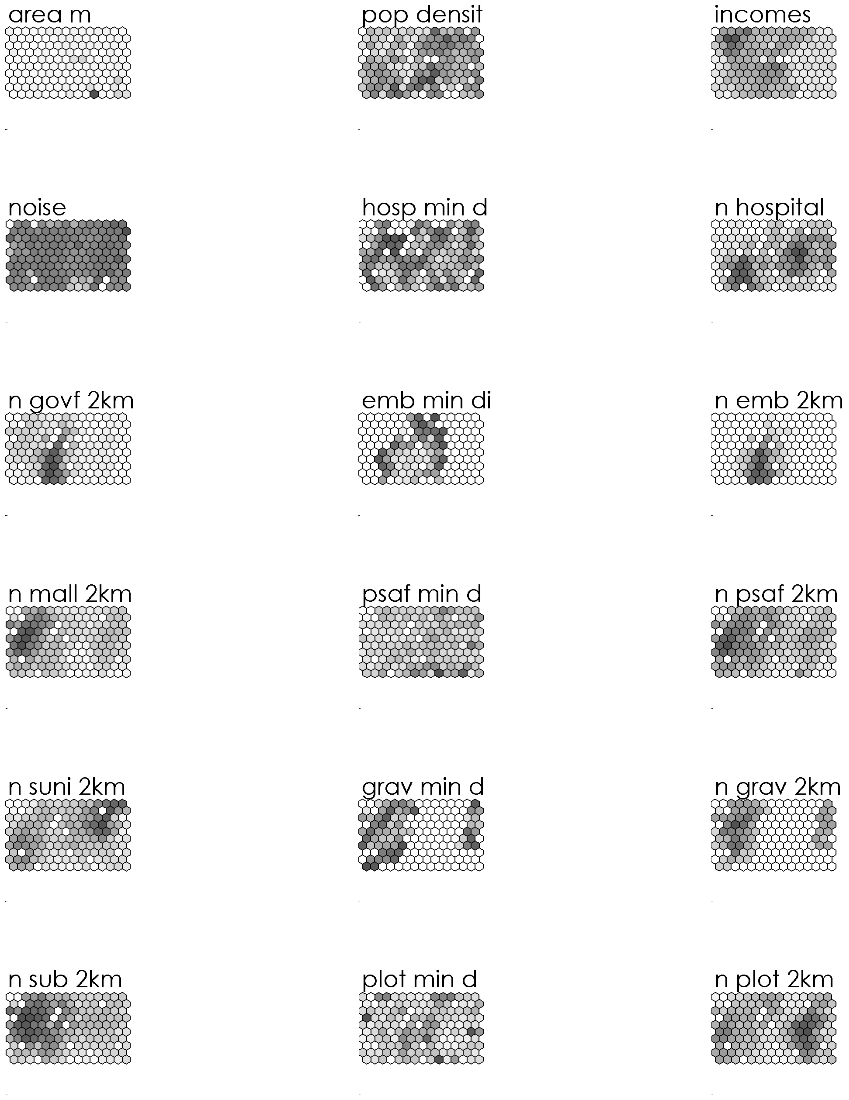


Figure A- 3. Component Planes from GeoSOM algorithm part 1

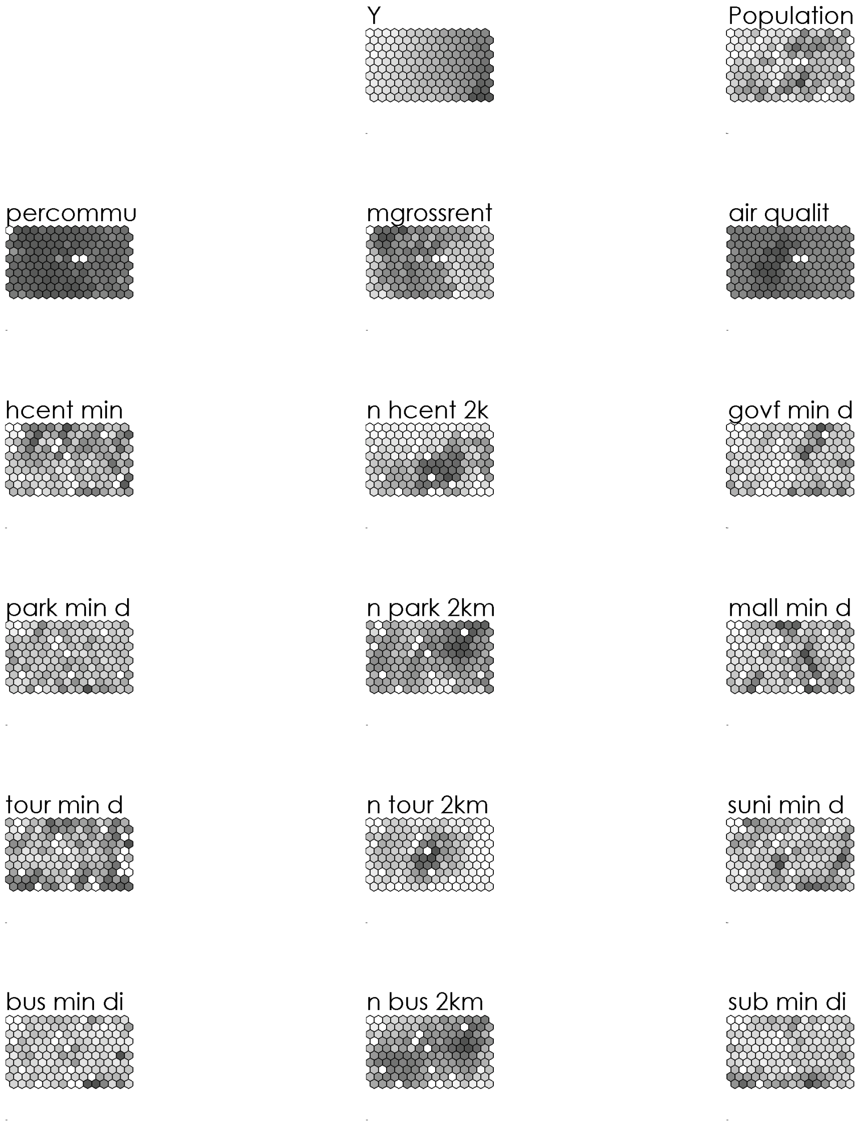


Figure A- 4. Component Planes from GeoSOM algorithm part 2

Annex 2

Integral Suitability Levels	UAM Platforms type			
	UAV	**Vertistop**	**Vertiport**	**Vertihub**
High/High	1442	1061	38	-
High/Medium-high	4063	1739	60	2
High/Medium-low	5902	1118	58	2
High/Low	210	85	16	-
Medium/High	1642	1032	70	8
Medium/Medium-high	4097	1758	79	3
Medium/Medium-low	5653	1153	66	5
Medium/Low	254	117	14	-
Low/High	1262	886	54	2
Low/Medium-high	3240	1425	107	12
Low/Medium-low	5068	1110	112	12
Low/Low	196	120	25	7
Total	33029	11604	699	53

Table B- 1. Distribution of integral suitability per UAM platform type

Integral Suitability Levels	GeoSOM Clusters							
	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
High/High	50	500	218	370	494	610	73	226
High/Medium-high	103	931	530	883	1108	1557	157	595
High/Medium-low	102	1435	583	855	708	1713	262	1422
High/Low	20	39	24	63	36	78	5	46
Medium/High	43	481	193	548	560	744	68	115
Medium/Medium-high	77	697	496	1233	1219	1692	198	325
Medium/Medium-low	51	988	528	1246	1025	2056	251	732
Medium/Low	5	53	23	83	67	115	19	20
Low/High	13	254	113	366	342	724	27	365
Low/Medium-high	28	410	269	821	720	1511	43	982
Low/Medium-low	15	456	262	781	649	2236	68	1835
Low/Low	8	37	22	56	47	87	8	83
Total	515	6281	3261	7305	6975	13123	1179	6746

