

Geometry-aware estimation of photovoltaic energy from aerial LiDAR point clouds

Chiara Romanengo^{a,1}, Tommaso Sorgente^{a,1}, Daniela Cabiddu^{a,*}, Matteo Ghellere^b, Lorenzo Belussi^b, Ludovico Danza^b, Michela Mortara^a

^aIstituto di Matematica Applicata e Tecnologie Informatiche “Enrico Magenes”, Consiglio Nazionale delle Ricerche, Via de Marini 6, Genova, 16149, Italy

^bIstituto per le Tecnologie della Costruzione, Consiglio Nazionale delle Ricerche, Via Lombardia 49, San Giuliano Milanese (MI), 20098, Italy

Abstract

Aerial LiDAR (and photogrammetric) surveys are becoming a common practice in land and urban management, and aerial point clouds (or the reconstructed surfaces) are increasingly used as digital representations of natural and built structures for the monitoring and simulation of urban processes or the generation of what-if scenarios. The geometric analysis of a “digital twin” of the built environment can contribute to provide quantitative evidence to support urban policies like planning of interventions and incentives for the transition to renewable energy. In this work, we present a geometry-based approach to efficiently and accurately estimate the photovoltaic (PV) energy produced by urban roofs. The method combines a primitive fitting technique for detecting and characterizing building roof components from aerial LiDAR data with an optimization strategy to determine the maximum number and optimal placement of PV modules on each roof surface. The energy production of the PV system on each building over a specified time period (e.g., one year) is estimated based on the solar radiation received by each PV module and the shadow projected by neighboring buildings or trees and efficiency requirements. The strength of the proposed approach is its ability to combine computational techniques, domain expertise, and heterogeneous data into a logical and automated workflow, whose effectiveness is evaluated and tested on a large-scale, real-world urban areas with complex morphology in Italy.

Keywords: Primitive Fitting, Primitive Recognition, Semantic Segmentation, Photovoltaics, Urban Digital Twins, Computational Geometry

1. Introduction

Renewable energy, particularly solar energy, plays a crucial role in the construction of more resilient cities. By harnessing the Sun’s energy, cities can reduce their dependence on fossil fuels, lower carbon emissions, and mitigate the impacts of climate change. In addition to the benefit for individual citizens, a computational tool that can quantitatively and reliably estimate the photovoltaic energy production of rooftops would be immensely beneficial to city administrations, allowing more effective planning and deployment of solar technologies, optimizing energy use, and promoting the transition to a sustainable behaviour through incentives grounded on well-founded estimates. This is particularly true in the panorama of increasing efforts in the development of urban digital twins, where a similar service can play a relevant role in supporting urban planning and management.

Estimating the energy production of a photovoltaic element is a challenging process. Indeed, the solar radiation hitting a roof surface depends on many factors, including seasonal variations in radiation intensity, weather conditions, geographic location, roof orientation, area and slope, and shadow cast by

surrounding structures, such as other buildings and trees. Previous attempts have relied only partially on the geometric aspect of the problem, providing averaged results of limited usefulness. Moreover, when georeferenced data are investigated, geographic information systems are typically used, which, on the one hand, facilitate the management of spatial data through visualization, documentation and spatial query functionalities, but, on the other hand, rely on 2D representations. As far as urban characterization is concerned, the full 3D geometry is crucial to provide an as accurate as possible estimate of roof features and shadow casting.

Our research is grounded in the design and development of *urban digital twins*, a shared research effort requiring multidisciplinary competences [1] and based on a representation of the facets of the city based on *layers*. Each layer stores knowledge, data, models and algorithms related to a specific field, such as environment, mobility, security. In this research, we exploit the geometric layer coupled with the built environment layer, if present. Specifically, we rely on point clouds from aerial LiDAR to reconstruct, characterize, and annotate the urban morphology and to measure on real data the salient dimensions involved in the estimate of PV energy production. The built environment layer contains information on the energy-related characteristics of buildings, such as construction period, intended use, and other attributes, as well as details on buildings subject to architectural or landscape constraints.

The main contribution of this paper is a comprehensive pipeline that integrates geometric processing, shadow analysis, and energy modeling to support the large-scale deployment of photovoltaic (PV) systems in urban environments. This pipeline combines a series of modular tools and procedures to:

*Corresponding author

Email addresses: chiara.romanengo@cnr.it (Chiara Romanengo), tommaso.sorgente@cnr.it (Tommaso Sorgente), daniela.cabiddu@cnr.it (Daniela Cabiddu), matteo.ghellere@cnr.it (Matteo Ghellere), lorenzo.belussi@cnr.it (Lorenzo Belussi), ludovico.danza@cnr.it (Ludovico Danza), michela.mortara@cnr.it (Michela Mortara)

¹These authors contributed equally to this work

(i) automatically extract roof geometries from aerial LiDAR point clouds; (ii) detect and characterize roof *pitches* (flat surface components) using a recognition approach based on the Hough Transform (HT); and (iii) optimize the placement of PV modules to maximize potential deployment. The computation of the solar radiation on each module is then combined with a geometry-based shadow analysis performed on a 3D surface mesh of the city, constructed from a resampled and cleaned up version of the LiDAR point cloud, to estimate the effective solar exposure time of each module. The geometric results are further coupled with energetic and logistic constraints to aggregate hourly simulations into seasonal or annual energy estimates and to filter out low-efficiency configurations. The strength of the proposed approach lies in its ability to integrate heterogeneous data, domain knowledge, and computational techniques into a coherent and automated workflow.

The paper is organized as follows. In Section 2, we review previous work related to the estimation of rooftop energy production and the detection of roofs from LiDAR point clouds. In Section 3, we summarize the theoretical concepts underlying our method. Then, in Section 4, we present our approach for estimating photovoltaic energy production, which consists of two main phases. The first phase (Section 5) involves a primitive fitting technique to detect and characterize roof pitches from aerial LiDAR point clouds. The second phase (Section 6) focuses on determining the maximum number and optimal placement of PV modules on each roof surface, and on estimating their energy production over a given time period (e.g., one year). In Section 7, we report experimental results conducted in large-scale urban environments with complex morphology, in the Italian cities of Genova and Matera. Finally, in Section 8, we conclude the paper.

2. Related Work

Estimating PV energy production relies on the geometric approximation of urban environments. Various methodologies use airborne LiDAR point clouds, digital surface models (DSMs), digital elevation models (DEMs), and 3D city models to assess solar exposure and energy potential. These approaches typically consider factors like roof shape, orientation, self-shading, and occlusions from nearby structures.

Accurately identifying roof geometries can improve the accuracy of energy yield predictions. The segmentation and analysis of roof structures from airborne LiDAR data is an active area of research in geometry processing and has significant implications for PV energy estimation. However, most existing methods prioritize roof detection as a step toward full building reconstruction, rather than focusing on detailed segmentation and analysis of roof shapes.

The following subsections review previous geometry-based methods for estimating PV energy production (Section 2.1) and present an overview of existing techniques for extracting roof structures from LiDAR data (Section 2.2).

2.1. Estimating of PV energy production

In the scientific literature, LiDAR data has long been used to reconstruct urban geometries and estimate PV production. Early work by Kodysh et al. [2] combined LiDAR-derived DEMs with building footprints from GIS software, but their

assumption that footprints match roof outlines led to inaccuracies, particularly for complex or irregular buildings. More recent methods generate high-resolution DSMs directly from LiDAR point clouds [3, 4, 5]. Achbab et al. [4] derive DSMs photogrammetrically, using roof contours and masking to extract slope and orientation, then integrate multiple raster maps to estimate usable roof areas. Similarly, Memme et al. [5] combine DSM and LiDAR data within a QGIS workflow to evaluate tilt and orientation for solar suitability.

With the increasing availability of satellite imagery and OpenStreetMap data, data-driven approaches have gained attention. Lee et al. [6] introduced DeepRoof, a convolutional neural network (CNN)-based method that extracts roof geometries from aerial images, relying on external data for slope and height, which may lead to geometric inconsistencies. Singh and Banerjee [7] proposed a hybrid approach that combines macro-scale land-use data with micro-scale geospatial analyses of satellite imagery. While practical, it relies on statistical extrapolations, which may introduce uncertainties.

To improve computational efficiency, some methods simplify urban geometry. Cheng et al. [8], for example, represent buildings as box-like volumes with flat roofs and vertical facades. While efficient, this oversimplification can reduce accuracy, particularly for complex roof structures. In contrast, Li et al. [9] and Manni et al. [10] employ 3D high level-of-detail (LoD) building models, ranging from LoD1 to LoD3, using SketchUp, based on LiDAR point clouds. These methodology ease PV production estimation for sloped roofs, but they only consider self-shadowing and neglect the influence of neighboring structures. Moreover, they rely on manual modeling, limiting scalability for large urban areas.

It is also worth mentioning that several initiatives have also contributed to PV energy estimation. The Photovoltaic Geographical Information System (PVGIS) by the European Commission provides global solar radiation and system performance data [11]. It estimates performance based on latitude, longitude, slope, and azimuth, with shadows partially considered via a horizon height database. However, 3D geometry is not explicitly modeled, and the low resolution excludes small objects unless users upload their own horizon data.

In contrast, Google’s Project Sunroof [12] provides personalized roof analysis by combining data from Google Maps and other databases, also considering 3D roof models, shadows from nearby structures and trees, and historical weather patterns like cloud cover and temperature. However, 3D modeling and shade calculations use Google’s machine learning algorithms over images and they report that the outcomes are not always precisely accurate. The service is currently available only in the US.

