IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 February 2025, accepted 27 February 2025, date of publication 5 March 2025, date of current version 21 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3548513

RESEARCH ARTICLE

PROMORE: A Procedural Modeler of Virtual Rural Environments With Artificial Dataset Generation Capabilities for Remote Sensing Contexts

TELMO ADÃO[1,2], JOÃO CERQUEIRA[1], MIGUEL ADÃO[3], NUNO SILVA[1,4], DAVID PASCOAL[1,4], LUÍS G. MAGALHÃES[2], TIAGO BARROS[5], CRISTIANO PREMEBIDA[5], (Senior Member, IEEE), URBANO J. NUNES[5], (Life Senior Member, IEEE), EMANUEL PERES[1,4,6], AND RAUL MORAIS[1,4,6]

[1]Department of Engineering, School of Sciences and Technology, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
[2]ALGORITMI Research Centre/LASI, University of Minho, 4800-058 Guimarães, Portugal
[3]Polytechnic Institute of Bragança, School of Communication, Administration and Tourism, Campus do Cruzeiro, 5370-202 Mirandela, Portugal
[4]Centre for the Research and Technology of Agro-Environmental and Biological Sciences, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
[5]Department of Electrical and Computer Engineering, Institute of Systems and Robotics, University of Coimbra, 3030-790 Coimbra, Portugal
[6]Institute for Innovation, Capacity Building and Sustainability of Agri-Food Production, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal

Corresponding author: Telmo Adão (telmoadao@utad.pt)

This work is funding by the Vine and Wine Portugal Project, co-financed by the Recovery and Resilience Plan (RRP) and the European Union (EU) NextGenerationEU funds, within the scope of the Mobilizing Agendas for Reindustrialization, under the reference C644866286-00000011.

ABSTRACT Remote sensing (RS) is a rapidly evolving field that facilitates the study of phenomena on the Earth's surface. Through various platforms, including satellites, manned aircraft, and remotely piloted aerial vehicles (RPAV), RS has been strategically applied to critical sectors like agriculture and forestry, which are essential for humanity's sustenance. Key applications include crops classification, yield estimation and livestock monitoring and quantification. In the era of artificial intelligence (AI), the development of deep learning (DL) models for such applications often requires extensive field data collection and labor-intensive image labeling, which are both time-consuming and resource-intensive. To address these challenges, this paper presents Procedural Modeling of Rural Environments (PROMORE), a parameterizable, ontology-driven system designed to generate 3D virtual environments encompassing forestry, farmland – mainly focused on vineyards – and village settings. This system also implements functionalities to automate the extraction of training data for deep learning applications in remote sensing, with the declared aim of providing complementary capabilities to data augmentation techniques, encompassing both traditional methods (e.g., flips, rotations, zooming) and advanced approaches such as generative adversarial networks (GANs). By simulating RPAV flights and managing virtual object visibility, PROMORE enables the automatic labeling, delineation, and highlighting of elements of interest (e.g., vine plants, trees, buildings), facilitating the generation of datasets tailored for tasks such as semantic segmentation, and object detection.

INDEX TERMS Computer graphics, virtual environments, 3D modeling, procedural modeling, artificial data engineering, artificial dataset, deep learning.

I. INTRODUCTION

Precision agriculture (PA) allies technology and data sciences to traditional practices for the sustainable optimization of the crops production in terms of both quantity and quality, with

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

an increasingly relevant sense of environmental awareness. For example, by applying specific sensors sets, an effective management information system for irrigation [1] and pests control [2] can be achieved, to maximize the efficiency of resources, simultaneously enhancing biophysical growth processes and mitigating resource wastage and ecological footprints.

One increasingly popular strategy for managing agricultural resources involves the use of remote sensing (RS), particularly in applications related to vineyard studies. Through the use of remotely piloted aerial vehicles (RPAVs) equipped with various sensors – including red-green-blue (RGB), multi/hyperspectral and near-infrared (NIR) –, a wide range of tasks to assess crops status can be performed within and beyond the visible range of the electromagnetic (EM) spectrum [3].