Table B- 2. Distribution of integral suitability per GeoSOM clusters

Annex 3

SQL Code for *Driving Distance* Algorithm

Complete code in: https://github.com/carlosjdelgadonovaims/rooftop-place_suitability_analysis_for_Urban_Air_Mobility_hubs

```
#######################################
/*Creation of the catchment table*/

CREATE TABLE public.manhattan_catchment_nodes as
(
WITH
nodes AS (
SELECT array_agg(n_id) AS nodes from census_blocks.closest_block_man_node
where b_id<= 4000)
SELECT from_v as start_node, node as end_node, agg_cost as cost from nodes,
pgr_drivingdistance(
    'SELECT gid as id, source as source, target as target, length_m as cost
FROM public.ways'::text, nodes, 2000, false)
)


/*Adding the field node_id to the original table of census blocks*/

ALTER TABLE census_blocks.f_manh_blocks_pop_4326
ADD column node_id bigint;
UPDATE census_blocks.f_manh_blocks_pop_4326
set node_id= (SELECT n_id from (
(SELECT n_id, res.id
FROM census_blocks.f_manh_blocks_pop_4326 as res
LEFT JOIN census_blocks.closest_block_man_node
ON res.id = census_blocks.closest_block_man_node.b_id))c
WHERE c.id = census_blocks.f_manh_blocks_pop_4326.id)


/*Creation of the closest nodes for each entity*/


CREATE TABLE poi.closest_hcenters_node as
SELECT
  hc.id as b_id,
  nodes.id as n_id,
  ST_Distance(geography(hc.geom), geography(nodes.the_geom)) as distance
FROM
   (SELECT DISTINCT ON (id, geom) *
    FROM poi.f_health_centers_4326
    WHERE id IS NOT NULL) AS hc
CROSS JOIN LATERAL
   (SELECT id, the_geom
    FROM public.ways_vertices_pgr
    ORDER BY hc.geom <-> the_geom
    LIMIT 1) AS nodes



CREATE TABLE poi.closest_government_node as
SELECT
```

```
  gov.id as b_id,
  nodes.id as n_id,
  ST_Distance(geography(gov.geom), geography(nodes.the_geom)) as distance
FROM
  (SELECT DISTINCT ON (id, geom) *
   FROM poi.f_government_facilities_4326
   WHERE id IS NOT NULL) AS gov
CROSS JOIN LATERAL
  (SELECT id, the_geom
   FROM public.ways_vertices_pgr
   ORDER BY gov.geom <-> the_geom
   LIMIT 1) AS nodes


CREATE TABLE poi.closest_embassies_node as
SELECT
  e.id as b_id,
  nodes.id as n_id,
  ST_Distance(geography(e.geom), geography(nodes.the_geom)) as distance
FROM
  (SELECT DISTINCT ON (id, geom) *
   FROM poi."f_embassies_point_4326_Int"
   WHERE id IS NOT NULL) AS e
CROSS JOIN LATERAL
  (SELECT id, the_geom
   FROM public.ways_vertices_pgr
   ORDER BY e.geom <-> the_geom
   LIMIT 1) AS nodes


/*Creation and feeding of the table minimum cost when considering all the
entities*/

CREATE TABLE poi.mom_minimum_cost AS
(select '1' as poi_type, start_node, min(cost) as cost from
(select fee.start_node, fee.end_node, cost from
public.manhattan_catchment_nodes as fee
    WHERE (start_node, end_node) in
    (select node_id, pe.n_id from census_blocks.f_manh_blocks_pop_4326,
poi.closest_hospital_node as pe))d GROUP BY d.start_node)

INSERT INTO poi.mom_minimum_cost
(select '2' as poi_type, start_node, min(cost) as cost from
(select fee.start_node, fee.end_node, cost from
public.manhattan_catchment_nodes as fee
    WHERE (start_node, end_node) in
    (select node_id, pe.n_id from census_blocks.f_manh_blocks_pop_4326,
poi.closest_hcenters_node as pe))d GROUP BY d.start_node)

INSERT INTO poi.mom_minimum_cost
(select '3' as poi_type, start_node, min(cost) as cost from
(select fee.start_node, fee.end_node, cost from
public.manhattan_catchment_nodes as fee
    WHERE (start_node, end_node) in
    (select node_id, pe.n_id from census_blocks.f_manh_blocks_pop_4326,
poi.closest_government_node as pe))d GROUP BY d.start_node)




/*Creation and feeding of the table which has the number of entities within
2km when considering all the entities*/

CREATE TABLE poi.mom_counts_2km AS
```

```
(select '1' as poi_type, f.start_node, count(f.start_node) as cnts from (
select distinct fee.start_node, fee.end_node, cost from
public.manhattan_catchment_nodes as fee
    WHERE (start_node, end_node) in
    (select node_id, n_id from census_blocks.f_manh_blocks_pop_4326,
poi.closest_hospital_node))f group by start_node)

INSERT INTO poi.mom_counts_2km
(select '2' as poi_type, f.start_node, count(f.start_node) as cnts from (
select distinct fee.start_node, fee.end_node, cost from
public.manhattan_catchment_nodes as fee
    WHERE (start_node, end_node) in
    (select node_id, n_id from census_blocks.f_manh_blocks_pop_4326,
poi.closest_hcenters_node))f group by start_node)

INSERT INTO poi.mom_counts_2km
(select '3' as poi_type, f.start_node, count(f.start_node) as cnts from (
select distinct fee.start_node, fee.end_node, cost from
public.manhattan_catchment_nodes as fee
    WHERE (start_node, end_node) in
    (select node_id, n_id from census_blocks.f_manh_blocks_pop_4326,
poi.closest_government_node))f group by start_node)



/* Altering the tables with the new fields */

ALTER TABLE census_blocks.f_manh_blocks_pop_4326
ADD column hosp_min_dist_m double precision;
UPDATE census_blocks.f_manh_blocks_pop_4326
set hosp_min_dist_m = (SELECT cost from (SELECT poi_type, start_node, cost
from poi.mom_minimum_cost WHERE poi_type='1')f
                   WHERE f.start_node =
census_blocks.f_manh_blocks_pop_4326.node_id)


ALTER TABLE census_blocks.f_manh_blocks_pop_4326
ADD column n_hospital_2km bigint;
UPDATE census_blocks.f_manh_blocks_pop_4326
SET n_hospital_2km = (SELECT cnts from (SELECT poi_type, start_node, cnts
from poi.mom_counts_2km WHERE poi_type='1')f
                 WHERE f.start_node =
census_blocks.f_manh_blocks_pop_4326.node_id)

ALTER TABLE census_blocks.f_manh_blocks_pop_4326
ADD column hcent_min_dist_m double precision;
UPDATE census_blocks.f_manh_blocks_pop_4326
set hcent_min_dist_m = (SELECT cost from (SELECT poi_type, start_node, cost
from poi.mom_minimum_cost WHERE poi_type='2')f
                   WHERE f.start_node =
census_blocks.f_manh_blocks_pop_4326.node_id)
```