To summarize, existing methodologies reflect a trade-off between accuracy, scalability, and data requirements. LiDAR-based techniques yield high-resolution geometry but are computationally demanding and often require extensive preprocessing. Photogrammetric methods offer a cost-effective alternative, though their accuracy depends on the quality of input datasets. Deep learning approaches benefit from abundant imagery but depend on large training datasets and external information sources, which may introduce inconsistencies. Simplified 3D modeling strategies enhance computational efficiency but often lack the geometric detail necessary for precise solar assessments. In contrast, methods employing LoD models offer greater accuracy but rely on manual processes, limiting their

scalability for large urban environments.

2.2. Roofs from LiDAR Point Clouds

In the last decade, several methodological frameworks have been developed for the extraction and segmentation of roof geometries from LiDAR point clouds. These approaches can be categorized into four main groups: global energy optimization, feature clustering, region growing and fitting primitives.

Global energy optimization methods treat segmentation as an optimization problem, where a global energy function incorporates geometric and spatial constraints [13, 14]. Notable examples include the works of Gu et al. [15] and Wang et al. [16], which introduce specialized energy minimization strategies for segmenting roof planes from LiDAR data. Despite these methods generally produce globally consistent segmentations with high accuracy, they often involve the minimization of complex non-linear functions, making them computationally expensive.

Feature clustering techniques segment point clouds by grouping points based on similarity in local geometric descriptors, such as surface normal or curvature. Common clustering algorithms include k-means [17] and DBSCAN [18], which are widely used due to their simplicity and computational efficiency. For instance, Kong et al. [19] employ a k-plane clustering approach that operates directly on LiDAR points, avoiding the need to compute surface normals. Approaches based on feature clustering generally remain sensitive to noise and heavily dependent on parameters such as neighborhood size and the number of clusters, which can affect their generalization.

Region growing methods segment point clouds by iteratively expanding regions from seed points or patches, adding neighboring points that meet specific criteria such as planarity, proximity, or orientation similarity [20, 21, 22]. These methods are effective for identifying continuous surfaces but struggle with selecting robust seed points and accurately defining boundaries between regions. To overcome these issues, Li et al. [23] propose a hybrid approach that combines octree-based planar patch extraction, hierarchical clustering for seed generation, and a relabeling strategy to refine boundaries. This method improves segmentation accuracy and boundary quality, outperforming existing techniques on real airborne LiDAR datasets.

Model fitting-based approaches can estimate robust primitive parameters even in the presence of high noise and outliers [24]. These include both stochastic methods, such as RANSAC (RANdom SAmple Consensus) [25], and parameter space techniques based on Hough-like voting and parameter space clustering [26]. Recently, a combination of the RANSAC methodology and the Hough transform is exploited by [27], where a recognition method able to segment the input point cloud into geometric primitives of different types (e.g., planes, cylinders, spheres) is proposed. Such a method is robust to issues of previous approaches and showed a performance superior to [23].

Existing segmentation techniques for identifying roof planes in urban LiDAR data are typically designed as preprocessing steps for general building reconstruction or urban modeling. These methods focus on geometric decomposition, delineating planar or primitive shapes without accounting for functional roof attributes relevant to energy analyses. As a result, they often overlook critical features, such as subtle roof inclinations, orientation continuity, and surface connectivity, that are essential for accurate energy assessment.

3. Background

We introduce here few basic theoretical notions that will be used throughout the paper.

3.1. Recognition technique based on the Hough transform

We consider each roof as composed of multiple flat components, called pitches, with different orientations. We adopt a recognition strategy based on the Hough transform (HT) to recognize the planes that compose the roof of a building and to characterize the roof pitches. Being the recognition based on a voting procedure, it is robust to noise, missing parts, and outliers [28], resulting particularly useful in this context as the acquired point clouds typically suffer from these kind of imperfections. The use of the HT was originally limited to the recognition of straight lines in images [29]; then it has been extended to a wide range of algebraic primitives in [30] and, recently, generalized to surface primitives in [31].

Given a family of surfaces $\mathcal{F} = \{S_a\} \subset \mathbb{R}^3$ depending on a set of parameters $a \in \mathbb{R}^n$ and a set of points $\mathcal{P} \subset \mathbb{R}^3$ to recognize, the problem is to find a set of parameters $\bar{a} = (a_1, \dots, a_n)$ such that the surface $S_{\bar{a}}$ best fits the points \mathcal{P} . To do this, the HT exploits the duality between \mathcal{P} and the parameters space \mathbb{R}^n : a point x in \mathcal{P} corresponds to a locus $\Gamma_x(\mathcal{F})$ in \mathbb{R}^n , and the variation of x on the surface S_a generates a set of loci $\{\Gamma_x\}$. The intersection of these $\{\Gamma_x\}$ contains the parameters $\bar{a} \in U$ that generate the best fitting surface $S_{\bar{a}}$. The duality concept is fundamental for the HT-based recognition algorithm, since it translates the recognition problem into detecting which values of the parameters that determine the family \mathcal{F} correspond to the surface that best fits the set of points \mathcal{P} . The common strategy to identify the solution (or a solution) is based on the so-called *accumulator function*; it consists in a voting system whereby each point in a point cloud votes for a n -tuple $a = (a_1, \dots, a_n)$ and the most voted n -tuple is selected as the most representative surface for the set of points \mathcal{P} .

3.2. Inset and offset of a polygon

Given two polygons $A, B \in \mathbb{R}^2$, we define their *Minkowski sum* and *difference* as the polygons

$$A \oplus B = \bigcup_{a \in A} B_a, \quad A \ominus B = \{A^c \oplus -B\}^c,$$

where B_a denotes the set B translated by a position vector starting from o_B the origin of B (the common initial point of all its position vectors) and ending at the point a , A^c is the complement of A , and $-B = \{2o_B - b : b \in B\}$ denotes the symmetric reflection of B w.r.t. o_B [32]. In other words, $A \oplus B$ is obtained by translating B so that its origin o_B coincides with each boundary point of A , and taking the boundary of the union of all resulting points.

Minkowski sums are used in many applications, such as motion planning and computer-aided design and manufacturing. In our context, we exploit them to compute the *Inset* and the *Offset* of a polygon A (representing a pitch of a building's roof) by a radius r , defined as the sets of points inside and outside A , respectively, at distance r from the boundary. Indeed, these sets can be expressed as the Minkowski sum and difference of A and a ball B_r of radius r . However, these sets are not linear polygons, being composed of line segments and circular arcs in

correspondence of the corners of A . To get linear polygons, we approximate B_r with an n -sided regular polygon B_r^n of radius r :

$$\begin{aligned}\mathcal{I}(A) &:= \{a \in A \mid \text{dist}(a, A) = r\} = A \ominus B_r \simeq A \ominus B_r^n, \\ \mathcal{O}(A) &:= \{a \in A^c \mid \text{dist}(a, A) = r\} = A \oplus B_r \simeq A \oplus B_r^n.\end{aligned}$$

We point out that $\mathcal{I}(A)$ may be empty if r is too large, and may be disconnected when A is non-convex and has some narrow components. Conversely, $\mathcal{O}(A)$ cannot be empty nor disconnected. We compute them using the *2D Minkowski Sums* [33] package of the CGAL library [34]. The higher n , the more accurate is the approximation of $\mathcal{I}(A)$ and $\mathcal{O}(A)$. In our tests, we used $n = 12$.

4. Overview of the method

Given a LiDAR point cloud relative to a large urban area and a set of 2D polygons representing building footprints, our method estimates the photovoltaic energy production of each building going through two main steps, as shown in Figure 1. In the first phase, we partition the point cloud building-wise (Section 5.1), and identify roof points (Section 5.2). Each roof is then segmented into planes representing the roof pitches via a HT-based primitive fitting algorithm, also determining the orientation and inclination of each pitch. Finally, we extract the bounding polygon of each pitch (Section 5.3). Primitive fitting allows also to reduce noise from the input LiDAR point cloud and to recognize trees as well, which we can subsequently represent according to a specific season, with the scope of creating a more realistic mesh of the city for the shadow computation (Section 5.4).

In the second phase, we apply an optimization strategy to determine the maximum number and optimal placement of photovoltaic modules on each pitch (Section 6.1), and estimate the electricity production of each building over a given time range (Section 6.2). This estimate is based on the hourly radiation of the Sun, also considering the shadow projected by neighboring buildings or trees, and efficiency requirements of PV systems.

The code relative to Section 5 is written in MATLAB, while the code relative to Section 6 is in C++ using the library cino-lib [35]. Both libraries will be publicly available on GitHub after acceptance.

5. Point cloud segmentation, characterization and refinement

In the following, we detail the analysis process that applies to the input LiDAR point cloud of the urban environment. It is divided in two main subparts, one for roof extraction via primitive fitting and roof pitches characterization, while the other focused on the point cloud refinement and mesh generation. The process is preceded by a building-wise partitioning to optimize and parallelize computation. Our approach takes in input an aerial LiDAR point cloud representing an urban context, the polygons representing the building footprints, and a “vegetation map” of tree positions within the selected area. These 2D information are provided, with no loss of generality, in the form of standard ESRI Shapefiles [36]; they are often available through online repositories, such as OpenStreetMap [37], and many municipalities do already offer this kind of information as OpenData. The parameters used in the steps of our pipeline

described in this section are automatically determined based on geometric properties of the input point cloud, such as the number of points, the density, or the minimum bounding box. For this reason, in the following we will only mention the a-priori fixed parameters.

5.1. Building-wise partitioning

To split the input point cloud into single buildings, we utilize the 2D building footprints provided in the input shapefile, represented as a set of polygons. Each 3D point is orthogonally projected onto the plane of the building footprints, and a point-in-polygon test is performed independently for each polygon to determine whether the point lies within it. If the point is inside a polygon, it is then associated with the building whose footprint is represented by that polygon (see Figure 2).