When combined with artificial intelligence (AI) approaches, RS significantly enhances its decision-support capabilities [4], enabling the automation of essential activities such as crops' yield estimation and livestock quantification. This integration also supports other relevant tasks, such as providing crop status information, which is useful for implementing corrective and preventive interventions with direct benefits for both yielding and the environment sustainability.

Nevertheless, the preparatory tasks related to AI, and more specifically, to deep learning (DL), requires a significant amount of logistics that increase the cost of the implementation of decision support systems (DSS), including: a) RPAV-based field missions for data acquisition; and b) burdensome and extensive manual data labeling for training the AI models [5]. Regarding the latter requirement, on the one hand, according to Steier et al. [6], manually-made annotations for RS tasks often result in unreliable ground truth, especially in natural environments (e.g., forestry contexts). Complementarily, to improve DL models' inference capabilities by expanding the number of training examples, data augmentation is a widely utilized practice in supervised tasks. Two prevalent techniques include traditional methods and generative adversarial networks (GANs) [7]. However, since both rely on strategies to generate dataset-based variants from existing images or pre-learned features, they carry an inherent risk of oversampling [8].

To tackle these drawbacks, the current work proposes a framework named of Procedural Modeling of Rural Environments – PROMORE. While relying on a specific ontology and user-defined parameterization to guide the stochastic generation of virtual 3D rural-like scenarios, PROMORE also includes functionalities for the automatic derivation of datasets designed to train deep learning models for decision-making in remote sensing applications. Regarding its (post-parameterization) operational pipeline, PROMORE starts by generating Voronoi-based parcels, and, then, the network of roads. Afterwards, each parcel is assigned to a type of user-defined role for rural-related assets distribution (e.g., vineyard, forestry, human occupation, etc.). Considering those role assignments, the assets are then placed, resorting to three styles of random distributions: grid-, radial-, noise-, and border-based. In the first stage of such placement, the representative objects rely on low-poly placeholders that are used to probe inconsistencies during the setting up of virtual environments. In a subsequent stage, these low-poly objects are replaced by the final models with enhanced visual fidelity. Afterwards, lower priority assets are randomly placed – e.g., grass, bushes and flowers. Once the virtual environment is finalized, a nadir navigation mode is made available, enabling automatic non-overlapped RPAV-like flights over the 3D terrain and its assets, with the aim of extracting accurately machine-based annotated datasets that serve as labeled ground-truth consisting either in region masks or bounding boxes, respectively, for semantic segmentation or object detection tasks. PROMORE not only mitigates the need for burdensome and time-consuming RPAV-based data acquisition and the associated manual, error-prone labeling and outlining activities, but also enables diversification in ways that common data augmentation techniques cannot achieve.

The main contributions of this work are as follows: i) the specification of a lightweight, high-level, reconfigurable, and scalable ontology for characterizing a rural context; ii) the development of an ontology-driven procedural modeling pipeline for generating realistic, rural-like virtual environments; and iii) the implementation of automated functionalities for extracting fine-tuned masks and regions from the generated virtual environments, designed to facilitate the creation of remote sensing-oriented datasets. These functionalities rely on the parametric flexibility provided by modern computer graphics engines to compose datasets tailored for AI-related tasks, such as object detection and segmentation. It is important to note that the validation of DL models using the synthetic data generated is beyond the scope of this work and will be addressed in future research.

The remainder of this work consists of the following: besides introduction, Section II addresses the related work in procedural modeling, including generative solutions conceived for synthetic datasets extraction; the concept and specification of the proposed rural ontology are presented in Section III, along with key-design aspects of PROMORE; Section IV outlines the implementation details; Section V documents the tests and results made to PROMORE, focusing operational effectiveness and computational performance; then, a brief discussion is presented in Section VI; and finally, Section VII sums up the paper and poses future work challenges.

