Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 163 (2016) 175 – 185

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

Representing urban geometries for unstructured mesh generation

Abel Gargallo-Peiró*, Arnau Folch, Xevi Roca

Dpt. Computer Applications in Science and Engineering, Barcelona Supercomputing Center, C/ Jordi Girona 29, 08034 Barcelona, Spain

---

Abstract

We present a robust and automatic method to generate an idealized surface geometry of a city landscape ready to be meshed for computer simulations. The city geometry is idealized for non viscous flow simulations and targets two main geometrical features: the topography and the city blocks. The procedure is fully automatic and demands no human interaction given the following source data: the city cadastre, a Digital Elevation Model (DEM) of all the target domain, and Light Detection And Ranging (LiDAR) data of the domain region covered by the cadastre. The geometry representation takes three main steps. First, a 2D mesh of the cadastre is generated, where the elements are marked according to street and block regions. Second, using a DEM of the city landscape the topography surface mesh is generated by finding the best surface mesh in the least-squares sense obtained by deforming the previous 2D mesh. Third, we extrude the block facades and we compute a planar ceiling taking into account all the buildings belonging to that city block. We describe the applicability of the geometry representation by presenting the work-flow required to generate an unstructured mesh valid for non-viscous flow or transport simulations. Finally, we illustrate the main application by obtaining a surface and tetrahedral mesh for the city of Barcelona in Spain.

Peer-review under responsibility of the organizing committee of IMR 25

Keywords: Surface mesh generation; Unstructured mesh generation; Mesh for simulation; Urban area mesh; Microscale CFD

---

1. Introduction

Cross-scale atmospheric modeling is essential to quantify and mitigate a number of growing environmental issues in urban areas, including extreme weather events, air quality or dispersal of toxic substances by accidental release [1]. During the last decade, the increase in computational resources has allowed to couple mesoscale numerical weather prediction models [2–4], having typical grid resolutions of kilometric size, with microscale Computational Fluid Dynamics (CFD) codes, solving for wind over complex terrains and/or at building scales [1,5,6]. This model downscaling strategy is challenging by a number of reasons, including inconsistencies in different model's physics, numerical strategies and space-time resolutions. In addition, urban-scale wind and atmospheric dispersal simulations require of appropriate geometric descriptions of the city area.

Computational meshes for urban flow simulations require an idealized geometrical description to reach a balance between simplicity and realism. The geometric model must preserve the main features influencing the flow, such as

---

* Corresponding author. Tel.: +34-93-413-7648
E-mail address: abel.gargallo@bsc.es

Peer-review under responsibility of the organizing committee of IMR 25
doi:10.1016/j.proeng.2016.11.044

streets and terrain, while removing smaller features that can not be physically modeled. Specifically, this geometry idealization has to address the following issues:

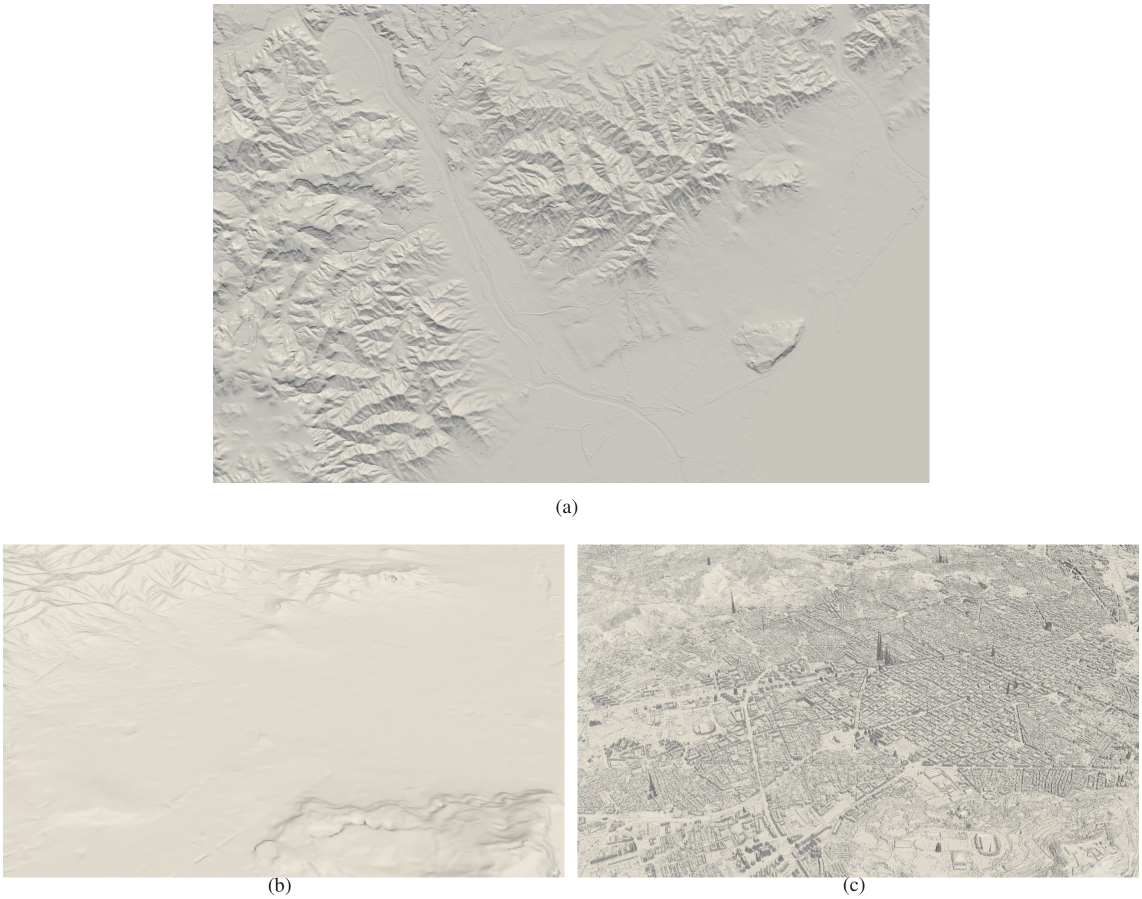
- **Detail removal**. Surface geometry of a city can be obtained through Light Detection And Ranging (LiDAR) campaigns, that provide high-resolution Digital Surface Models (DSMs). However, raw data contains any geometrical feature present at the moment of the data extraction such as cars, trees and other elements. These details are not required for the simulation and have to be ignored by the computational mesh.
- **Streets and buildings**. The fluid flow will be determined by the distribution of the city streets, determined by the surrounding buildings. City cadastre provides this information through a 2D description of the building contours and urban components however, the topographic information is not available.
- **Topography**. The slope of the city underlying terrain influences the flow results and therefore, has to be captured by the simulation geometry. Standard Digital Elevation Models (DEMs) represent the bare ground surface topography however, no information of the buildings and other urban equipments is available.
- **Automation**. It is desirable to automate all the meshing process in a user-friendly manner in order to reduce tedious, time-consuming and error-prone human intervention.
- **Element count**. For computational purposes, the element count poses an additional constrain because the urban surface geometry is the input of a 3D mesher and the resulting volume mesh can easily multiply by more than two orders of magnitudes the number of elements. This aspect is particularly critical in daily operational frameworks, such as air quality forecasts, where a balance is needed between the number of elements and the computational resources available in order to deliver simulation results timely.
- **Surface marks**. Features such as urban surface roughness [7,8] or pollutant emission inventories may need to be properly marked in the geometry representation. In this manner, the mesh can inherit the geometry marks for the corresponding flow or transport model.