The point-in-polygon test is performed using a ray-casting algorithm, based on the Jordan Curve Theorem [38]. According to this theorem, a point x lies inside a polygon S if a semi-infinite ray cast from x intersects the polygon boundary an odd number of times. We extended the classic algorithm by W. Randolph Franklin [39] to support complex polygonal geometries, including those with holes, since building footprints often include internal courtyards. To evaluate whether x belongs to S , represented as an ordered list of vertices, the orthogonal projection \hat{x} of the point onto the polygon plane is first checked against the axis-aligned bounding box of S . If the projection falls outside the bounding box, the point is excluded. Otherwise, a ray is cast from x in an arbitrary direction (typically, horizontally to the right), and intersections with the polygon edges are counted. An even (or zero) number of intersections implies that the point lies outside S ; an odd number confirms that it lies inside.

Since we are dealing with multiple disjoint polygons representing building footprints, the point-in-polygon test is applied independently to each polygon, using parallel computation to enhance scalability. We note that this test could be further improved using alternative techniques, such as winding number estimation [40].

5.2. Roof extraction

Each sub-cloud representing a building is first translated to the origin to avoid numerical issues due to the geographical coordinates. Next, we apply a segmentation approach based on region growing [41] to obtain an initial segmentation into planes, as shown in Fig. 4(a). This requires each point to be associated with its normal vector, computed from its 3D coordinates as in [42]. Points are required to be ordered along XY-aligned scanlines, typical of airborne LiDAR data, to estimate the line of sight for orienting normals; otherwise, they are directly estimated from point positions. When available, LAS attributes like scan angle and scanline ID can help improve this estimation.

For the segmentation, we set a minimum segment size as a very small percentage (0.25%) of the total number of points in the original point cloud, even at the cost of producing an oversegmentation. This choice helps preserve detail, as point cloud density is typically non-uniform. We then apply the strategy in [31] to maximally aggregate segments belonging to the same plane, based on the parameters obtained by the Hough

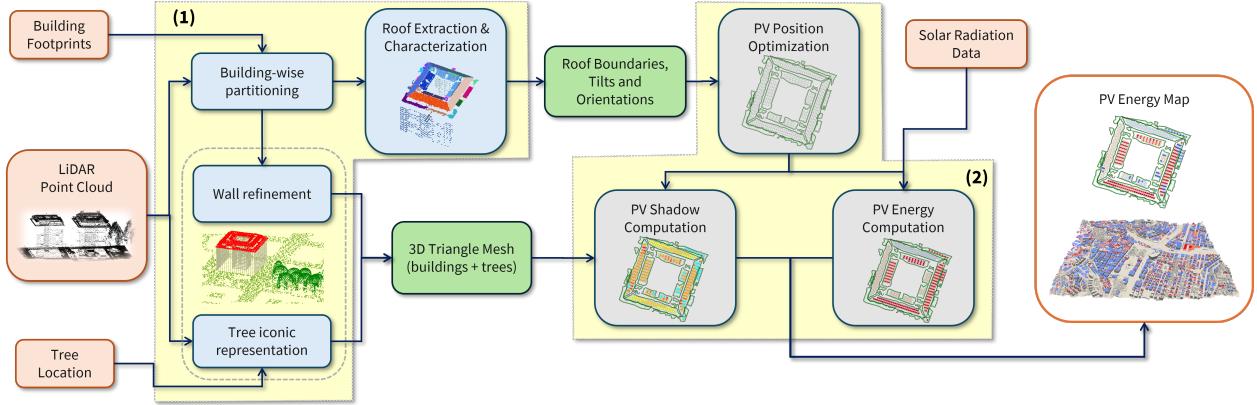


Figure 1: Overview of the proposed pipeline, consisting of two main phases: (1) roof characterization, including building-wise point cloud partitioning, roof point extraction, HT-based roof pitch segmentation, and seasonal tree modeling; and (2) PV energy estimation, involving module placement optimization and energy production computation based on solar radiation and shading effects.

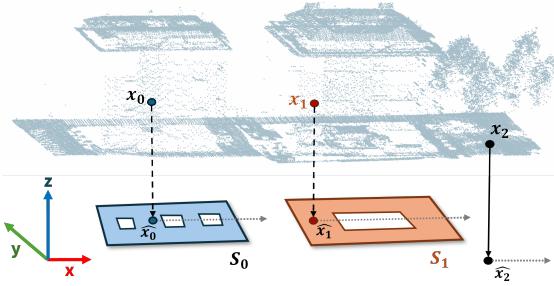


Figure 2: Building-wise partitioning of the LiDAR point cloud using a point-in-polygon test. Each 3D point is orthogonally projected onto the plane of the 2D building footprints. The test is performed independently for each footprint, using a ray-casting algorithm to determine whether the point lies within the footprint polygon.

Transform. Specifically, we use complete linkage, a hierarchical clustering method that assigns each segment to its own cluster and iteratively merges the closest clusters, to build a dendrogram according to the following map:

$$D(C_h, C_j) := \max_{\alpha_k \in C_h, \alpha_l \in C_j} d(\alpha_k, \alpha_l),$$

where (C_h, C_j) is a given pair of clusters and d is a measure of distance. In our case, the distance is $d(\alpha_1, \alpha_2) = \|\mathbf{n}_1 \times \mathbf{n}_2\|_2 + \min_{\mathbf{p}_1 \in \alpha_1, \mathbf{p}_2 \in \alpha_2} \|\mathbf{p}_1 - \mathbf{p}_2\|_2$, where \mathbf{n}_1 and \mathbf{n}_2 are the normals and \mathbf{p}_1 and \mathbf{p}_2 are points. Note that, if $d(\alpha_1, \alpha_2) = 0$ (or less than a threshold, in our implementation), then the primitives α_1 and α_2 are equal with respect to the selected criterion. The result is a collection of subsets of points $\{\mathcal{P}\}$ belonging to the same plane. Each subset \mathcal{P} is processed with the HT-based recognition technique (Section 3.1) to obtain the parameters of the best fitting plane. More in details, we use the approach in [43] that exploits the mathematical expression of planes in the Hesse normal representation:

$$x \cos \phi \sin \theta + y \sin \phi \sin \theta + z \cos \theta - \rho = 0,$$

in which the parameters are the polar coordinates of the normal vector to the plane $\phi \in [0, 2\pi]$, $\theta \in [0, \pi]$ and the distance $\rho \geq 0$ from the plane to the origin of the coordinate system, see Fig. 3. The work in [43] proposes an optimization strategy that

builds upon the technique in [31] by discretizing the parameter space in a parameter-driven manner. This approach reduces computational cost and makes the Hough Transform applicable to large-scale datasets, such as urban contexts.

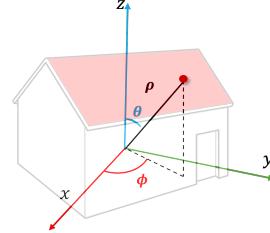


Figure 3: Hesse normal parametrization of a plane.

To classify a plane as a roof pitch, we exploit the normal vector to the plane $\mathbf{n} = (\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta)$ automatically obtained by the parameters θ and ϕ computed by the HT and its position in the 3D space. More in detail, we use a semantic segmentation strategy similar to [27] that takes advantage of the components of \mathbf{n} to determine whether the plane is vertical, horizontal or oblique and to classify accordingly in an automatic way if it is a part of roof, façade or pavement: vertical planes are annotated as façades, oblique planes are labeled as roofs, and horizontal planes are classified as pavement or roofs based on their elevation. An example of semantic segmentation of a building is shown in Fig. 4(b).

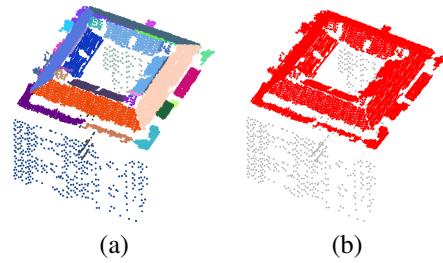


Figure 4: Segmentation of the point cloud relative to a building: (a) initial segmentation into planes, (b) semantic segmentation into roofs and façades.

5.3. Characterization of roof pitches

Every \mathcal{P} classified as roof corresponds to a roof pitch, which we characterize by two angles, θ_R, ϕ_R , representing the tilt of the pitch with respect to the horizontal and azimuthal planes, respectively, and a polygonal mesh R encoding its geometry.

The angle θ_R coincides with the angle of the normal to the plane with respect to the xy -plane, and is obtained from θ as:

$$\theta_R = \begin{cases} \theta & \text{if } \theta \in [0, \frac{\pi}{2}] \\ \pi - \theta & \text{if } \theta \in (\frac{\pi}{2}, \pi] \end{cases}$$

The angle ϕ_R represents the clockwise angle between the projection of the normal onto the xy -plane and the positive y -semiaxis (the North direction). In case $\theta \in (\frac{\pi}{2}, \pi]$, ϕ is first updated as $\phi += \pi$ if $\phi \in [0, \pi]$ and $\phi -= \pi$ if $\phi \in (\pi, 2\pi]$, so that $\mathbf{n} = (\cos \phi \sin \theta_R, \sin \phi \sin \theta_R, \cos \theta_R)$ points outside the building. Then, ϕ_R is obtained by aligning ϕ to the North direction and defining it clockwise:

$$\phi_R = 2\pi - \begin{cases} \phi + \frac{3\pi}{2} & \text{if } \phi \in [0, \frac{\pi}{2}] \\ \phi - \frac{\pi}{2} & \text{if } \phi \in [\frac{\pi}{2}, 2\pi] \end{cases}$$

The roof mesh R is composed of an outer polygon R_1 , representing the external boundary of \mathcal{P} , and possibly other polygonal cells R_i , representing internal holes, such as courtyards or atriums, or protrusions, like chimneys. Consequently, we have $R = \{R_i\}_{i=1,\dots,N}$, with $R_1 \supset R_i$ and $R_i \cap R_j = \emptyset$ for all $i, j = 2, \dots, N$. We determine R_1 by constructing an α -shape [44] from \mathcal{P} and extracting its boundary (Fig. 5(a)). We then exploit the boundary MATLAB function to tighten the boundary around the points, using a 0.8 shrink factor, and apply the Douglas Peucker (DP) algorithm [45] to reduce the number of boundary points (Fig. 5(b)). Next, if the building footprint contains any internal cavities in correspondence of \mathcal{P} , these are added to R as new polygons R_i . Similarly, if any of the polygons bounding the other pitches are contained within R_1 , we add them to the mesh. To ease the computation, \mathcal{P} is preliminarily aligned to the xy -plane, and the same rotation is then applied backwards to the resulting mesh.