Annex 4

Python code for rooftop statistics calculation.

```
#-------------------------------------------------------------------------------
----
# Name:        Parallel_Processing_Launcher.py
# Purpose:     Launching parallel processes for each of the codes involved
in the
#              rooftop suitability analysis
#
# Author:      Carlos Javier Delgado
#              Subprocess structure from UPRA GitHub
# Created:     22/10/2019
# Copyright:
# Licence:
#-------------------------------------------------------------------------------
----

import os
import sys
import subprocess
import time
import datetime

def main():

    print("Launching parallel process...")
    initial_t = time.clock()
    print(datetime.datetime.now())

    #Getting path of python shell application
    pydir = sys.exec_prefix
    pyexe = os.path.join(pydir, "python.exe")
    print(pyexe)

    #Defining number of parallel process
    num_parallel_processes = 5

    #Defining the python files that will be executed hen using parallel
processing
    script_1 =
r"D:\Geo_Tech_Master\Thesis_Research\Code\Initial_Rooftop_Statistics_Calcula
tion.py"
    chain_subprocess = ""
    chain_wait = ""

    aCommands = []
    for i in range(num_parallel_processes):
        aCommands.append(r"start python %s %s"%(script_1, i+1))

    print(aCommands)

    #Defining the chain of parameters necessary for subprocess execution
    for j in range(num_parallel_processes):

        if j+1 < num_parallel_processes:
            chain_subprocess += "%s = subprocess.Popen(aCommands[%s],
stdin=None,stdout=subprocess.PIPE,shell=True);"%("ch"+str(j+1), j)
            chain_wait += "astdout, astderr =
%s.communicate();"%("ch"+str(j+1))
```

```
        else:
            chain_subprocess += "%s = subprocess.Popen(aCommands[%s], 
stdin=None,stdout=subprocess.PIPE,shell=True)"%("ch"+str(j+1), j)
            chain_wait += "astdout, astderr = 
%s.communicate()"%("ch"+str(j+1))

    print(chain_subprocess)
    print(chain_wait)

    chain_subprocess = compile(chain_subprocess, '<string>', 'exec')
    exec(chain_subprocess)
    chain_wait = compile(chain_wait, '<string>', 'exec')
    exec(chain_wait)


    print("Ending parallel processing...")
    print(datetime.datetime.now())
    print("Parallel process executed in " + str(((time.clock() - 
initial_t))/60))

if __name__ == '__main__':
    main()


#----------------------------------------------------------------------------
----
# Name:        Initial_Rooftop_Statistics_Calculation.py
# Purpose:     Calculation of the main statistics per roof in the study area
#
# Author:      Carlos Javier Delgado
#
# Created:     14/10/2019
# Copyright:
# Licence:
#----------------------------------------------------------------------------
----

#Importing libraries implemented
import arcpy
import pandas as pd
import os
import datetime
import sys
arcpy.env.overwriteOutput = True

#Main function in the calculation process
def main(iteration_p):

    print("Starting Process %s"%(iteration_p))
    print(datetime.datetime.now())

    #Main paths for saving information
    folderLASfiles = 
r'D:\Geo_Tech_Master\Thesis_Research\Lidar_Data\USGS_NYC2014'
    shpRooftops = 
r'D:\Geo_Tech_Master\Thesis_Research\Processing\Inputs\Suitability_Analysis_
Inputs.gdb\buildings'
    draftGDB = 
r'D:\Geo_Tech_Master\Thesis_Research\Processing\test_lidar_results\draft.gdb
'
    folderLIDARCropped = 
r"D:\Geo_Tech_Master\Thesis_Research\Processing\Lidar_Cropped_%s"%(iteration
_p)
```

```
    folderBK_lasCropped = 
r'D:\Geo_Tech_Master\Thesis_Research\Processing\LAS_Cropped_%s'%(iteration_p
)
    gdbFiltered3dpoints = 
r"D:\Geo_Tech_Master\Thesis_Research\Processing\test_lidar_results\Filtered_
3dpoints_%s.gdb"%(iteration_p)
    gdbSingle3dpoints = r"in_memory"
    gdbLidarVector = r"in_memory"
    gdbOutliers = r"in_memory"
    create_Folder(folderLIDARCropped)
    create_Folder(folderBK_lasCropped)
    create_gdb

    #Function to build the LIDAR as Feature Class only is executed in the 
first iteration
    if iteration_p == 0:
        gridL = buildLIDARgrid(folderLASfiles, draftGDB)
    #gridL = os.path.join(draftGDB, "lidar_grid")

    #Calling the function that build the rooftop indexes and then the 
statistcs
    indexingRooftops(shpRooftops, gridL, draftGDB, folderLASfiles, 
folderLIDARCropped, \
                    folderBK_lasCropped, gdbOutliers, gdbLidarVector, 
gdbSingle3dpoints, gdbFiltered3dpoints, iteration_p)

    print("Ending Process 1")
    print(datetime.datetime.now())

def create_Folder(path_create):
    os.mkdir(path_create)
    print("Folder Created")