In the literature of computer vision, we can find many approaches to represent a 3D model of a city [9–17]. These approaches, model the cities starting from high-resolution, sub-metric size, LiDAR data. However, meshing requirements in computer vision are different to those in simulation. As opposed to computer vision, where the main meshing target is the visual realism of the mesh, computational meshes requires of well defined meshes around buildings which must be watertight and composed by well-shaped elements.

Many previous studies have already confronted with the issue of urban flow simulations and, consequently, with the previous generation of meshes for computational purposes. From the meshing point of view a clear division exists. On the one hand, several works [18–21] define the urban geometry by means of a CAD surface or an STL that has required dedicated manual design and cleaning. In this case, the geometry definition holds less idealization and it is closer to reality but, in contrast, the computational domain is normally constrained to a small portion of a city. On the other hand, other works (*e.g.* urban canopy models) may cover larger city areas but assume planar underlaying terrain and idealized street canyon building distributions [22]. In contrast with both approaches, we target at an automatic definition of geometry without manual design and cleaning and including topography and an idealization of the city as a set of blocks, each with a height resulting from averaging its individual buildings.

The main contribution of this works is to present a method to represent the surface geometry of urban areas taking into account all the previous requirements. The main application is to generate unstructured 3D computational meshes for non-viscous flow simulation of urban areas. To this purpose, our novel methodology combines information from a cadastre, a DEM and 3D LiDAR data point clouds and align them to obtain an idealized geometry for flow/transport simulation purposes. In particular, we use a DEM to build a city topography where no low-scale urban noise is present. Next, we use a simple block description given by the cadastre to build a clean definition of the block contours. Finally, we use LiDAR data strictly to provide the building/block actual height, avoiding the non-representative information (noise) existing in the LiDAR data, particularly at street level. In this way, the procedure can be automated and does not require of any manual cleaning or filtering in any of the three data sources. In this paper, this methodology is applied to mesh the whole city of Barcelona at a block level. However, it could be straightforwardly exported to any metropolitan area for which these sources of data are available.

The remainder of the paper is organized as follows. Section 2 presents the scope of this work, states the problem and the solution steps. Section 3 describes the approach to generate the surface model of a given city landscape.



(a)

(b) (c)

Figure 1. (a) DEM of the neighboring area around Barcelona. Zoom of the (b) DEM and (c) LiDAR data in the same area.

Finally, in Section 4 we present the main highlights of the tetrahedral mesh generation, and in Section 5 we analyze the application to the particular case of Barcelona city.

2. Problem statement and methodology

2.1. Sources of data

The methodology presented in this work requires of three different sources of data. First, a DEM file of the city representing the bare ground and consisting of a regular grid with elevations at each node, a cloud point, or a triangular surface mesh. This DEM is used to obtain a final clean mesh of the ground that features no urban morphologies and is the starting surface where city blocks will be extruded. Figures 1(a) and 1(b) show the DEM for the case of Barcelona city. Second, one requires a cadastre of the target city (or metropolitan area) or any equivalent two dimensional Computer-Aided Design (CAD) geometry (see Figure 2). Different levels of detail (blocks, single buildings, gardens, urban elements, etc.) can be switched on/off through the CAD layers. Here we focus on the city at a block level only, *i.e.* we only retain the blocks of the target city and remove the rest of layers. The set of closed curves defining each block is used to obtain clean building facades and roofs, without undesired features. In particular, for the case of Barcelona, it consists of 4834 blocks for a domain of 11×15 km^2 (the underlying DEM extends 30×20 km^2 so that a buffer zone exists having topographic features only). Third, LiDAR data of the city (Figure 1(c)) will be used to retrieve a heigh-resolution map of the buildings/blocks automatically and without need for cleaning or filtering.