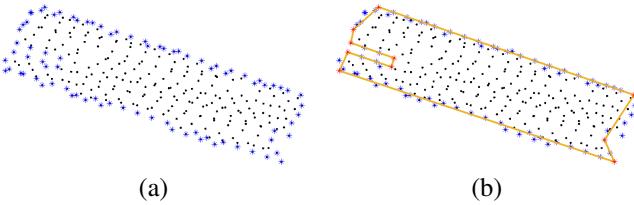


Figure 5: Extraction of the outer polygon of a roof pitch: blue points represent the boundary of the α -shape and red points represent the vertices of the polygon, extracted through the DP algorithm.

5.4. Meshing the semantic point cloud

As explained later on, the shadow computation contributing to the energy production estimation needs a surface representation of the urban environment. Directly meshing the LiDAR point cloud typically produces noisy and unbalanced meshes, with some very dense and other unsampled areas (such as whole missing facades occluded in the flight direction), hard to manage afterwards. For these reasons, we exploit the method described in [46] to refine or generate vertical walls of buildings,

reduce noise in the representation of roofs, and create an iconic representation of trees. In this way, we prepare the point cloud of the whole urban area, we annotate it and we create the final mesh necessary for the shadow computation.

We apply the HT-based recognition method using different families of geometric primitives, such as spheres and ellipsoids, to recognize the tree crowns, or cylinders to identify bare trees in the winter season. Therefore, we provide two iconic representations of trees for the two periods of the year: autumn/winter and spring/summer; this is our way to account for the different type of shadow that a tree can cast on adjacent roofs during the year. Fig. 6 shows the resulting iconic representation: trees free of leaves in the autumn/winter season are represented by cylinders, while trees composed of crown and trunk in the spring/summer season are rendered through half ellipsoids combined with cylinders.

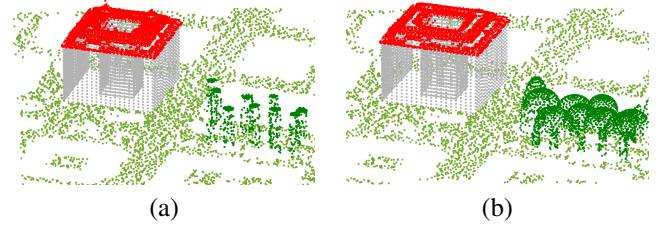


Figure 6: Iconic representation of trees in the autumn/winter (a) and spring/summer season (b).

Once the city point cloud is cleaned up from noise and resampled, we apply the widely-used Agisoft Metashape² software to generate the 3D surface mesh.

6. Energy production of a photovoltaic system

The estimation of the energy production of a PV system installed on a building roof is based on two routines. The first analyzes the roof geometry and determines the maximum number of PV modules that can be installed, while the second computes the amount of energy produced by each of these modules in a given time range.

6.1. Construction of a photovoltaic system

We start from the output of Section 5.3, i.e., a collection of polygonal meshes $\{R\}$ representing the roof pitches of a building and the corresponding tilt angles θ_R, ϕ_R . Each polygonal mesh is composed of an outer polygon and possibly other polygons representing holes of the roof pitches. For simplicity, we detail the case of a roof composed of a single pitch, with the case of multiple pitches being straightforward to extend. We consider a PV module p of length p_x and width p_y .

In a preliminary stage, the roof R is translated to the origin to avoid numerical issues related to geographical coordinates, and rotated around the z axis by an angle ϕ_R , so that the roof gets flattened on the xy plane. Then, we compute the direction of the maximum edge in R and apply a second rotation to align it with the x axis.

The installation of a PV system on a roof typically requires to allow a minimum free space ϱ , called *buffer*, from the roof

²<https://www.agisoft.com/>

boundary. We therefore need to compute an inset of R_1 and an offset of each $R_{i>1}$ of size ϱ . Geometrically speaking, the *buffered roof* is a mesh defined as

$$R^B = \{R_1 \ominus B_\varrho^n; R_2 \oplus B_\varrho^n; \dots; R_N \oplus B_\varrho^n\},$$

where B_ϱ^n is a regular dodecagon ($n = 12$) of radius ϱ , and \oplus and \ominus are the Minkowski sum and difference, see Section 3.2.

Next, we decide the direction of the installation of the PV modules, provided that all modules must be installed in the same direction, and therefore this choice affects the total number of modules that will fit on the roof. We start by considering the Axis-Aligned Bounding Box (AABB) of R and observing that the reasonable directions for placing the modules in the AABB are only those represented by the coordinate axes x and y . Let $p_x \times p_y$ be the module size and $R_x \times R_y$ the AABB size, then the number of modules that fit the AABB is

$$\begin{aligned} n_1 &= \lfloor R_x/p_x \rfloor \lfloor R_y/p_y \rfloor && \text{if modules are along } x \\ n_2 &= \lfloor R_x/p_y \rfloor \lfloor R_y/p_x \rfloor && \text{if modules are along } y \end{aligned}$$

where $\lfloor \cdot \rfloor$ indicates the integer part. If $n_2 > n_1$, the orientation along y is more convenient and we turn the modules by switching p_x and p_y . Now we can build a grid mesh of n_1 (or n_2) cells of size $p_x \times p_y$ (or $p_y \times p_x$) inside the AABB, and delete all cells with at least one vertex outside R^B , also considering holes.

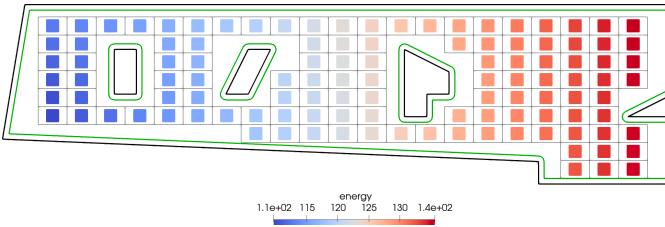


Figure 7: Original roof R (black), buffered roof R^B (green) and PV modules, colored with respect to their yearly energy production $E(p)$.

A synthetic example of a PV system is presented in Fig. 7. The output of the procedure consists of a text file containing a list of PV modules that can be installed on the building roof. Specifically, for each module p we print the centroid \mathbf{x}_p , an integer b_p indicating the ID of the building, and the horizontal and azimuthal tilt angles $\theta_p = \theta_R, \phi_p = \phi_R$, i.e., PV modules are assumed to be coplanar with their roof pitch.

6.2. Energy of a photovoltaic module

We detail here the estimation of the electric energy produced by a PV module p in a time period $T = [t_0, t_1]$, with a time step of one hour. We use hourly solar radiation data from the database of the Typical Meteorological Year (TMY), defined by ISO 15927-4 [47] and elaborated by the Comitato Termotecnico Italiano (CTI) for the geographic context of application [48]. The methodology is based on the determination of the global solar radiation incident to p at each hour $t \in T$.

In particular, we compute the Sun elevation α_t , the Sun azimuth from south γ_t^s , and the angle of incidence of solar radiation $\xi_{p,t}$ on p as

$$\begin{aligned} \sin(\alpha_t) &= \cos(\varphi) \cos(\delta_t) \cos(\omega_t) + \sin(\varphi) \sin(\delta_t), \\ \sin(\gamma_t^s) &= \cos(\delta_t) \sin(\omega_t) / \cos(\alpha_t), \\ \cos(\xi_{p,t}) &= \cos(\alpha_t) \cos(\theta_p) + \sin(\alpha_t) \sin(\theta_p) \cos(\gamma_t^s - \phi_p), \end{aligned}$$

where δ_t represents the daily Sun declination w.r.t. the orbital angle and ω_t the hourly Sun angle w.r.t. the local solar angle, computed as in [49], see Fig. 8.

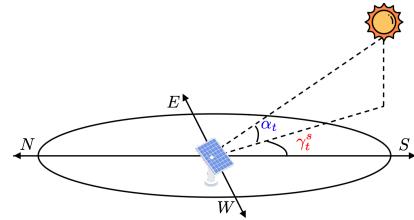


Figure 8: Angles involved in the radiation computation.

The solar beam radiation $I^{\text{beam}}(p, t)$ and diffuse radiation $I^{\text{diff}}(p, t)$ on p are then derived from the reference solar beam and diffuse radiation on a horizontal surface $I_t^{\text{h, beam}}, I_t^{\text{h, diff}}$ as:

$$I^{\text{beam}}(p, t) = \frac{\cos(\xi_{p,t})}{\sin(\alpha_t)} I_t^{\text{h, beam}}, \quad (1)$$

$$I^{\text{diff}}(p, t) = \left(\frac{1 + \cos(\theta_p)}{2} + \sigma \frac{1 - \cos(\theta_p)}{2} \right) I_t^{\text{h, diff}}. \quad (2)$$

The first term in $I^{\text{diff}}(p, t)$ represents the component from the sky, while the second is the fraction reflected by the ground with albedo σ (in our tests, we assumed a constant $\sigma = 0.2$ for the urban environment).