def create_gdb(path_complete):
    arcpy.CreateFileGDB_management(os.path.dirname(path_complete), 
os.path.basename(path_complete))
    print("GDB Created")

#Function that build the LIDAR grid for management
def buildLIDARgrid(foldLASfiles, pathGDB):

    listLASfiles = listLIDAR(foldLASfiles)
    print(len(listLASfiles))
    flag1 = 0

    for lf in listLASfiles:
        print(flag1)
        fileLAS = os.path.join(foldLASfiles, lf)
        srLAS = arcpy.Describe(fileLAS).spatialReference

        if flag1 == 0:
            arcpy.CreateFeatureclass_management(pathGDB, "lidar_grid", 
"POLYGON", spatial_reference=srLAS)
            pathLASfc = os.path.join(pathGDB, "lidar_grid")
            arcpy.AddField_management(pathLASfc, "alt_id", "TEXT")


        nameLASfile = os.path.splitext(os.path.basename(fileLAS))[0]

        #Getting the envelope of each lidar file
        desc = arcpy.Describe(fileLAS)
        xmin = desc.extent.XMin
        ymin = desc.extent.YMin
        xmax = desc.extent.XMax
        ymax = desc.extent.YMax
        geoEnvelope = arcpy.Array([arcpy.Point(xmin, ymin),
                         arcpy.Point(xmax, ymin),
                         arcpy.Point(xmax, ymax),
                         arcpy.Point(xmin, ymax)
                         ])

        polygonEnevelope = arcpy.Polygon(geoEnvelope)

        cursor = arcpy.da.InsertCursor(pathLASfc,['alt_id', 'SHAPE@'])
        cursor.insertRow([nameLASfile, polygonEnevelope])
        flag1 += 1

    return pathLASfc

#Function that list the .LAS files
def listLIDAR(pathDirectory):
    return [k for k in os.listdir(pathDirectory) if k.endswith('.las')]

#Core function of rooftop indexing and later call to statistics
def indexingRooftops(roofBuildings, gridF, pathGDB, folderOriLIDAR,
folderCrop_LIDAR, folderbklas, fold_Outliers, \
                   gdbV_lidar, gdb_Single3dP, gdb_Filtered3dP, iteration):

    outputIntergrid = os.path.join(pathGDB, "inter_buildings_grid")
    outputSumTablegrid = os.path.join(pathGDB, "summaryTable_building_grid")


    output_table_verify = os.path.join(r"in_memory",
"table_verify_duplicated_" + iteration)
    output_tabled_duplicates = os.path.join(r"in_memory",
"table_duplicates_" + iteration)
    arcpy.analysis.Intersect([gridF, roofBuildings], outputIntergrid, "ALL",
None, "INPUT")

    #Creating the combined table of the buildings and the corresponding tile
of the grid
    arcpy.analysis.Statistics(outputIntergrid, outputSumTablegrid, "alt_id
COUNT", "FID_buildings;FID_lidar_grid;alt_id")


    arcpy.analysis.Statistics(outputSumTablegrid, output_table_verify,
"FID_buildings COUNT", "FID_buildings")
    arcpy.analysis.TableSelect(output_table_verify,
output_tabled_duplicates, "FREQUENCY > 1")
    #Creating pandas df only for rooftops that share more than one LIDAR
tile
    pdID_Duplicates = fcToPandasDF(output_tabled_duplicates,
["FID_buildings", "FREQUENCY"])

    fl_buildings = "fl_buildings_%s"%(iteration)
    arcpy.management.MakeFeatureLayer(roofBuildings, fl_buildings)
    count = 0
    pdDF_StatisticsperBuilding = ""
    flag_duplicated = 0
    array_paths_duplicated = []
    flp = 0

    #Cursor to iterate each of the rooftops and extract the lidar
information
    with arcpy.da.SearchCursor(outputSumTablegrid, ["FID_buildings",
"FID_lidar_grid", "alt_id", "OBJECTID"]) as cursor:
        print("Processing extraction...")
        for row in cursor:
            print(count)
            print(row[0])
            arcpy.management.SelectLayerByAttribute( fl_buildings,
"NEW_SELECTION", "OBJECTID = " + str(row[0]), None)
            lasds = os.path.join(folderCrop_LIDAR, "C_" + str(row[0]) + "_"
+ str(row[2]) + ".lasd")
            arcpy.ddd.ExtractLas(os.path.join(folderOriLIDAR,
row[2])+".las", folderbklas, "DEFAULT", \
                                fl_buildings, "PROCESS_EXTENT", "_R" +
str(row[0]) , "MAINTAIN_VLR", "REARRANGE_POINTS", \
                                 "NO_COMPUTE_STATS", lasds)

            outlier3dP_building = findOutliers(fold_Outliers, lasds, "O_" +
str(row[0]) + "_" + str(row[2]))
            multipoint_las = lasToVectorPoint(os.path.join(folderbklas,
str(row[2])+ "_R" + str(row[0]) + ".las"), \
                                            gdbV_lidar, "V_" + str(row[0]) +
"_" + str(row[2]))
            single3dp_wo_ouliers = deletingOutliers(multipoint_las,
outlier3dP_building, gdb_Single3dP, \
                                    gdb_Filtered3dP, "F_" + str(row[0]) +
"_" + str(row[2]))


            #Big step
            validation_dup = row[0] in set(pdID_Duplicates.FID_buildings)