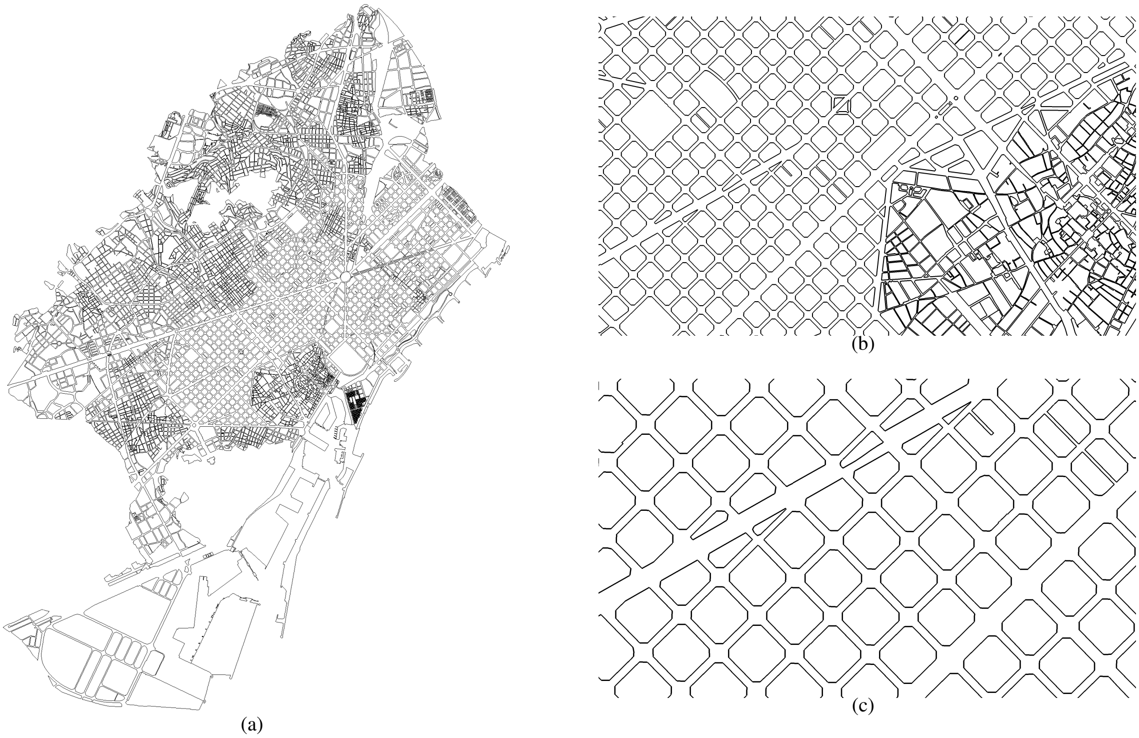

(a) (b) (c)

Figure 2. Cadastre of Barcelona city: (a) complete cadastre, (b) sample of Eixample and Gotic neighborhoods, and (c) sample of Eixample neighborhood.

2.2. Methodology

The procedure to define a 3D geometry of a city at block level and to generate a tetrahedral mesh is composed of 4 steps, illustrated in Figure 3:

1. **Generate a mesh of the cadastre.** In Section 3.1, we generate a 1D mesh of the cadastre with elements of the desired edge length for the ground, and afterwards a planar triangle mesh that is constrained to the previously generated edge mesh. Figure 3(a) shows the 2D mesh obtained for the 13 selected cadastre blocks, where the color of the elements differentiates ground and block id.
2. **Generate a surface mesh of the topography.** In Section 3.2, the surface mesh is derived from the previous 2D mesh that better approximates topography in a least-squares sense. Figure 3(b) shows the obtained surface mesh which, in this region, has a small topographic gradient of 25 meters only. Note that the surface mesh is still conformal with the blocks defined by the cadastre description.
3. **Generate the city block geometry.** In Section 3.3, we first duplicate the boundary nodes of the blocks and construct a triangle mesh that topologically connects the already existent ground nodes with the new roof ones, generating the block facades. Next, we compute the roof heights using the LiDAR data. In particular, we compute the averaged height of a block integrating the LiDAR height data in the region belonging to a block. Figure 3(c) shows the resulting city block surface obtained for the selected test case.
4. **Generate a tetrahedral volume mesh.** In Section 4, using the defined geometry as an STL representation and closing it with a box of the desired height, one can use any well established tetrahedral mesh generation code to obtain a tetrahedral mesh (*e.g.* TetGen [23] or NetGen[24]). Note that we require to prescribe the desired element size in the ground and buildings, and in the top ceiling and boundary walls. Moreover, due to the