To obtain a more realistic estimation of the solar beam radiation, we also compute a binary shadow coefficient $S(p, t)$ to determine if p is reached by Sun light at time t , through the ray-tracing algorithm presented in [50]. Using the city coordinates (φ, λ) , we determine the hourly Sun rays direction \mathbf{d}_t and consider the ray r originating from \mathbf{x}_p with direction $-\mathbf{d}_t$. Intersecting this ray with the city mesh, we check if the module p at time t is shadowed by obstacles like trees or other buildings:

$$S(p, t) = \begin{cases} 0 & \text{if } r(\mathbf{x}_p, \mathbf{d}_t) \cap \mathcal{M} = \emptyset \\ 1 & \text{else} \end{cases} \quad (3)$$

Note that, this is the first (and only) time we use the surface mesh of the city, as all computations done so far were based on the roofs polygons extracted from the point cloud.

The electric energy produced by the module p at the t -th hour of the year is then obtained according to the procedure described by EN15316-4-6:2007 [49] as

$$E(p, t) = (I^{\text{diff}}(p, t) + I^{\text{beam}}(p, t) S(p, t)) \Omega f, \quad (4)$$

where Ω is the module power and f an efficiency factor relative to the type of PV modules installed.

By summing up the hourly E values, we straightforwardly define the total energy of a module in a given time range T as $E(p) = \sum_{t \in T} E(p, t)$. In the synthetic example of Fig. 7, we mimick the scenario of a building with a shadowing structure at its left, e.g. a tree or another building, which blocks sunlight and causes the leftmost PV modules to produce less energy.

We measure the efficiency of a PV module through the concepts of Tilt and Orientation Factor (TOF), Solar Access Factor

(SAF), and Total Solar Resource Fraction (TSRF):

$$\text{TOF}(p) = \sum_{t \in T} \frac{I^{\text{diff}}(p, t) + I^{\text{beam}}(p, t)}{I_t^{\max}}, \quad (5)$$

$$\text{SAF}(p) = \sum_{t \in T} \frac{I^{\text{diff}}(p, t) + I^{\text{beam}}(p, t) S(p, t)}{I^{\text{diff}}(p, t) + I^{\text{beam}}(p, t)}, \quad (6)$$

$$\text{TSRF}(p) = \text{TOF}(p) \text{SAF}(p). \quad (7)$$

The TOF takes into account how the slope and direction of a module impact the solar energy that it receives, comparing it to the energy I_t^{\max} received by a module with optimal orientation ($\theta = 33, \phi = 198$ for Matera, $\theta = 35, \phi = 202$ for Genova). The SAF quantifies how significantly shade is reducing the sunlight, expressing the available solar energy as a percentage of shade-free conditions. The TSRF is therefore a measure of the available solar energy for the module, considering both the amount of energy that the module will receive in the time range T given its position, tilt and orientation (TOF), and how much of that energy will reach the module given shading (SAF).

6.3. Energy of a photovoltaic system

The energy production of a PV system installed on a roof R is obtained by collecting the energy contributions from all its modules: $E(R) = \sum_{p \in R} E(p)$; if the building roof is composed of multiple pitches we sum the contribution from each pitch.

Table 1 summarizes the input required by the energy computation algorithm, which includes constants about the city and the PV modules, a list of modules computed as in Section 6.1, the solar radiation database, and the efficiency constraints.

When analyzing large numbers of buildings (e.g., at district or city scale), we introduce constraints to improve efficiency and reduce computation. For example, we can exclude north-oriented roofs with θ_p outside $[\beta_{\min}, \beta_{\max}]$, as they receive little sunlight. Modules with TSRF below a threshold \mathcal{T} are also discarded as inefficient. Finally, buildings with very small PV systems can be excluded if their potential yield does not justify refurbishment. The coverage threshold C defines the minimum power ($\Omega \times \text{number of modules}$) per building footprint area.

In Fig. 9, we report the PV system computed on a real building from the test case of Genova (see Section 7). From a comparison with the real building in Fig. 9(a), we can appreciate how the characterization correctly identifies the main pitches, despite the non-trivial shape of the balcony, the inner courtyard and the four rooflights. However, some small pitches are identified as well, in correspondence of these structural elements. Moreover, two of the four small protrusions in the middle of the inner pitches are missed, maybe due to their relatively small size with respect to the resolution of the point cloud.

Following Section 6.1, we estimate the largest PV system for each pitch, graphically represented as a grid (Fig. 9(b)). We note that the use of the buffer, besides meeting practical installation requirements, also hinders placing modules on tiny pitches. Indeed, a denoising step removes all pitches without modules. Last, we use the 3D mesh of the city to compute the energy of each module, as explained in Section 6.2 (Fig. 9(c)).

7. Results

We test the proposed method on two large-scale, real-world urban areas with complex morphology: the cities of Genova and

Category	Symbol	Parameter	Unit
Constants	φ, λ	city latitude and longitude	°
	σ	ground albedo	
	M	city surface mesh	
	ϱ	buffer width	m
	$p_x \times p_y$	PV module size	m
	Ω	PV module power	W
PV Modules List	f	PV module efficiency factor	
	x_p	module centroid	m
	b_p	building ID	
	θ_p	module horizontal tilt angle	°
Local Solar Database	ϕ_p	module azimuthal tilt angle	°
	δ_t	daily Sun declination	°
	ω_t	hourly Sun angle	°
	$I_t^{\text{h, beam}}$	hourly beam horizontal radiation	W/m ²
Efficiency Constraints	$I_t^{\text{h, diff}}$	hourly diffused horizontal radiation	W/m ²
	I_t^{\max}	hourly maximum global irradiation	Wh/m ²
Efficiency Constraints	$\beta_{\min}, \beta_{\max}$	orientation bounds	°
	\mathcal{T}	minimum TSRF	
	C	minimum coverage	W/m ²

Table 1: Input parameters for photovoltaic energy computation.

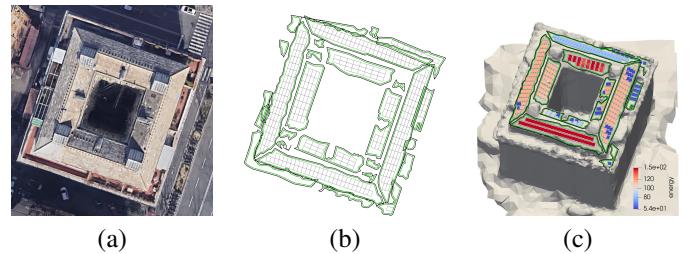


Figure 9: Photovoltaic energy production of a building. From left to right: real building, PV system, cleaned PV system with energy production on the surface mesh.

Matera, respectively in the north and the south of Italy. For both datasets, solar radiation data is based on the TMY, as described in Section 6.2. We considered polycrystalline modules of size $p_x \times p_y = 0.8 \times 1.3m$, with power $\Omega = 200W$ and efficiency $f = 0.75$; the buffer was set to $\varrho = 0.25m$.

7.1. Energy production of a building

We start by analyzing the energy production of a single building selected from the Genova dataset and, to better appreciate the results of our method, we do not apply any efficiency constraint (i.e., $[\beta_{\min}, \beta_{\max}] = [0, 360]$, $\mathcal{T} = 0$, $C = 0$). In Fig. 10 we show the PV energy produced by each module p in a random day of the year. On the left, the energy without considering shadows projected by neighboring urban elements is shown. This corresponds to the sum of the solar beam radiation and the diffuse radiation of (1). The global radiation is approximatively constant within each pitch, with higher values for south- and east-oriented pitches. In the middle, modules are colour-coded according to the amount of sunlight received, in terms of numbers of hours in which the module centroid is in light, computed with (3). Note how areas next to the chimneys in the internal pitches exhibit less sunlight hours, due to the shadow projected by the chimneys, and similarly for the eastern and western outer pitches in correspondence of the four rooflights. On the right, the final result, combining the two factors as in (4), with the same color scale as the left image.

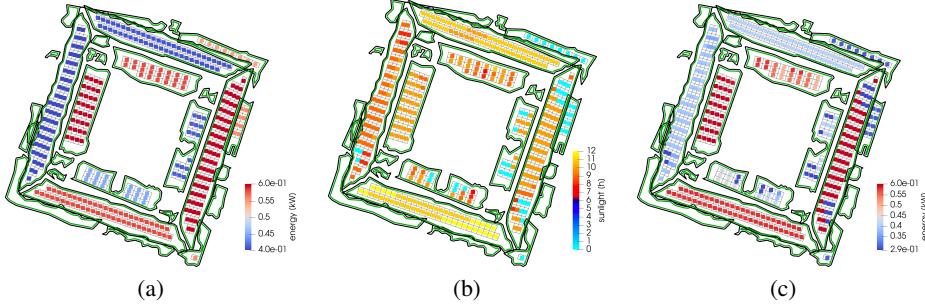


Figure 10: Energy production on a random day: (a) global solar radiation, (b) number of sunlight hours, (c) electric energy production.

In Fig. 11 we compare the energy production of the PV system in different days of the year. As expected, the energy produced in winter and in cloudy days is much lower than in summer and in sunny days. Moreover, pitches facing South correctly show a higher energy production with respect to those facing North, as they receive more direct radiation, supporting the choice to discard modules oriented northwards. When the weather is cloudy, the direct radiation is low and the modules mostly receive diffuse radiation, which does not depend on the orientation; for this reason, in cloudy days all modules produce a similar amount of energy.