            if validation_dup is True:
                frequency =
pdID_Duplicates.loc[pdID_Duplicates['FID_buildings'] == row[0],
'FREQUENCY'].iloc[0]
                flag_duplicated += 1
                if flag_duplicated < frequency:
                    array_paths_duplicated.append(single3dp_wo_ouliers)
                else:
                    print("Analizing duplicates...")
                    array_paths_duplicated.append(single3dp_wo_ouliers)
                    arcpy.management.Merge(array_paths_duplicated,
os.path.join(gdb_Filtered3dP, \
                                            "CM_" + str(row[0]) + "_" +
str(row[2])))
                    flag_duplicated = 0
                    pdDF_Elevation = fcToPandasDF(single3dp_wo_ouliers,
["POINT_Z"])
                    if flp == 0 and count > 0:
                        pdDF_StatisticsperBuilding =
creatingStatistics(pdDF_Elevation, pdDF_StatisticsperBuilding, 0, row[0])
                        flp = 1
                    else:
                        pdDF_StatisticsperBuilding =
creatingStatistics(pdDF_Elevation, pdDF_StatisticsperBuilding, count,
row[0])
                    print("Statistics...")

            else:
                #Statistics are calculated from pandas df to speed up the
processing time
                pdDF_Elevation = fcToPandasDF(single3dp_wo_ouliers,
["POINT_Z"])
                pdDF_StatisticsperBuilding =
creatingStatistics(pdDF_Elevation, pdDF_StatisticsperBuilding, count,
row[0])
                print("Statistics...")


            count += 1
```

```
    print("Exporting to excel...")
    #Statistics are saved as excel file for security
    pdDF_StatisticsperBuilding.to_excel(os.path.join(folderCrop_LIDAR,
"General_Statistics_Per_Building_%s.xlsx"%(iteration)), \
                                        index = True, header=True)

#Function to find outliers and elevation values greater than 1
def findOutliers(fOutliers, lasdataset, nameLASd):
    print("Finding outliers...")
    arcpy.ddd.LocateOutliers(lasdataset, os.path.join(fOutliers, nameLASd),
"APPLY_HARD_LIMIT", 1, 600, \
                            "NO_APPLY_COMPARISON_FILTER", 0, 150, 0.5, 2500)
    return (os.path.join(fOutliers, nameLASd))

#Function to convert the LIDAR points into vector feature class
def lasToVectorPoint(lasC, gdbV, nameLAS_CropVector):
    print("LAS cropped to vector...")
    desc = arcpy.Describe(lasC)
    sr_desc = desc.spatialReference
    arcpy.ddd.LASToMultipoint(lasC, os.path.join(gdbV, nameLAS_CropVector),
0.3, None, "ANY_RETURNS", \
                            "CLASSIFICATION Class", sr_desc, "las", 1,
"NO_RECURSION")
    return (os.path.join(gdbV, nameLAS_CropVector))

#Function to delete outliers from vector feature class and later statistics
calculation
def deletingOutliers(total_3dpoints, outliers3dpoints, gdb_draft_single_las,
gdb_filtered, nameLASdel):
    print("Deleting outliers...")
    arcpy.management.MultipartToSinglepart(total_3dpoints,
os.path.join(gdb_draft_single_las, "S_" + nameLASdel))
    single3dp_layer = "single3dp_layer_1"
    arcpy.management.MakeFeatureLayer(os.path.join(gdb_draft_single_las,
"S_" + nameLASdel), single3dp_layer)
    arcpy.management.SelectLayerByLocation(single3dp_layer, "INTERSECT",
outliers3dpoints, None, "NEW_SELECTION", "INVERT")
    output_deleted = os.path.join(gdb_filtered, nameLASdel)
    arcpy.management.CopyFeatures(single3dp_layer, output_deleted)
    arcpy.management.AddXY(output_deleted)

    return(output_deleted)

#Function that converts Feature Class table to pandas df
def fcToPandasDF(fcobj, aAttributes):
    return (pd.DataFrame( arcpy.da.FeatureClassToNumPyArray(in_table =
fcobj, field_names = aAttributes,  \
            skip_nulls = False, null_value = -99999)))

#Function that calculates main statistics from pandas df
def creatingStatistics(test_pandasdf, output_df_ststs, flag_df,
id_building_m):

    new_df = pd.DataFrame(test_pandasdf['POINT_Z'].describe())
    new_df["id_build"] = id_building_m
    new_df["stats"] = new_df.index
    new_df.set_index("id_build")
    pivot_df = new_df.pivot(index = "id_build", columns = "stats", values =
"POINT_Z")

    if flag_df == 0:
        output_df_ststs = pivot_df
    else:
        output_df_ststs =
output_df_ststs.append(pivot_df.loc[id_building_m], ignore_index=False)
```

```
    return output_df_ststs

#Constructor that recive the id of the process according to parallel
processing python function using subprocess
if __name__ == '__main__':
    num_iteration_process = str(sys.argv[1])
    print(num_iteration_process)
    main(num_iteration_process)
```