II. RELATED WORK

Procedural modeling is a field of computer graphics focused on the efficient and visually faithful production of virtual models and extensive environments. Typically, it operates by deploying and/or converting rule-based structures into coherent visible layouts (including geometry, textures, etc.) to represent real-world phenomena or elements, built upon algorithmic engines capable of systematizing outputs with diversification that is often parameterizable at a high level [9], [10]. The production of buildings [10], cities [11], [12], game environments [13], [14], 3D cinema [15], as well as the generation of vegetation [16], [17] are among the vast range of applications sustained by this modeling approach.

Beyond well-known methods for procedural content generation such as noise functions [18] and tiling [19], other approaches relying in rule-based strategies are worth considering. For example, rewriting grammars (e.g., [11], [20]), which involve iterative rule-based replacements that transform a shape's geometry, have gained popularity in the field. Additionally, ontology-based methods organize architectural knowledge by domain, vocabulary, relations, and instances, fostering scalability and reusability through object-oriented principles [9], [21]. Within the context of ontology-like structures, models have been proposed to establish standards for visualizable geometric data. A note-worthy example is the Building Information Modeling (BIM) [22], which supports the digital detailing of architectural, structural, and engineering elements and enables the management of buildings' construction life-cycle [23]. Other standard proposal for regulating the production of virtual scenes, but more focused in wider landscapes such as urban representations, is CityGML [24]. With a simpler element-wise data composition compared to BIM, CityGML supports the assignment of levels-of-detail (LOD), placing greater emphasis on aspects that are also relevant for the visualization/interaction experience, such as the effective management of computational resources.

Regarding the generation of environments involving natural elements and landscapes, recent works have been proposed, highlighting the pertinence and timeliness of the topic. For example, in [25], a hybrid approach combining geometry-based and volumetric modeling techniques with gradually transitioning LOD was proposed for automating the production of forest-like environments. More specifically, for near trees, a particle flow algorithm was incorporated to generate ramifications from leaves to roots. Conversely, a pixel-based-technique allowed rendering vast forests with millions of unique trees in distant view plans. Such work reported benefits in the management of GPU resources. Another extensive forest generation approach was addressed in [26], with the development of a method for maximizing vegetation cover relying in Perlin noise-based generated terrains, simulating the typical competitive process of nature. Good performances and adequate visual quality were reported. The management of green spaces in populated cities was addressed in [27] with a procedural modeling-based framework for urban planning activities encompassing ecosystem services, which integrates tools such as CityEngine (Esri R&D, Zurich, Switzerland) and ArcGIS (Esri, CA, USA). Acknowledging the need for mitigating urban heating and stormwater runoff, the design capabilities of that framework were demonstrated with the proposal of scenarios for increasing trees and turfs, considering Old Town Pocatello (Idaho, USA) as a case-study. Following a similar research scope, Niese et al. [28] proposed an approach identified as procedural placement models (PPMs) for vegetation in urban layouts, enhancing realism by integrating plant positioning sensitive to city geometry. Using satellite images and land register data, the models can learn and parameterize plant distributions, allowing the automatic population of virtual cities with detailed vegetation. This framework was validated through a perceptual user study, namely, involving experts in urban scene design sessions. In [29], a novel approach to generate a digital replica of the Great Barrier Reef resorting to the diffusion-limited aggregation (DLA) model as the main procedural technique was considered. The results show a good degree of similarity between the procedural generation and photos of the real environment. DL and procedural modeling strategies were also combined with success in the context of virtual vegetation [30], highlighting the great potential that can be harnessed from the cooperation between both research fields. More specifically, an inverse procedural modeling system was proposed to infer grammars out of plant-like geometric structures. To that end, the authors trained a convolutional neural network (CNN) with several L-system grammars and respective branches representations allowing the learning of atomic structures. Therefore, the purpose of their work is to obtain grammar-based rules out of already created geometries, instead of the traditional rendering of virtual structures oriented from grammars.