These considerations are confirmed by Fig. 12, where we compare the energy production of the PV system across the winter and summer seasons (3 months each), and in the whole year. Note that each subfigure uses a distinct color scale, due to large disparities in value ranges that make cross-figure comparisons unreliable.

In Fig. 13 we compute the energy production of a building using two different meshes for determining the illumination, relative to the spring/summer and autumn/winter times, respectively. The difference between the two meshes is given by the shape of the trees, which are free of leaves in autumn/winter, as explained in Section 5.4. This difference results in a different energy production of the PV system, as the shadow generated by spring/summer trees is larger than the one generated by autumn/winter trees.

7.2. Energy production of a district

We now analyze the PV energy production of a whole district, in the two considered cities, adding the efficiency constraints about pitch orientation $[\beta_{\min}, \beta_{\max}] = [45, 315]$, TSRF $\mathcal{T} = 0.7$, and minimum power/footprint ratio $C = 0.05$.

For the Genova case study, we use aerial point cloud data in LAS format, georeferenced with EPSG 7791, relative to the 2018 aerial acquisition, and integrate them with the building footprints dataset and the vegetation map indicating the position of each tree. All the data are publicly available in the MapStore2 public geoportal, hosted by the Genova Municipality [51]. Accurate instructions about how to get the data used in this paper are available on the project GitHub repository. The dataset is structured into square tiles, and the case study specifically focuses on the portion of the town captured by tile number 3820, an area of approximately 2.3km^2 that contains around 1500 buildings surrounding the main train station “Brignole”. In Fig. 14 we present the energy production in a time period of one year, with PV modules embedded in the mesh and colored based on their energy production.

The city of Matera features a unique, irregular historical center (the “Sassi di Matera”, a UNESCO World Heritage site). Our case study spans about 1.1km^2 , covering part of the Sassi and the historical center, with around 900 buildings (Fig. 15). We use 2023 aerial point cloud data (LAS format, EPSG 25833), confidentially provided by the Matera Municipality, combined with building footprints and tree maps from the public Geo-Topographic Database (DBGT) [52, 53]. We also integrate the “Piano Paesaggistico Regionale” (PPR) GIS shapefile [54] to exclude protected buildings and landscapes where refurbishments are not permitted.

The most delicate part of our pipeline is the recognition strategy, introduced in Section 5, that identifies the tilt of a roof pitch \mathcal{P} with respect to the horizontal and azimuthal planes starting from the equation of its best-fitting plane \mathcal{S} . To evaluate the accuracy of this step, we use the *Mean Fitting Error* (MFE), an approximation quality measure introduced in [55] for primitives recognition on CAD objects:

$$\text{MFE}(\mathcal{P}, \mathcal{S}) = \frac{\sum_{x \in \mathcal{P}} d(x, \mathcal{S})}{\#\mathcal{P} |\mathcal{P}|}$$

where $\#\mathcal{P}$ is the cardinality of \mathcal{P} and $|\mathcal{P}|$ is the length of its AABB diagonal. Being the MFE normalized by definition, its value can be interpreted as a percentage. We computed the MFEs of all buildings in the districts of Genova and Matera selected in our study. The statistics obtained for Genova are: mean 1.8%, median 0.9%, std 3%, mode 0.1%, while those of Matera are: mean 1.4%, median 0.4%, std 2.9%, mode 0.1%. The mean value obtained in both cases is less than 2%, the median is less than or equal than 1%, and the value that occurs for most of buildings is 0.1%. Finally, the low standard deviation values confirm that the MFEs tend to be close to the mean in both district, despite their complex morphology.

7.3. Impact of the threshold parameters

The threshold parameters introduced in Section 6.3 directly influence the number of modules (M), systems (S), and overall city energy production (E). We assess their impact on energy estimates using the Matera case study, comparing M , S , E , and the average energy per module (E/M) and per system (E/S).

We start by analyzing the behavior of these ratios when varying one parameter at a time. Fig. 16 shows the energy produced in one year by an average PV module (E/M), or system (E/S) for different values of $\beta_{\min}, \beta_{\max}, \mathcal{T}$, and C . In all cases, a higher threshold value leads to higher energy production, meaning that we have less modules or systems, but more efficient.

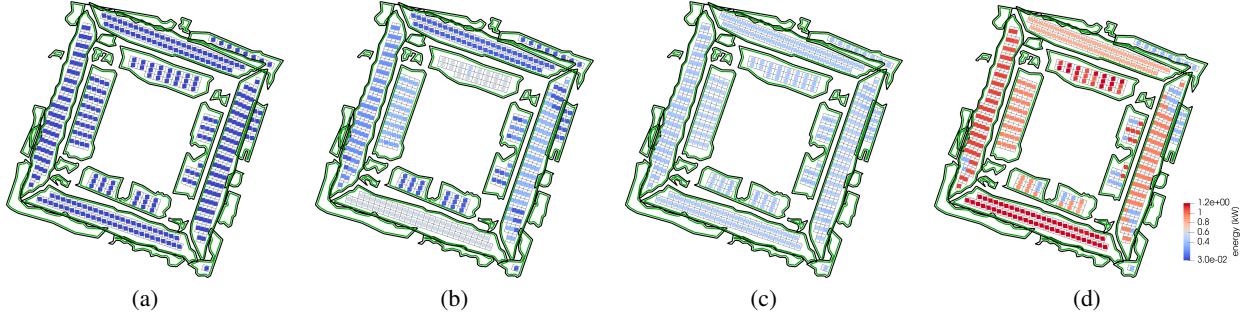


Figure 11: Energy production on different days, in the same color scale: (a) winter cloudy day, (b) winter sunny day, (c) summer cloudy day, (d) summer sunny day.

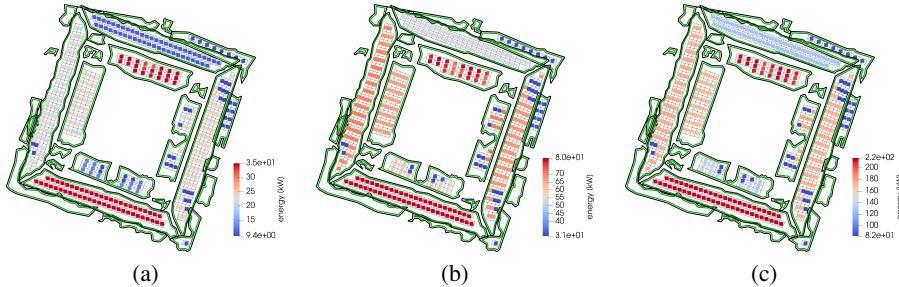


Figure 12: Energy production for different time periods, in different color scales: (a) winter, (b) summer, (c) all year.

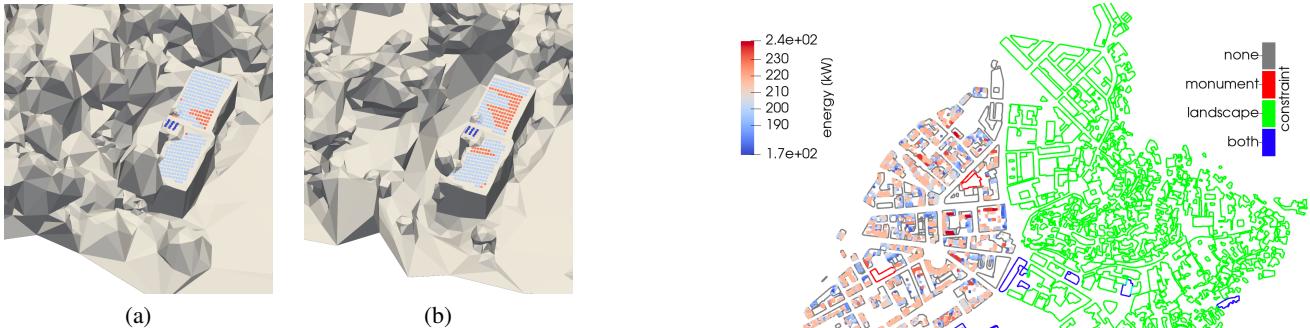


Figure 13: Difference in energy production with different types of trees: (a) mesh with iconic trees representing the spring/summer season, (b) mesh with iconic trees representing the autumn/winter season.



Figure 14: PV energy simulation on the case study area of Genova.