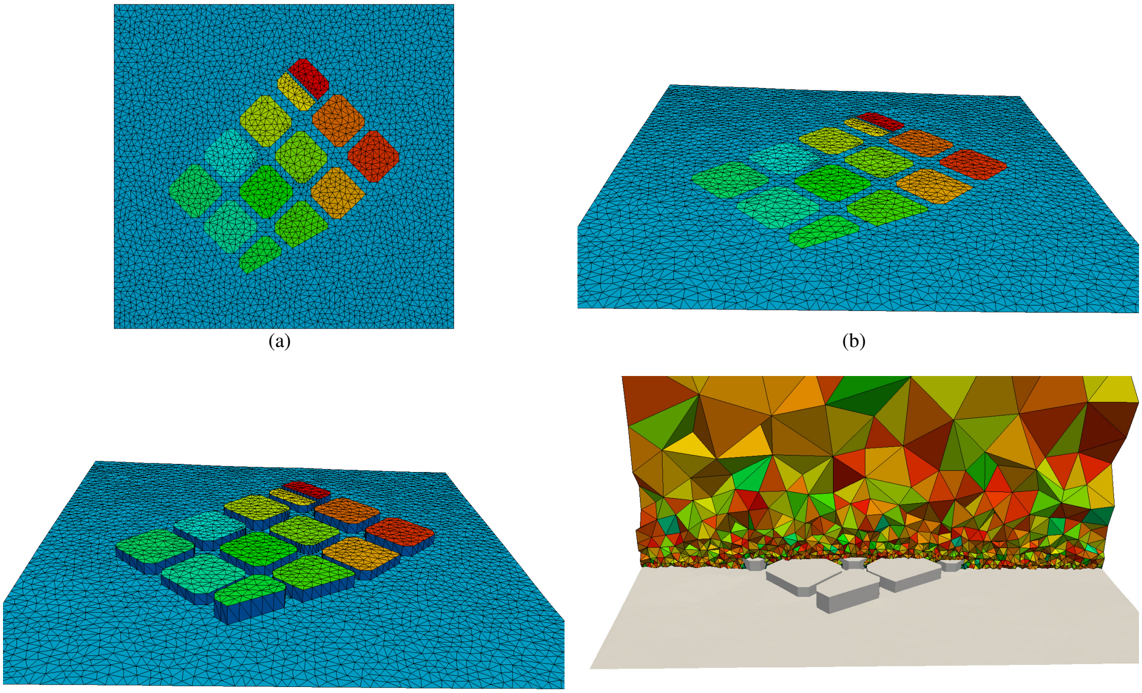

(a) (b) (c) (d)

Figure 3. Main steps of the geometry definition and mesh generation process for a sample of 13 blocks at Barcelona: (a) 2D cadastre mesh, (b) topography surface mesh conformal with the cadastre blokcs (colored by ground or block id), (c) geometry definition with topography and blocks (colored by ground or block id), and (d) final tetrahedral mesh (colored with respect to the quality of the elements: red represents good quality, and dark blue bad quality).

complexity of the geometry, we will use quality measures to assess the validity of the obtained meshes and optimize them [25,26]. The obtained mesh for the studied test case is illustrated in Figure 3(d).

3. Geometrical representation of a city: blocks, heights, and topography

The main objective of this work is to obtain an idealized model of a city landscape that is valid for meshing purposes and has no human intervention in the process. To do so we focus on reproducing two main geometrical features. First, we want to include the topography in our idealized surface. Although there are many geometrical features in a city, one of the most relevant ones is the topography. Even if there are no mountains around the city, a small slope through several kilometers influences and determines significantly the behavior of the air flow. Moreover, in cities like Barcelona, our test example, we find several hills and two mountain ranges in the target domain that is mandatory to include in the geometry. Particularly in our region of interest around Barcelona, we cover a region from the sea level to peaks and crests of a height that reach 600 meters.

Second, among all the urban features present in the city, we are strictly interested in the city at the block level. To constrain strictly to the city blocks we make to main simplifications. On the one hand, we want to avoid any other urban equipment rather than the buildings. In the resulting idealized model we do not want to include fire hydrants, traffic lights, trees, antennas... since a proper discretization of urban features that are smaller than buildings would increase significantly the number of elements of the final mesh, preventing the overall procedure to be actually operative. In addition, those components area geometrically negligible with respect to the size of the blocks.

On the other hand, we have to choose between representing each single building or the city blocks. Herein, we qualitatively assume that at the city (or metropolitan area) scale the influence of the feature of each single building is negligible with respect to the influence of the blocks themselves in the overall domain. This simplification allows obtaining a clean geometry of the whole city with a small element count in the surface mesh.

In future works we will study the quantitative difference between the building and block approaches. However, the generated meshes are to be used in an operative procedure to obtain daily data about the weather and pollution forecast in the city. Therefore, the geometrical accuracy has to be balanced with a reduced final element count so that the overall procedure becomes actually operative.

The generation of the geometry is divided in three main steps: meshing the cadastre (2D), generating the topography mesh (surface mesh without city blocks), and generating the city blocks.

3.1. Cadastre mesh

As highlighted in Section 2.2, the first step towards generating an idealized city modeling is to obtain the cadastre (floor plan) of the city. Using the cadastre will provide a clean mesh representation of the city blocks, avoiding the noise present in the LiDAR data, which features not only the buildings but at the same time any other urban element present at the moment of the data extraction. For instance, a fire hydrant featuring in the LiDAR data would require a too fine mesh in order to be captured properly, but it would not provide actual information for the wind assessment of the city, wasting therefore computational resources.

Hence, provided the cadastre of the target region that is to be simulated, we will generate a 1D mesh composed by edges of the desired mesh size. Next, we generate a planar triangle mesh which conforms the 1D mesh. In the 2D mesh, taking into account that the 1D mesh defines closed regions, we classify each element marking to which block does it belong or if it belongs to the ground. To generate the triangle mesh we use the Delaunay-based mesher Triangle [27].