In [31], an application to import and manage photogrammetry-based virtual environments was developed, focusing rural fire simulations and the automatic extraction of datasets ready for training CNN segmentation models. Promising results in the use of these models in detecting flames in real images were attained. However, procedural modeling offers a potentially more expeditious and diverse approach for setting up datasets. In the scope of urban landscapes, in [32], a method to elaborate synthetic datasets for semantic segmentation is proposed, modeled based on procedurally generated real-world-like urban environments that are capable of variate diverse influence factors (weather and lightning, mostly). The influence of these factors on the performance of DeepLab v3+ [33] was then assessed. Puddle and cloud levels were found to have a greater impact compared to rain levels. Another method worthy of note is Meta-Sim2 [34], which is designed to learn scene structure from real images in an unsupervised manner. It combines rules sampling, reinforcement learning and feature space divergence between synthesized and target images to successfully train models capable of generating realistic data, which can be useful for extending datasets. In [35], a framework was proposed dedicated to the creation of large-scale synthetic datasets for architectural applications, including 2D and 3D annotations. The underlying pipeline generated customizable and class-balanced building models, supporting various DL tasks, including 3D reconstruction and segmentation. In the field of autonomous driving, a configurable system for generating synthetic street scenes using 3D graphics named Cosy was developed [36]. The advantages of this tool includes computational power-aware data gathering, as well as the generation of automatic ground-truth annotations. Designed to be extendable, Cosy enables researchers to tailor datasets to their specific needs. Targeting terrestrial robotic systems training, in [37], a solution capable of generating RGB images, point clouds with pixel-wise and point-wise annotations, as well as depth maps was proposed, alleviating the efforts for manual dataset elaboration. In 3 hours of autonomous operation, the system can generate 1000 frames dully annotated for machine learning tasks.

To tackle the apparent scarcity of procedural modeling tools for the semi-automatic generation of virtual rural environments and annotated datasets compliant with DL training for remote sensing contexts, this work proposes the PROMORE system, which will be detailed in the following sections, addressing both specification and implementation. Currently, PROMORE is capable of producing datasets compatible with recent object detection methods, such as YOLO [38], as well as region segmentation approaches in the line of U-Net [39] and Mask R-CNN [40].

III. PROMORE SPECIFICATION

This section details PROMORE's specification, including a guiding ontological structure, as well as its generation pipeline – from virtual terrain parceling and object distribution and placement procedures until the dataset extraction stage.

A. ONTOLOGICAL SPECIFICATION

Inspired by previous works [9], [22], [24], a guiding ontology has been specifically designed for PROMORE, which is depicted in Figure 1. It is built upon the abstract *VirtualObject* class, which, in turn, contains a *Representation* class associated to visualizable geometry and styling, common to all entities integrated in a generated virtual environment. From this base, the *Terrain* class emerges as a core category, holding *Road* components, representing transportation routes, and *Parcel* delimitations, which serve as land units housing both natural and artificial entities.

Each *Parcel* contains *Animal* and *Vegetation* instances. The *Animal* category is further specialized into subcategories, including *Cattle*, *Domestic*, and various other deeper taxonomic groups of increasing specificity. Similarly, *Vegetation* spans *Grass/Bush*, *Tree*, *Vine* and other eventual flora species to reflect ecological diversity. Alongside these natural elements, *Parcel* units also include man-made structures such as *Building*, *Machinery*, and *Furniture* entities, encompassing the human imprint.

Seasons can also be represented. One of the possible procedures may rely in extending some of the current elements and specialized them in association with the target season, without disregarding the importance of considering knowledge on phenological cycles [41], which vary from specie to specie.