Annex 5

Python code for rooftop flatness assessment.

```
#-------------------------------------------------------------------------------
----
# Name:        Parallel_Processing_Launcher_Stage2.py
# Purpose:     Launching parallel processes for each of the codes involved
in the
#              rooftop suitability analysis
#
# Author:      Carlos Javier Delgado
#              Subprocess structure from UPRA GitHub
# Created:     22/10/2019
# Copyright:
# Licence:
#-------------------------------------------------------------------------------
----

import os
import sys
import subprocess
import time
import datetime

def main():

    print("Launching parallel process...")
    initial_t = time.clock()
    print(datetime.datetime.now())

    #Getting path of python shell application
    pydir = sys.exec_prefix
    pyexe = os.path.join(pydir, "python.exe")
    print(pyexe)

    #Defining number of parallel process
    num_parallel_processes = 5

    #Defining the python files that will be executed hen using parallel
processing
    script_1 =
r"D:\Geo_Tech_Master\Thesis_Research\Code\Rooftop_Flatness_Elongation_Assess
ment.py"
    chain_subprocess = ""
    chain_wait = ""

    aCommands = []
    for i in range(num_parallel_processes):
        aCommands.append(r"start python %s %s"%(script_1, i+1))

    print(aCommands)

    #Defining the chain of parameters necessary for subprocess execution
    for j in range(num_parallel_processes):

        if j+1 < num_parallel_processes:
            chain_subprocess += "%s = subprocess.Popen(aCommands[%s],
stdin=None,stdout=subprocess.PIPE,shell=True);"%("ch"+str(j+1), j)
            chain_wait += "astdout, astderr =
%s.communicate();"%("ch"+str(j+1))
```

```
        else:
            chain_subprocess += "%s = subprocess.Popen(aCommands[%s],
stdin=None,stdout=subprocess.PIPE,shell=True)"%("ch"+str(j+1), j)
            chain_wait += "astdout, astderr =
%s.communicate()"%("ch"+str(j+1))

    print(chain_subprocess)
    print(chain_wait)

    chain_subprocess = compile(chain_subprocess, '<string>', 'exec')
    exec(chain_subprocess)
    chain_wait = compile(chain_wait, '<string>', 'exec')
    exec(chain_wait)


    print("Ending parallel processing...")
    print(datetime.datetime.now())
    print("Parallel process executed in " + str(((time.clock() -
initial_t))/60))

if __name__ == '__main__':
    main()


#---------------------------------------------------------------------------
----
# Name:        Rooftop_Flatness_Elongation_Assessment.py
# Purpose:     Estimate the flatness polygons of each rooftop and calculate
\
#              the suitability by considering the compactness/elongation of
#              the polygons.
#
# Author:      Carlos Javier Delgado
#
# Created:     18/10/2019
# Copyright:
# Licence:
#---------------------------------------------------------------------------
----

#Importing main libraries implemented in the algorithm
import arcpy
import pandas as pd
import os
import datetime
import time
import sys
import os
import sys
from os import listdir
from os.path import isfile, isdir, join
import glob
from time import sleep

arcpy.env.overwriteOutput = True
arcpy.env.outputCoordinateSystem = arcpy.SpatialReference(102387)

#Main function to assess the flatness of the rooftops
def main(indicator):

    print("Starting Algorithm...")
    print(str(indicator))
    print(datetime.datetime.now())
    print(arcpy.CheckExtension("3D"))
    print(arcpy.CheckExtension("Spatial"))
    arcpy.CheckOutExtension("3D")
    arcpy.CheckOutExtension("Spatial")
    initial_t = time.clock()

    path_p =
r"D:\Geo_Tech_Master\Thesis_Research\Processing\Results_P1\Union_3\LAS_Cropp
ed_%s"%(indicator)
    result_raw_images =
r"D:\Geo_Tech_Master\Thesis_Research\Processing\Result_raw_lidar_images\raw_
lidar_images_%s"%(indicator)
    vectorized_roof =
r"D:\Geo_Tech_Master\Thesis_Research\Processing\Roof_Vectorized\v_rooftops_%
s.gdb"%(indicator)
    arcpy.env.workspace = vectorized_roof
    export_space =
r"D:\Geo_Tech_Master\Thesis_Research\Processing\Results_P2_Detailed"
    path_folder_image_tif =
r"D:\Geo_Tech_Master\Thesis_Research\Processing\Images_Tiff_Masked\IM10_%s"%
(indicator)
    path_roofs_manh =
r'D:\Geo_Tech_Master\Thesis_Research\Processing\Inputs\Suitability_Analysis_
Inputs.gdb\buildings'
    scratch_gdb =
r"D:\Geo_Tech_Master\Thesis_Research\Processing\Roof_Vectorized\scratch_%s.g
db"%(indicator)

    #Creation of a feature layer
    roof_manh_target = "roof_manh_target"