3.2. Topography surface mesh

The second step of the geometry definition is to define a the topography surface mesh. To do so we compute the least squares approximation of the Digital Elevation model with the generated triangle mesh of the cadastre.

Let $h(\mathbf{x})$ be a continuous field (DEM) that we want to approximate. In particular, we want to compute a nodal linear approximation:

$$z(\mathbf{x}) = \sum_{i=1}^{n_N} z_i\, \phi_i(\mathbf{x}),$$

where ϕ_i is the shape function corresponding to node i and z_i is the height of node i.

Instead of performing an interpolative approach, we want to minimize the error between our approximation and the exact data. In particular, we want the coefficients z_i, $i = 1, \ldots, n_N$ that give the best approximation of the DEM data in the least squares sense. That is,

$$\{z_1, \ldots, z_{n_N}\} = \underset{z_i}{\operatorname{argmin}} \|z - h\|^2 = \underset{z_i}{\operatorname{argmin}} < z - h\ ,\ z - h >,$$

where the scalar product is defined as

$$< f, g >= \int_\Omega f(\mathbf{x}) \cdot g(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

being Ω the DEM domain covered by the mesh. The corresponding normal equations are:

$$\sum_{j=1}^{n_N} < \phi_i,\ \phi_j >\ z_j =< \phi_i,\ h > \quad \forall i = 1, \ldots, n_N,$$

from which we can compute the height of the nodes in a straight-forward manner solving the linear system with a SDP matrix.

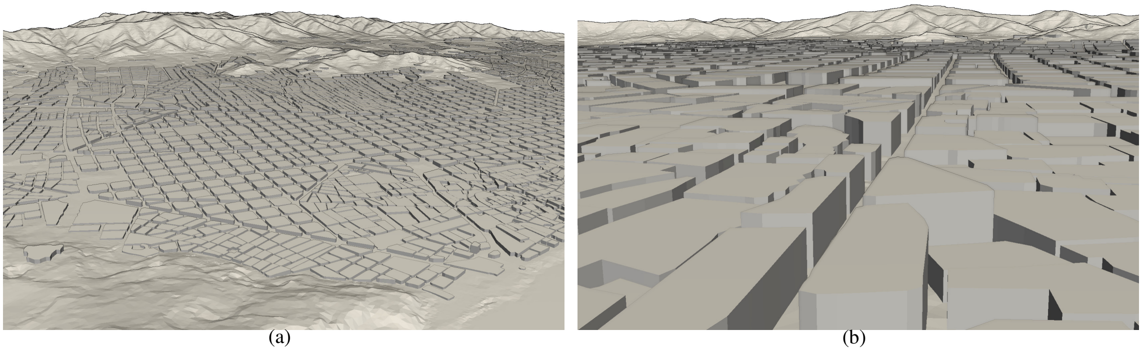

(a) (b)

Figure 4. Barcelona surface geometry.

Recall that we could also compute the best aproximation taking into account the (x_i, y_i, z_i) node components. However, we want to keep the conformity of the surface with the cadastre in order to extrude the blocks in the next step. Therefore, we fix the x_i and y_i components of the nodes and strictly find the z_i component that gives the best approximation of the topography, with one degree of freedom for each mesh node.

3.3. Block surface mesh

Given the topography mesh of a city landscape, we generate the block mesh in two steps. First, we set the topology of the facades. It is important to highlight that the topography mesh is conformal with the block facades from the cadastre by construction. During the process, the planar cadastre mesh is converted to a surface mesh by elevating the nodes adding a third coordinate component and therefore, ensuring that the cadastre is conformal with the topography. Furthermore, we have a mark for each element that sets its block id, or its belonging to the ground. Therefore, we get all the boundary edges of a given block id and we duplicate the nodes belonging to those edges. Next, we generate for each new edge two triangles that connect the new nodes on the roof with the already existent ground nodes. Those new triangles compose the facades of the blocks.

Following, we compute the height of the roof of the blocks integrating the LiDAR data on the block region. That is, the height H_b of a block b is

$$H_b = \frac{1}{A_b} \sum_{E_e \in R_b} \int_{E_e} l(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{1}$$

where E_e denotes the e-nth element, R_b denotes the set of elements that compose the block b, A_b the area of the block (sum of the area of its elements) and $l(\mathbf{x})$ denotes the height field provided by the LiDAR data.

Note that in the cadastre one can have regions where buildings are registered but have not been build yet or have been demolished (that is, data from a cadastre and a LiDAR campaign will very likely be not synchronous). Therefore, in the block generation procedure we filter cadastre regions that are not actual buildings. To this end, we use both the LiDAR data and the DEM comparing the block height obtained with both surface representations. If the block height is equal in both cases up to an input tolerance, we will remove this region and consider it ground. Note that this filtering is also necessary for data in the cadastre that does not match the desired block idealization, such as roundabouts, lakes or parks. The default tolerance is 3 meters, which is the standard value for a one floor building.

The resulting mesh obtained for Barcelona city is illustrated in Figure 4. The obtained geometry is composed by 4834 city blocks and the triangle surface mesh that defines the geometry is composed by around 2 million elements, with an element size between one and 15 meters.