B. VIRTUAL ENVIRONMENT GENERATION PIPELINE

The proposed pipeline consists of a set of stages that automate the creation of virtual rural environments. The process begins with the configuration of a portion of terrain, which is then subdivided into smaller, manageable parcels, wherein edges are considered as road delimiters. Afterwards, parcels are assigned to particular roles, reflecting their specific purpose within the virtual environment (e.g. forestry areas to zones for human exploration). Considering such assignment, placeholders are distributed across the parcels, booking the positions of various assets that will populate the scene. To complete the virtual environment, a random distribution of assets with a lower priority – i.e., more relevant for complementary purposed than for primary visual impact – is applied (e.g. low vegetation). The process culminates in the final rendering, where all elements are seamlessly integrated into a coherently structured virtual stage with rural characteristics. Figure 2 clearly identifies the described set of steps.

When the production of the virtual environment reaches its end, far-range remote sensing-oriented datasets can be rapidly and automatically generated through virtual camera flights simulating RPAV operations. These datasets are tailored to support object detection and segmentation tasks.

1) TERRAIN, PARCELS, ROADS, AND FUNCTIONAL ASSIGNMENT

The initial steps for the automated production of the virtual rural environment involve configuring a designated portion of terrain, wherein surface curling operations are applied to create topographic features. To that end, height fluctuations are induced to create a landscape with oscillations and, therefore, more realistic. Afterwards, a Voronoi tessellation method (based in [42]) is applied to a flat plane having a size and a position that match the referred terrain portion, partitioning it into distinct parcels – i.e., containment polygons further assigned with thematic labels for conditioning virtual objects' distribution – based on a proximity criterion that considers a set of randomly seeded points. Following the parceling process, roads are delineated along the edges of the Voronoi cells. Essentially, this step involves projecting the edges to the terrain, simulating paths that connect the different parcels. Afterwards, each generated parcel is assigned to role specifying land use – forest, vineyard, etc.. This classification determines the types of virtual assets used to populate the terrain's role-specific areas.

2) OBJECTS PLACEMENT

The placement of virtual objects, whether manually modeled or procedurally generated, is a key step in PROMORE, which