    arcpy.management.MakeFeatureLayer(path_roofs_manh, roof_manh_target)
    create_Folder(path_p)
    create_Folder(result_raw_images)
    create_Folder(path_folder_image_tif)
    create_gdb(vectorized_roof)
    create_gdb(scratch_gdb)

    #Get all the .las files already cropped
    filez = listLIDARCropped(path_p)
    final_pivot = ""
    flag = 0

    for las in filez:

        try:
            print("Processing roof: " + str(flag))

            jname = las.replace(".las","")
            idroof = (jname.split("_")[2]).replace("R","")

            #Lidar to Raster
            resulting_image = os.path.join(result_raw_images, "I_" + idroof)
            resulting_image_masked = os.path.join(result_raw_images, "M_" +
idroof)
            resulting_poly_roof = os.path.join(vectorized_roof, "V_" +
idroof)
            arcpy.conversion.LasDatasetToRaster(os.path.join(path_p, las),
resulting_image, "ELEVATION", "BINNING AVERAGE LINEAR", "INT", "CELLSIZE",
0.3, 1)

            #Mask the new raster with the rooftop footprint
            arcpy.management.SelectLayerByAttribute(roof_manh_target,
"NEW_SELECTION", "C = " + idroof, None)
            out_raster = arcpy.sa.ExtractByMask(resulting_image,
roof_manh_target)
            out_raster.save(resulting_image_masked)
            #Raster to polygon
            arcpy.conversion.RasterToPolygon(resulting_image_masked,
resulting_poly_roof, "NO_SIMPLIFY", "VALUE", "SINGLE_OUTER_PART", None)
            pdDFRoof = fcToPandasDF(resulting_poly_roof,
["Id","gridcode","Shape_Area", "Shape_Length"])
            sumTotalArea = pdDFRoof["Shape_Area"].sum()
            #Initial compactness calculation
            pdsorted["Percentage"] = (pdsorted["Shape_Area"] /
sumTotalArea)*100
            pdsorted["Compactness"] = (4*np.pi)*(pdsorted["Shape_Area"] /
np.power(pdsorted["Shape_Length"],2))


            #Creating the structure of the resulting table
            if pdsorted.shape[0] < 5:

                aOrder = []
                aPercen = []
                aElevation_Avg = []
                aCompact = []
                aShape_F = []
                aShape_A = []

                for i in range(1,pdsorted.shape[0]+1):
                    aOrder.append("A%s"%(i))
                    aPercen.append("P%s"%(i))
                    aElevation_Avg.append("E%s"%(i))
                    aCompact.append("Compact_%s"%(i))
                    aShape_F.append("Shape_%s"%(i))
                    aShape_A.append("Per_Efec_%s"%(i))
                pdsorted["Order"] = aOrder
                pdsorted["Percen"] = aPercen
                pdsorted["Elevation_Avg"] = aElevation_Avg
                pdsorted["Compact"] = aCompact
                pdsorted["Shape_Factor"] = aShape_F
                pdsorted["Shape_Area_Efect"] = aShape_A

            else:

                pdsorted["Order"] = ["A1","A2","A3","A4","A5"]
                pdsorted["Percen"] = ["P1","P2","P3","P4","P5"]
                pdsorted["Elevation_Avg"] = ["E1","E2","E3","E4","E5"]
                pdsorted["Compact"] = ["Compact_1", "Compact_2",
"Compact_3", "Compact_4", "Compact_5"]
                pdsorted["Shape_Factor"] = ["Shape_1", "Shape_2", "Shape_3",
"Shape_4", "Shape_5"]
                pdsorted["Shape_Area_Efect"] = ["Per_Efec_1", "Per_Efec_2",
"Per_Efec_3", "Per_Efec_4", "Per_Efec_5"]


            resulting_poly_roof_layer = "resulting_poly_roof_layer"
            arcpy.management.MakeFeatureLayer(resulting_poly_roof,
resulting_poly_roof_layer)

            aShapeRelations = []
            aShapePercentageBbox = []

            #Asssesing the shape by generating the envelope of each polygon
            for row in range(0, pdsorted.shape[0]):
                id_value = pdsorted.iloc[row]["Id"]
                area_value = pdsorted.iloc[row]["Shape_Area"]

arcpy.management.SelectLayerByAttribute(resulting_poly_roof_layer,
"NEW_SELECTION", "Id = " + str(id_value), None)
                out_bbox = r"in_memory\minbbx_%s_%s"%(idroof, row)
                out_lines_bbox = r"in_memory\oLines_%s_%s"%(idroof, row)
                out_split_lines_bbox =
os.path.join(scratch_gdb,"splitLines_%s_%s"%(idroof, row))

arcpy.management.MinimumBoundingGeometry(resulting_poly_roof_layer,
out_bbox, "RECTANGLE_BY_WIDTH", "NONE", None, "NO_MBG_FIELDS")
                arcpy.management.PolygonToLine(out_bbox, out_lines_bbox,
"IGNORE_NEIGHBORS")
                arcpy.management.SplitLine(out_lines_bbox,
out_split_lines_bbox)
                pdDF_bbox_split = fcToPandasDF (out_split_lines_bbox,