4. Application to tetrahedral mesh generation

Given the surface mesh of the city landscape geometry obtained in Section 3, we next close the geometry defining a box of the desired height. The resulting closed mesh can be given as input to any well-established and robust

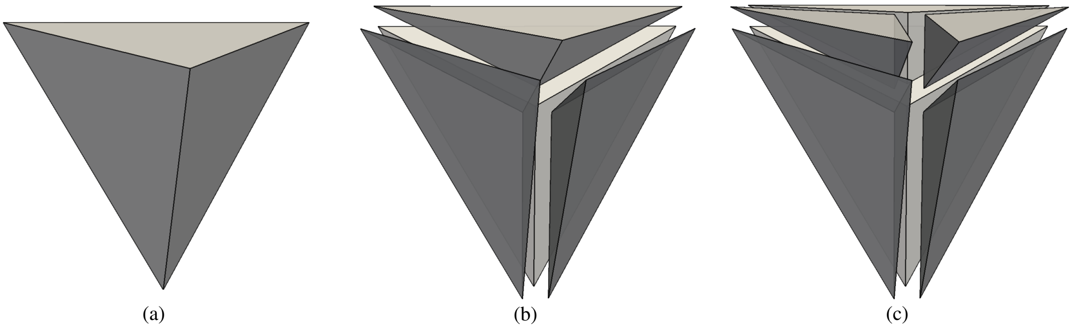


Figure 5. (a) Original tetrahedron with an upper face that it is not on the boundary but has all the nodes on the boundary. (b) Barycentric split of the element. (c) Face split to ensure at least one degree of freedom on the face.

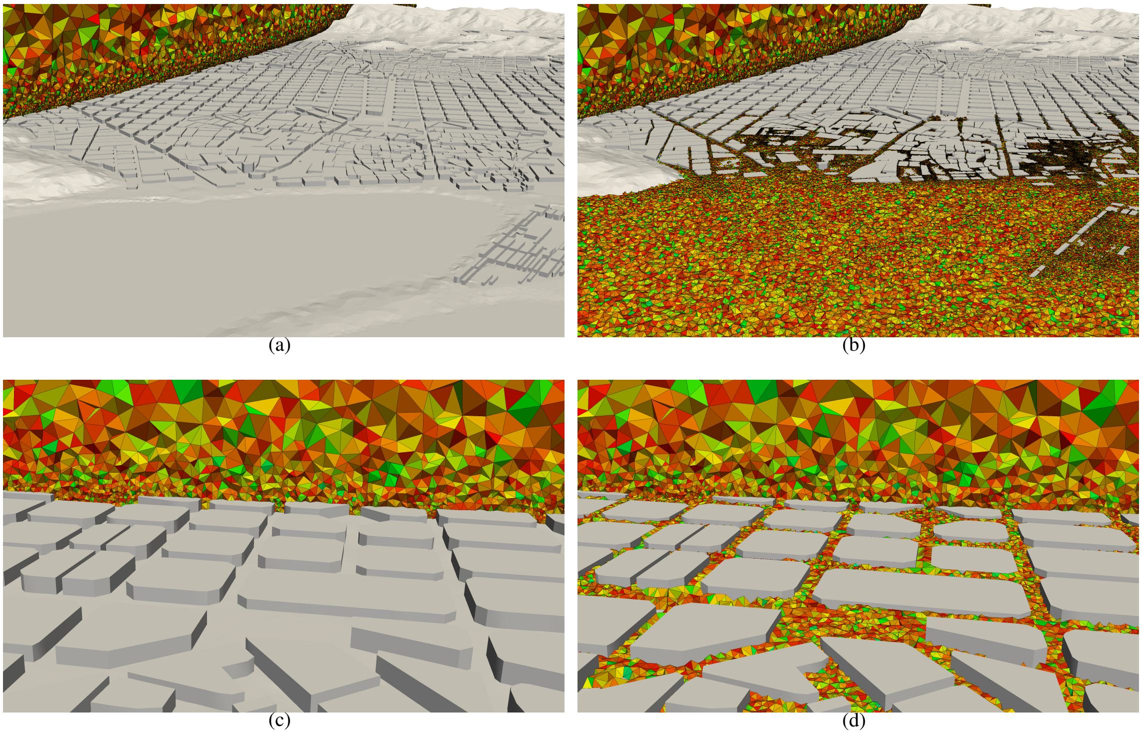
tetrahedral mesher. In particular we have used TetGen [23] to obtain a Delaunay-based mesh with the desired element size. We consider isotropic tetrahedra and, in order to control the element size, we prescribe the surface mesh size fine on the ground and coarser on the ceiling. The ground edge length is in the range of 1 to 15 meters (featuring elements on highly geometrically constrained areas of edges between 0.1 and 1 meters) whereas the ceiling length is of the order of 100 meters. Moreover, the element size on the ground grows up to 50 meters away from the target city. The resulting mesh has to be post-processed in order to avoid elements with all nodes on the boundary. This is necessary in oder to prevent tetrahedra where all the nodes are on the boundary, something possible in case of sharp geometric features. To avoid this issue, we have implemented two templates to ensure that no elements or non-boundary faces have all the nodes on the boundary. In particular, we first subdivide all the elements that have either all the nodes on the boundary, or a non-boundary face will all the nodes on the boundary. As illustrated in Figure 5(b) we compute a barycentric subdivision of the elements into four new tetrahedra. Next, we get all the resulting tetrahedra with a non-boundary face with all nodes on the boundary and we split that face into three triangles, and each of the two adjacent elements into three new tetrahedra, see Figure 5(c).
Another possible issue in this kind of geometry is the existence of alleys with a width of 1-2 meters that would be composed by strictly one element width. These cases can be automatically resolved by the tetrahedral mesher by imposing that the tetrahedral elements are n times finer than the surface ones, where n is an input parameter ($n = 2$ by default). Once the mesh is generated, we will compute the new approximation of the DEM file delivered by the finer surface mesh. To end with, we highlight that the generated tetrahedral mesh is constrained to a complex geometry, which in some regions may derive in low-quality elements. To assess the validity of the mesh we will take into consideration the shape quality measure [25], a Jacobian-based measure that is suitable for smoothing and untangling [26,28], and that quantifies the quality of the element in terms of a target ideal tetrahedraon. Since we are dealing with isotropic Delaunay-based meshes, we consider as ideal the equilateral tetrahedron. In particular, we will use the optimization framework provided in the work [26], which allows smoothing tetrahedral meshes with respect to the desired ideal element, and features untangling if inverted elements are present.