As a further analysis, Table 2 reports the effect of applying multiple constraints simultaneously. Starting from test 1 (no constraints), we progressively introduce more restrictive filters on acceptable modules and systems. This results in a gradual

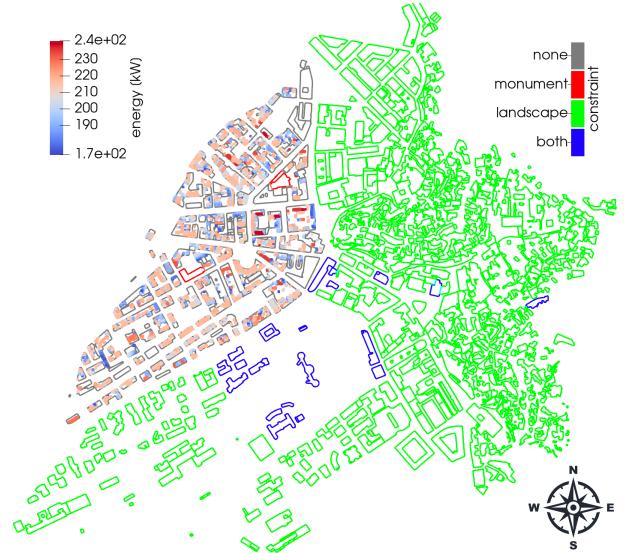


Figure 15: PV energy simulation on the case study area of Matera. Building footprints are colored with respect to their constraint characterization and only those without constraints have been processed.

reduction in the number of modules, systems, and total energy produced. However, excluding north-facing roofs (test 2) and low-efficiency modules (test 3) significantly increases the average energy per module (E/M). Although E/S slightly decreases, this motivates the final constraint C , which filters out inefficient systems without affecting module efficiency. Comparing test 4 to test 1, we observe 35% fewer systems, 55% fewer modules, and 40% less total energy, but a 36% increase in E/M , while E/S remains nearly stable (-5%). Additionally, test 2 yields a 10% reduction in computation time, as angle filtering is applied before energy simulation; the other constraints

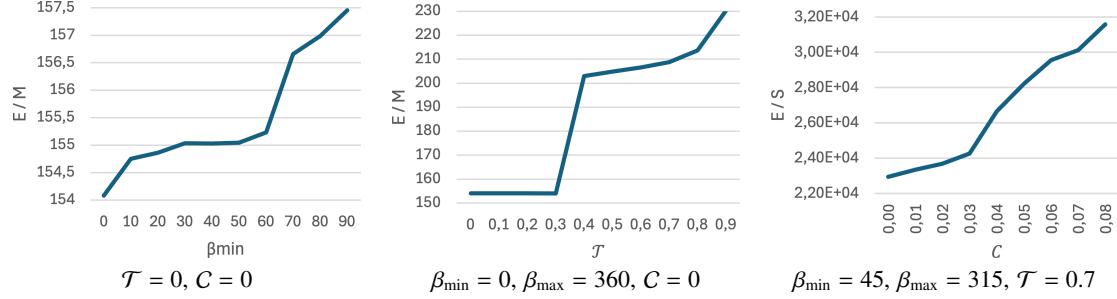


Figure 16: Yearly energy (in kW) produced on average by the PV modules and PV systems with different values of parameters β_{\min} , β_{\max} , T , and C . In the first plot only β_{\min} is reported, assuming conventionally that $\beta_{\max} = 360 - \beta_{\min}$.

are post-processing filters and do not affect timing.

7.4. Limitations and discussions

Although the roof extraction method is fairly robust with respect to noisy input data, it may still miss small obstacles such as chimneys; therefore, the resulting estimation should be considered an upper bound of the true energetic production. Due to limited model resolution, shadow computation is performed only at each panel’s centroid, as using more points would increase cost without guaranteed accuracy gain.

The process relies on an initial building-wise point cloud segmentation, guided by building footprints. These are sometimes given by local authorities, or downloaded from open-source platforms like OpenStreetMap [37]. Unfortunately, they may suffer of geometric defects, such as overlaps and self-intersections. We automatically resolve some problematic configurations, but a few buildings remain unresolved and require manual correction. As future improvement, we are experimenting buffer regions around footprints, larger areas with possible overlapping, and other spatial subdivision strategies to minimize the occurrence of this issue.

Additionally, roofs may extend beyond their footprint boundaries. To address this, we are currently testing a Voronoi-based segmentation strategy for the initial point cloud. We avoid using buffers in this context, as Mediterranean historic city centers typically feature narrow streets with densely packed buildings.

We chose a small buffer width and compact PV module size to emphasize the effect of roof shape on module placement and enhance sensitivity to shading, particularly relevant for dense, irregular cities like Matera, where larger modules would leave significant roof areas unused. In contrast, larger modules may suit simpler geometries, such as in industrial zones.

Another key design choice is module orientation. As detailed in Section 6.1, only two meaningful orientations remain after alignment. We select the one maximizing module count. Tests show that choosing the opposite orientation in Matera reduces the number of installed modules by about 200, with a proportional drop in energy output.

Regarding computational cost, most operations in the pipeline scale linearly with the number of building roofs analyzed, and thus with the number of buildings considered. In our implementation, the main bottleneck is shadow computation, as it relies on ray tracing and scales with mesh resolution. For Matera, the full simulation (test 4) on a mesh with 500K vertices and 1M triangles took 70 minutes for 900 buildings, while a smaller mesh (300K vertices, 600K triangles) reduced runtime to 50 minutes. The Genova case study (using a mesh

with 1.1M vertices and 2.3M triangles) required over 19 hours for 1500 buildings, on a 3.60 GHz Intel Core i7 (16 cores, 32 GB RAM). However, the process is highly parallelizable, and computation time could be significantly reduced by leveraging multicore architectures or by reimplementing the ray tracing method for GPUs.

8. Conclusions

In this paper, we have presented a workflow to estimate the energy production of a PV system installable on a building roof, taking into account the geometric nature of the urban space. This aspect, though crucial, has often been neglected or poorly addressed in related works. The contribution of this work is a pipeline integrating the significant number of factors determining the photovoltaic energy; though not all the routines are new or particularly challenging from the geometry processing point of view, the added values is in the integration of heterogeneous knowledge, data and processes to reach the final outcome. This work positions itself in the design and developments of Urban Digital Twins, and leverages computational geometry and engineering of the built environment in an interdisciplinary effort to improve evidence-based city management. However, we presented preliminary results that need thorough field experimentation and validation to provide insights on the quality of the achieved estimation. The next steps, therefore, foresee a careful validation of the results with experts, and through a comparison with already deployed implants. All the modeling procedures (roof recognition and characterization, pitch polygon extraction, wall refinement, tree crown seasonal modeling) will be essential parts of the reconstruction of an iconic city mesh model, corresponding to a Level of Detail LoD2 according to the CityGML standard [56].

Acknowledgments

This work is partially supported by the project “House of emerging technologies of Matera” (CTEMT) funded by the Ministry of Economic Development of Italy, CUP I14E20000020001.

D. Cabiddu, M. Mortara, C. Romanengo, and T. Sorgente are members of the RAISE Innovation Ecosystem, funded by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project “RAISE - Robotics and AI for Socio-economic Empowerment” (ECS00000035).

Table 2: For each combination of parameters $\beta_{\min}, \beta_{\max}, \mathcal{T}, C$, we report the number of modules M and systems S , the total energy produced E (in kW), the average energy per module and per system, and the computational time t (in minutes).

test	$[\beta_{\min}, \beta_{\max}]$	\mathcal{T}	C	M	S	E	E/M	E/S	t
1	[0,360]	0	0	$2.45 \cdot 10^4$	126	$3.78 \cdot 10^6$	$1.54 \cdot 10^2$	$3.00 \cdot 10^4$	77.17
2	[45,315]	0	0	$2.23 \cdot 10^4$	123	$3.46 \cdot 10^6$	$1.55 \cdot 10^2$	$2.81 \cdot 10^4$	70.42
3	[45,315]	0.7	0	$1.23 \cdot 10^4$	112	$2.57 \cdot 10^6$	$2.10 \cdot 10^2$	$2.29 \cdot 10^4$	70.73
4	[45,315]	0.7	0.05	$1.09 \cdot 10^4$	81	$2.28 \cdot 10^6$	$2.10 \cdot 10^2$	$2.82 \cdot 10^4$	70.17

T. Sorgente is affiliated to the Italian Gruppo Nazionale Calcolo Scientifico - Istituto Nazionale di Alta Matematica (GNCS-INdAM).

We also thank the Genova municipality for the LiDAR data of the city.