["OBJECTID","Shape_Length"])

                length_bbox = pdDF_bbox_split.iloc[0]["Shape_Length"]
                width_bbox = pdDF_bbox_split.iloc[1]["Shape_Length"]
                #Considerations  of Elongation factor formula given by
Harris
                if length_bbox >= width_bbox:
                    shape_relation = width_bbox/length_bbox
                else:
                    shape_relation = length_bbox/width_bbox
                area_bbox_m = length_bbox * width_bbox
                shape_percentage = (area_value/area_bbox_m)*100

                aShapeRelations.append(shape_relation)
                aShapePercentageBbox.append(shape_percentage)
                del(out_bbox)
                del(out_lines_bbox)

            pdsorted["id_building"] = idroof
            pdsorted.set_index("id_building")
            pdsorted["Shape_Relation"] = aShapeRelations
            pdsorted["Shape_Per_Area_Bbox"] = aShapePercentageBbox

            #Organizing pandas df
            if pdsorted.shape[0] < 5:
                for i in range(pdsorted.shape[0],5):
                    print(i)
                    pdsorted.loc[i] = [9999 + i, 0, 0, 0, 0, 0, "A%s"%(i+1),
"P%s"%(i+1), "E%s"%(i+1), "Compact_%s"%(i+1), "Shape_%s"%(i+1),
"Per_Efec_%s"%(i+1), 0, 0, 0]
                pdsorted["id_building"] = idroof
                pdsorted.set_index("id_building")

            pivot_df = pdsorted.pivot(index = "id_building", columns =
"Order", values = "Shape_Area")
            pivot_df2 = pdsorted.pivot(index = "id_building", columns =
"Percen", values = "Percentage")
            pivot_df3 = pdsorted.pivot(index = "id_building", columns =
"Elevation_Avg", values = "gridcode")
            pivot_df4 = pdsorted.pivot(index = "id_building", columns =
"Compact", values = "Compactness")
            pivot_df5 = pdsorted.pivot(index = "id_building", columns =
"Shape_Factor", values = "Shape_Relation")
            pivot_df6 = pdsorted.pivot(index = "id_building", columns =
"Shape_Area_Efect", values = "Shape_Per_Area_Bbox")


            if flag == 0:
                final_pivot =
pd.concat([pivot_df,pivot_df2,pivot_df3,pivot_df4,pivot_df5, pivot_df6],
axis=1)
                flag += 1
            else:
                momen_pivot = 
pd.concat([pivot_df,pivot_df2,pivot_df3,pivot_df4,pivot_df5, pivot_df6], 
axis=1)
                final_pivot = pd.concat([final_pivot, momen_pivot])
                flag += 1

            #Option to create the images as RGB by triplicating the bands
            #resultimg_composite = os.path.join(path_folder_image_tif, "C_" 
+ idroof + ".tif")

#arcpy.management.CompositeBands([resulting_image_masked,resulting_image_mas
ked,resulting_image_masked], resultimg_composite)

        except Exception as err:
            print("Erro {}".format(err))
            print("Error when processing roof no " + str(flag))
            sleep(90)
            #Saving temporarily the resulting table in case of an error
            final_pivot.to_excel(os.path.join(export_space, 
"Results_Temporary_%s.xlsx"%(indicator)), index = True, header=True)

    #Saving the results to an Excel file for categorization of suitabilities
    final_pivot.to_excel(os.path.join(export_space, 
"Results_P2_Iteration_%s.xlsx"%(indicator)), index = True, header=True)

    print("Ending process...")
    print(datetime.datetime.now())
    print("Processing time:  " + str(((time.clock() - initial_t))/60))

#Fucntion to get all the LAS cropped files
def listLIDARCropped(mainFolder):

    aFiles = []
    os.chdir(mainFolder)
    for file in glob.glob("*.las"):
        aFiles.append(file)

    return(aFiles)

#Function that converts Feature Class table to pandas df
def fcToPandasDF(fcobj, aAttributes):
    return (pd.DataFrame( arcpy.da.FeatureClassToNumPyArray(in_table = 
fcobj, field_names = aAttributes,  skip_nulls = False, null_value = -
99999)))

def create_Folder(path_create):
    os.mkdir(path_create)
    print("Folder Created")

def create_gdb(path_complete):
    arcpy.CreateFileGDB_management(os.path.dirname(path_complete), 
os.path.basename(path_complete))
    print("GDB Created")


#Constructor that recive the id of the process according to parallel 
processing python function using subprocess
if __name__ == '__main__':
    num_iteration_process = str(sys.argv[1])
    print(num_iteration_process)
    main(num_iteration_process)
```

Masters
Program
in **Geospatial**
Technologies

Supported by:
Education and Culture
ERASMUS MUNDUS