5. Results

Figure 6 illutrates the final tetrahedral mesh obtained for Barcelona after splitting the boundary elements and optimizing the mesh quality. We stress that this geometry has been obtained automatically merging the cadastre, DEM and LiDAR data, without human intervention, which demonstrates the robustness of the presented approach.

We would like to highlight that the mesh generation procedure requires two input sizing values: the edge length h_s on the surface (topography and buildings) and the edge length h_t at the top ceiling. Provided these two values the mesh sizing of the procedure is automatically determined in the following hierarchical process:

- 1D cadastre meshing: the maximum edge length is determined by h_s, and it may feature smaller edges due to geometric constrains (distance between the vertices of the geometry).



(a) (b)

(c) (d)

Figure 6. Mesh at block level of Barcelona city. The elements are colored with respect to their quality.

- 2D surface meshing: the maximum edge length is determined by h_s. Some elements may feature smaller size due to the resolution of the previously generated 1D mesh and to ensure good-quality elements. In particular, the mesher [27] allows ensuring that no triangles have angles lower than a desired value (herein, defaulted at 10 degrees).
- 3D volume meshing: the maximum edge length at the ground level and the top ceiling are determined by h_s and h_t, respectively. In addition, the mesh may feature smaller element lengths according to the geometric features of the surface mesh and to enforce the quality of the volume mesh. In particular, two quality critera are applied to generate the volume mesh [23]. First, a maximum allowable radius-edge ratio (defaulted at 2), and second a minimum allowable dihedral angle (herein, defaulted also at 10 degrees).

In the test example for Barcelona city illustrated in Figure 6 the selected mesh sizes are $h_s = 15$ and $h_t = 100$. The resulting surface mesh is composed by 2M elements and 1K nodes. The computational time to generate it on a MacBook Pro (with one dual-core Intel Core i7 CPU, a clock frequency of 3.0 GHz, and a total memory of 16 GBytes) is of 113 seconds, from which 19s correspond to the 2D mesh generation of the cadastre, 14s correspond to generating the block facades, and 80s correspond to computing the surface mesh.

The obtained volume mesh is composed by 50M tetrahedral elements and 10M nodes. The computational time to generate the volume mesh is 104 minutes, from which 93 minutes correspond to the mesh generation procedure and 11 minutes to the optimization of the mesh. In this example, the mesh sizing, and the geometry and quality constrains of the hierarchical process lead to a mesh with minimum edge length of 0.1 meters. We would like to highlight that the optimization framework is able to enhance the quality of the tetrahedral mesh and improve the quality of its elements, stepping form a minimum shape quality of 0.03 to 0.06.

6. Concluding remarks

We have presented an automatic method to generate idealized city geometries ready for meshing for computational purposes. The presented method requires three sources of data that are standardly available in any city: the cadastre, a DEM (which contains the underlaying topography) and LiDAR data of the city. The procedure assumes that there are two main geometrical features to be included in the idealized geometry: the topography and the blocks. This level of idealization allows obtaining a technique which presents several features. First, the geometry definition is robust and automatic, does not require manual intervention and is not influenced by the noise present in the data. Second, it presents a low element count of the geometry mesh, taking into account that we are dealing with domains of the order of a hundred square kilometers and thousands of blocks. Moreover, we are taking into account in the geometry definition that generated mesh will be used to daily compute several times a weather and pollution forecast in the city. Therefore, a balance is needed between the number of elements and the computational resources available in order to deliver simulation results timely.

Finally, we would like to highlight that this is an ongoing work where two main features have to be studied in the near future. First, a study between the improvement derived from going to the block level to the building level and the resulting increase in the computational cost. Second, we would like to automatically detect singular buildings, such as churches, where a different idealization could be used, analyzing again the gain in precision of the derived simulation versus the increase in the computational cost.

7. Acknowledgements

This work has been developed in the context of the EU H2020 GrowSmarter project which has Barcelona as one of the three lighthouse cities. The work of the last author was supported by the European Commission through the Marie Sklodowska-Curie Actions (HiPerMeGaFlows project).

References

[1] F. Chen, H. Kusaka, R. Bornstein, J. Ching, C. S. B. Grimmond, S. Grossman-Clarke, T. Loridan, K. W. Manning, A. Martilli, S. Miao, D. Sailor, F. P. Salamanca, H. Taha, M. Tewari, X. Wang, A. A. Wyszogrodzki, C. Zhang, The integrated wrf/urban modelling system: development, evaluation, and applications to urban environmental problems, International Journal of Climatology 31 (2011) 273–288.

[2] S. S. Chen, W. Zhao, M. A. Donelan, J. F. Price, E. J. Walsh, The CBLAST-hurricane program and the next-generation fully coupled atmosphere-wave-ocean models for hurricane research and prediction, Bulletin of the American Meteorological Society 88 (2007) 311–317.