References

- [1] De Benedictis, R, Cesta, A, Pellegrini, R, Diez, M, Pinto, DM, Ventura, P, et al. Digital twins for intelligent cities: the case study of matera. *Journal of Reliable Intelligent Environments* 2025;11(6). doi:10.1007/s40860-025-00245-3.
- [2] Kodysh, JB, Omaitamu, OA, Bhaduri, BL, Neish, BS. Methodology for estimating solar potential on multiple building rooftops for photovoltaic systems. *Sustainable Cities and Society* 2013;8:31–41. doi:10.1016/j.scs.2013.01.002.
- [3] Nex, F, Remondino, F, Agugiaro, G, De Flippi, R, Poletti, M, Furlanello, C, et al. 3d solarweb: A solar cadaster in the italian alpine landscape. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2013;XL-7/W2:173–178. doi:10.5194/isprsarchives-XL-7-W2-173-2013.
- [4] Achbab, E, Lambarki, R, Rhinane, H, Saifaoui, D. Estimation of photovoltaic potential at the urban level from 3d city model (solar cadaster): Case of casablanca city, morocco. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2022;XLVI-4/W3-2021:9–16. doi:10.5194/isprs-archives-XLVI-4-W3-2021-9-2022.
- [5] Memme, S, Lepore, D, Priarone, A, Fossa, M. Estimating the solar potential of the rooftops of the city of genova through the 3d modelling of the built environment and anisotropic tiled sky analysis. *Journal of Physics: Conference Series* 2024;2893:012121. doi:10.1088/1742-6596/2893/1/012121.
- [6] Lee, S, Iyengar, S, Feng, M, Shenoy, P, Maji, S. Deeproof: A data-driven approach for solar potential estimation using rooftop imagery. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD ’19; New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016; 2019, p. 2105–2113. doi:10.1145/3292500.3330741.
- [7] Singh, R, Banerjee, R. Estimation of rooftop solar photovoltaic potential of a city. *Solar Energy* 2015;115:589–602. doi:10.1016/j.solener.2015.03.016.
- [8] Cheng, L, Zhang, F, Li, S, Mao, J, Xu, H, Ju, W, et al. Solar energy potential of urban buildings in 10 cities of china. *Energy* 2020;196:117038. doi:10.1016/j.energy.2020.117038.
- [9] Li, Y, Liu, C. Estimating solar energy potentials on pitched roofs. *Energy and Buildings* 2017;139:101–107. doi:10.1016/j.enbuild.2016.12.070.
- [10] Manni, M, Nocente, A, Kong, G, Skeie, K, Fan, H, Lobaccaro, G. Solar energy digitalization at high latitudes: A model chain combining solar irradiation models, a lidar scanner, and high-detail 3d building model. *Frontiers in Energy Research* 2022;10. doi:10.3389/fenrg.2022.1082092.
- [11] European Commission, JRC. Photovoltaic Geographical Information System (PVGIS). 2001. Accessed: 2025-04-19.
- [12] Google, . Project Sunroof. 2024. Accessed: 2025-03-11.
- [13] Nguyen, RMH, Brown, MS. Fast and effective l0 gradient minimization by region fusion. In: 2015 IEEE International Conference on Computer Vision (ICCV). 2015, p. 208–216. doi:10.1109/ICCV.2015.32.
- [14] Lin, Y, Li, J, Wang, C, Chen, Z, Wang, Z, Li, J. Fast regularity-constrained plane fitting. *ISPRS Journal of Photogrammetry and Remote Sensing* 2020;161:208–217. doi:10.1016/j.isprsjprs.2020.01.009.
- [15] Gu, Y, Cao, Z, Dong, L. A hierarchical energy minimization method for building roof segmentation from airborne lidar data. *Multimedia Tools and Applications* 2017;76(3):4197–4210.
- [16] Wang, X, Ji, S. Roof plane segmentation from lidar point cloud data using region expansion based l0 gradient minimization and graph cut. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 2021;14:10101–10116. doi:10.1109/JSTARS.2021.3113083.
- [17] Forgy, EW. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics* 1965;21:768–769.
- [18] Ester, M, Kriegel, HP, Sander, J, Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD’96; AAAI Press; 1996, p. 226–231.
- [19] Kong, D, Xu, L, Li, X, Li, S. K-plane-based classification of airborne lidar data for accurate building roof measurement. *IEEE Transactions on Instrumentation and Measurement* 2014;63(5):1200–1214. doi:10.1109/TIM.2013.2292310.
- [20] Zhang, K, Yan, J, Chen, SC. Automatic construction of building footprints from airborne lidar data. *IEEE Transactions on Geoscience and Remote Sensing* 2006;44(9):2523–2533.
- [21] Cao, R, Zhang, Y, Liu, X, Zhao, Z. Roof plane extraction from airborne lidar point clouds. *International Journal of Remote Sensing* 2017;38(12):3684–3703.
- [22] Xu, Y, Yao, W, Hoegner, L, Stilla, U. Segmentation of building roofs from airborne lidar point clouds using robust voxel-based region growing. *Remote Sensing Letters* 2017;8(11):1062–1071.
- [23] Li, L, Yao, J, Tu, J, Liu, X, Li, Y, Guo, L. Roof plane segmentation from airborne lidar data using hierarchical clustering and boundary relabeling. *Remote Sensing* 2020;12(9). doi:10.3390/rs12091363.
- [24] Xia, S, Chen, D, Wang, R, Li, J, Zhang, X. Geometric primitives in lidar point clouds: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 2020;13:685–707. doi:10.1109/JSTARS.2020.2969119.
- [25] Schnabel, R, Wahl, R, Klein, R. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum* 2007;26(2):214–226. doi:10.1111/j.1467-8659.2007.01016.x.
- [26] Limberger, FA, Oliveira, MM. Real-time detection of planar regions in unorganized point clouds. *Pattern Recognition* 2015;48(6):2043–2053. doi:10.1016/j.patcog.2014.12.020.
- [27] Romanengo, C, Cabiddu, D, Pittaluga, S, Mortara, M. Building semantic segmentation from large-scale point clouds via primitive recognition. *Graph Models* 2024;136(C). doi:10.1016/j.gmod.2024.101234.
- [28] Mukhopadhyay, P, Chaudhuri, BB. A survey of Hough transform. *Pattern Recognition* 2015;48(3):993 – 1010. doi:10.1016/j.patcog.2014.08.027.
- [29] Hough, PVC. Method and means for recognizing complex patterns. 1962. US Patent 3,069,654.
- [30] Beltrametti, MC, Robbiano, L. An algebraic approach to Hough transforms. *J of Algebra* 2012;37:669–681. doi:10.1016/j.jalgebra.2012.09.012.
- [31] Raffo, A, Romanengo, C, Falcidieno, B, Biasotti, S. Fitting and recognition of geometric primitives in segmented 3D point clouds using a localized voting procedure. *Computer Aided Geometric Design* 2022;97:102123. doi:10.1016/j.cagd.2022.102123.
- [32] Barki, H, Denis, F, Dupont, F. A new algorithm for the computation of the minkowski difference of convex polyhedra. In: 2010 shape modeling

- international conference. 2010, p. 206–210. doi:10.1109/SMI.2010.12.
- [33] Wein, R, Baram, A, Flato, E, Fogel, E, Hemmer, M, Morr, S. 2D minkowski sums. In: CGAL User and Reference Manual; 6.0.1 ed. CGAL Editorial Board; 2024, p. 1. URL: <https://doc.cgal.org/6.0.1/Manual/packages.html#PkgMinkowskiSum2>.
- [34] The CGAL Project, . CGAL User and Reference Manual. 6.0.1 ed.; CGAL Editorial Board; 2024. URL: <https://doc.cgal.org/6.0.1/Manual/packages.html>.
- [35] Livesu, M. *cinolib*: a generic programming header only c++ library for processing polygonal and polyhedral meshes. *Transactions on Computational Science XXXIV* 2019;:64–76.
- [36] ESRI, . ESRI Shapefile Technical Description - An ESRI White Paper. 1998. URL: <https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf>.
- [37] OpenStreetMap contributors, . Planet dump retrieved from <https://planet.osm.org>. <https://www.openstreetmap.org>; 2017.
- [38] Jordan, C. *Cours d'analyse de l'École polytechnique*; vol. 1. Gauthier-Villars et fils; 1893.
- [39] Franklin, WR. Pnpoly - point inclusion in polygon test. 1970. URL: https://wrfraunfelder.org/Research/Short_Notes/pnpoly.html; last update: March 2023.
- [40] Barill, G, Dickson, NG, Schmidt, R, Levin, DI, Jacobson, A. Fast winding numbers for soups and clouds. *ACM Transactions on Graphics (TOG)* 2018;37(4):1–12.
- [41] Lafarge, F, Mallet, C. Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation. *International Journal of Computer Vision* 2012;99:69–85. doi:10.1007/s11263-012-0517-8.
- [42] Hoppe, H, DeRose, T, Duchamp, T, McDonald, J, Stuetzle, W. Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on computer graphics and interactive techniques. 1992, p. 71–78.
- [43] Romanengo, C, Falcidieno, B, Biasotti, S. Discretisation of the hough parameter space for fitting and recognising geometric primitives in 3d point clouds. *Mathematics and Computers in Simulation* 2025;228:73–86.
- [44] Edelsbrunner, H, Kirkpatrick, D, Seidel, R. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory* 1983;29(4):551–559. doi:10.1109/TIT.1983.1056714.
- [45] Douglas, DH, Peucker, TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 1973;10:112–122.
- [46] Romanengo, C, Falcidieno, B, Biasotti, S. From aerial lidar point clouds to multiscale urban representation levels by a parametric resampling. *Computers & Graphics* 2024;123:104022. doi:10.1016/j.cag.2024.104022.
- [47] ISO, E, et al. Hygrothermal performance of buildings-calculation and presentation of climatic data-part 4: Hourly data for assessing the annual energy use for heating and cooling. European Committee for Standardization (CEN) 2005;:16.
- [48] Comitato Tecnico Italiano, . Anni tipo climatici d'italia. Accessed: April 15, 2025. URL: <https://www.cti2000.it/index.php?controller=news&action=show&newsid=34985>.
- [49] Duffie, JA, Beckman, WA, Blair, N. Solar engineering of thermal processes, photovoltaics and wind. John Wiley & Sons; 2020.
- [50] Scalas, A, Cabiddu, D, Mortara, M, Spagnuolo, M. Potential of the geometric layer in urban digital twins. *ISPRS International Journal of Geo-Information* 2022;11(6):343.
- [51] Comune di Genova, . Mapstore2 3d city model viewer. Web Application; 2023. URL: <https://mappe.comune.genova.it/MapStore2/#/viewer/1000003072>; interactive 3D viewer for CityGML data of Genoa, Italy.
- [52] Regione Basilicata, . Database Topografico Regionale. Tema: Edificato. Accessed: April 15, 2025. URL: http://rsdi.regione.basilicata.it/webGis/shape-zip/edifici_max_estens.zip.
- [53] Regione Basilicata, . Database Topografico Regionale. Tema: Vegetazione. Accessed: April 15, 2025. URL: http://rsdi.regione.basilicata.it/webGis/shape-zip/albero_isolato.zip.
- [54] Regione Basilicata, . Piano Paesaggistico Regionale. Accessed: April 15, 2025. URL: <https://rsdi.regione.basilicata.it/viewGis/?project=SFCCEE499-0BEB-FA86-7561-43913D3D1B65>.
- [55] Romanengo, C, Raffo, A, Qie, Y, Anwer, N, Falcidieno, B. Fit4cad: A point cloud benchmark for fitting simple geometric primitives in cad objects. *Computers & Graphics* 2022;102:133–143. doi:<https://doi.org/10.1016/j.cag.2021.09.013>.
- [56] Open Geospatial Consortium (OGC), . Ogc city geography markup language (citygml) part 1: Conceptual model standard. Standard OGC 20-010; Open Geospatial Consortium; 2021. URL: <https://www.ogc.org/standard/citygml/>; version 3.0.