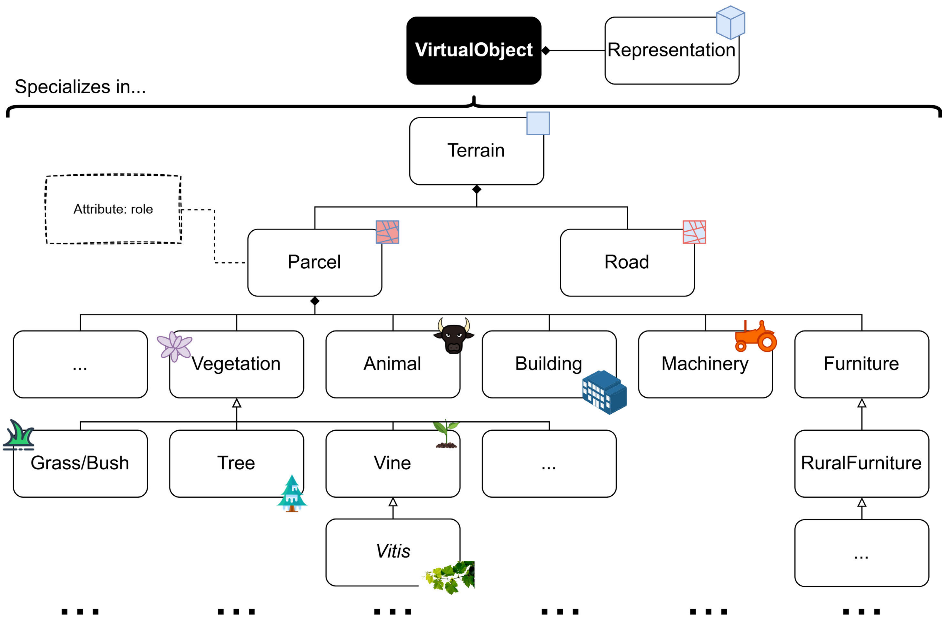


FIGURE 1. **PROMORE's guiding ontology.**

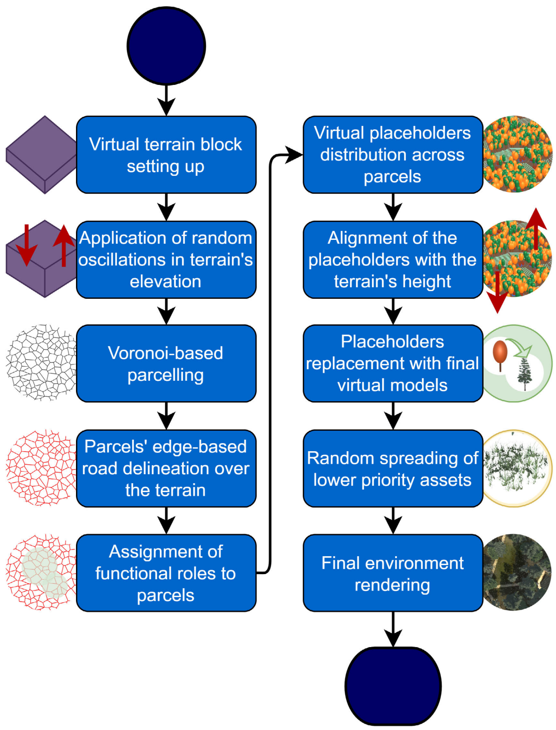


FIGURE 2. **PROMORE conceptual pipeline.**

uses grid-, radial-, noise-, and border-based distribution methods to spread seeds across the terrain plane (x, z coordinates).

The distribution of points within a polygon using a grid configuration, excluding non-intersecting ones, is given by the Algorithm 1. The algorithm iterates over the grid coordinates defined by an input spacing value and checks whether each determined point lies within a given polygon. Points that satisfy this condition are added to a list that is returned in the end.

Algorithm 1 Grid-Based Point Distribution Within a Polygon

Input: A polygon P defined by vertices $(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$.
Input: Grid spacing d.
Output: A list of evenly distributed points within the polygon P.

```
Function GridPointsWithinPolygon(P, d):
    x_min ← min(x_0, x_1, ..., x_{n-1});
    x_max ← max(x_0, x_1, ..., x_{n-1});
    y_min ← min(y_0, y_1, ..., y_{n-1});
    y_max ← max(y_0, y_1, ..., y_{n-1});
    points_list ← EmptyList();
    for x ← x_min to x_max by d do
        for y ← y_min to y_max by d do
            (x, y) ← GeneratePoint(x, y);
            if PointInsidePolygon((x, y), P) then
                Append(points_list, (x, y));
    return points_list;
```

The process for radial spreading of points within a polygon is another object distribution strategy among the adopted. The underlying concept is to evenly distribute points in circular configurations that are iteratively expanded by a given factor, until a stopping conditions is reached. Two of these conditions are based on: a) a maximum radius to limit

Algorithm 2 Noise-Based Point Distribution Within a Polygon

Input: A polygon P defined by vertices $(x_0, y_0), \ldots, (x_{n-1}, y_{n-1})$.
Input: Minimum distance d between points.
Input: Number of points N to generate.
Output: A list of N points randomly distributed within P.

```
Function GeneratePoints(P, d, N):
    points_list ← EmptyList();
    while Length(points_list) < N do
        (x, y) ← GenerateRandomPoint(P);
        if PointInsidePolygon((x, y), P) then
            isInSafeDistance ← true;
            for i ← 0 to Length(points_list) −1 do
                if Distance((x, y), points_list[i]) < d then
                    isInSafeDistance ← false;
                    break;
            if isInSafeDistance = true then
                Append(points_list, (x, y));
    return points_list;
```

the expansion; and b) the detection of the first point lying outside the polygon – which also results in the discard of that point. In mathematical terms, let r_0 be the initial radius, f be the expansion factor, and R be the maximum radius, the radial expansion is given by Equation 1.