[3] C.-H. Moeng, J. Dudhia, J. Klemp, P. Sullivan, Examining two-way grid nesting for large eddy simulation of the pbl using the wrf model, Monthly Weather Review 135 (2007) 2295–2311.

[4] C. Talbot, E. Bou-Zeid, J. Smith, Nested mesoscale large-eddy simulations with wrf: Performance in real test cases, Journal of Hydrometeorology 13 (2012) 1421–1441.

[5] C. Draxl, M. Churchfield, J. Mirocha, S. Lee, J. Lundquist, J. Michalakes, P. Moriarty, A. Purkayastha, M. Sprague, B. Vanderwende, Coupling a mesoscale numerical weather prediction model with large-eddy simulation for realistic wind plant aerodynamics simulations (poster), in: Symposium on Boundary Layers and Turbulence, 2014, p. 1.

[6] F. Castro, C. Silva Santos, J. Lopes da Costa, One-way mesoscale-microscale coupling for the simulation of atmospheric flows over complex terrain, Wind Energy 18 (2015) 1251–1272.

[7] B. Fisher, J. Kukkonen, M. Piringer, M. W. Rotach, M. Schatzmann, Meteorology applied to urban air pollution problems: concepts from cost 715, Atmospheric Chemistry and Physics 6 (2006) 555–564.

[8] J. Millward-Hopkins, A. Tomlin, L. Ma, D. Ingham, M. Pourkashanian, Mapping the wind resource over {UK} cities, Renewable Energy 55 (2013) 202 – 211.

[9] F. Lafarge, C. Mallet, Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation, International Journal of Computer Vision 99 (2012) 69–85.

[10] Y. He, Automated 3D building modelling from airborne LiDAR data, Ph.D. thesis, The University of Melbourne, 2015.

[11] F. Rottensteiner, C. Briese, Automatic generation of building models from lidar data and the integration of aerial images, in: In: H.-G. Maas, G. Vosselman and A. Streilein (eds), Proc. ISPRS working group III/3 workshop on 3-D reconstruction from airborne laserscanner and InSAR data, Dresden, October, IAPRS Vol. XXXIV, Part 3/W13, 2003, pp. 174–180.

[12] V. Verma, R. Kumar, S. Hsu, 3d building detection and modeling from aerial lidar data, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, 2006, pp. 2213–2220. doi:10.1109/CVPR.2006.12.

[13] F. Tarsha-Kurdi, T. Landes, P. Grussenmeyer, Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data, in: ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, volume XXXVI, Espoo, Finland, 2007, pp. 407–412. URL: `https://halshs.archives-ouvertes.fr/halshs-00264843`.

[14] A. Huerta, A. Angeloski, X. Roca, J. Peraire, Efficiency of high-order elements for continuous and discontinuous Galerkin methods, Int. J. Numer. Meth. Eng. 96 (2013) 529–560.

[15] Q.-Y. Zhou, U. Neumann, A streaming framework for seamless building reconstruction from large-scale aerial lidar data, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 2759–2766.

[16] Q.-Y. Zhou, U. Neumann, Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 115–128.

[17] Q.-Y. Zhou, U. Neumann, 2.5d building modeling with topology control, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 2489–2496.

[18] M. J. Leach, S. T. Chan, J. K. Lundquist, High-Resolution CFD Simulation of Airflow and Tracer Dispersion in New York City, 2005, pp. 1–6. URL: `http://www.osti.gov/scitech/servlets/purl/884757`.

[19] S. R. Hanna, M. J. Brown, F. E. Camelli, S. T. Chan, W. J. Coirier, S. Kim, O. R. Hansen, A. H. Huber, R. M. Reynolds, Detailed simulations of atmospheric flow and dispersion in downtown manhattan: An application of five computational fluid dynamics models, Bulletin of the American Meteorological Society 87 (2006) 1713–1726.

[20] A. Huber, P. Georgopoulos, R. Gilliam, G. Stenchikov, S. Wang, B. Kelly, H. Feingersh, Modeling air pollution from the collapse of the world trade center and potential impact on human exposures, Environmental Mmanagement (2004) 35–40.

[21] B. Blocken, J. Carmeliet, Pedestrian wind environment around buildings: Literature review and practical examples, Journal of Thermal Envelope and Building Science 28 (2004) 107–159.

[22] C.-H. Chang, R. N. Meroney, Concentration and flow distributions in urban street canyons: wind tunnel and computational data, Journal of Wind Engineering and Industrial Aerodynamics 91 (2003) 1141 – 1154.

[23] H. Si, Tetgen, a delaunay-based quality tetrahedral mesh generator, ACM Trans. Math. Softw. 41 (2015) 11:1–11:36.

[24] J. Schöberl, Netgen an advancing front 2d/3d-mesh generator based on abstract rules, Computing and Visualization in Science 1 (1997) 41–52.

[25] P. M. Knupp, Algebraic mesh quality metrics, SIAM J. Numer. Anal. 23 (2001) 193–218.

[26] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes, Int. J. Numer. Meth. Eng. 103 (2015) 342–363.

[27] J. R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in: M. C. Lin, D. Manocha (Eds.), Applied Computational Geometry: Towards Geometric Engineering, volume 1148 of *Lecture Notes in Computer Science*, Springer-Verlag, 1996, pp. 203–222. From the First ACM Workshop on Applied Computational Geometry.

[28] A. Gargallo-Peiró, X. Roca, J. Sarrate, A surface mesh smoothing and untangling method independent of the CAD parameterization, Comput. Mech. 53 (2014) 587–609.