$$r_i = r_0 \times f_i \tag{1}$$

Then, for each ring r_i, n points are distributed at the angular position θ_j. The distribution of the points along a given ring r_i is given by the expression in Equation 2.

$$\theta_j = \frac{2\pi j}{n_i} \quad \text{for } j = 1, 2, \ldots, n_i \tag{2}$$

The conversion of polar coordinates (r_i, θ_j) into Cartesian coordinates (x_{ij}, y_{ij}) is given by the equations presented in 3.

$$x_{ij} = r_i \cos(\theta_j), y_{ij} = r_i \sin(\theta_j) \tag{3}$$

There are two parametric conditions defined to stop the radial distribution process:

- if $r_i > R$, wherein R corresponds to the radius maximum limit;
- if (x_{ij}, y_{ij}) is outside the containment polygon.

Regarding the noise-based distribution, as proposed in Algorithm 2, a specified number N of random points is generated within a polygon P, ensuring that each point maintains a minimum distance d from all previously generated points. The final output is a list of points that are both within the polygon and properly spaced apart.

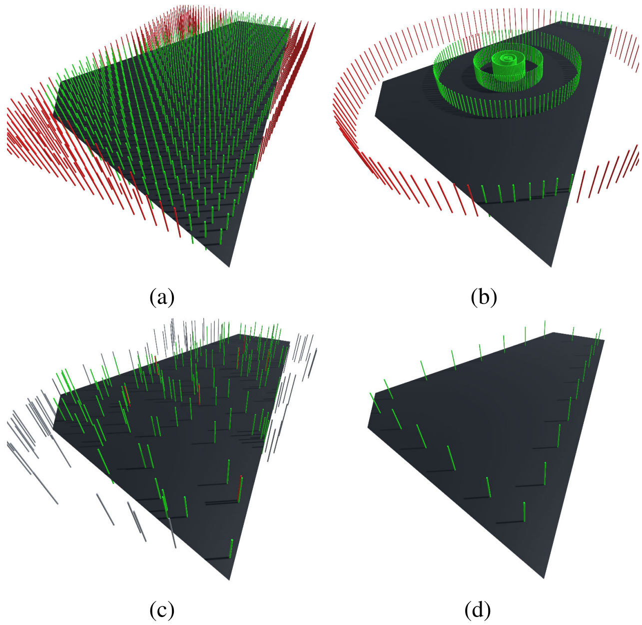


FIGURE 3. PROMORE distribution approaches: a) grid-based; b) radial-based; c) noise-based; and d) border-based.

The border-based distribution method involves placing points along or near the edges of a polygonal parcel. This approach is particularly useful for positioning elements such as buildings along access roads. Given an offset distance O from the polygon's margin and a spacing value Sp between added points, the following steps are carried out to consolidate the application of this method:

- initialize a list I to store points as empty;
- extract the polygon's ground-plane representation (assuming the xy-plane).
- determine the bisector angles (β) associated with the polygon vertices $V_{i..n}$, where each vertex sequentially serves as a pivot V_i over n iterations, joining two consecutive polygon edges, V_i, V_{i-1} and V_i, V_{i+1};
- replicate the vertices and, based on the previously calculated β values, apply an offset O to evenly scale down their area, by resorting to the equation $V'_i = (V_{i_x} + O\cos(\beta_i), V_{i_y} + O\sin(\beta_i))$, where V'_i represents the transformed replica of the original vertex and β_i corresponds to the bisector angle associated to V'_i.
- for each line L formed by V'_i and V'_{i+1}, intermediary points $V''_{j..k}$ are progressively placed at a distance of Sp from each other, following a process that continues until the cumulative length of the next $V''_{j..k}$ to add exceeds the length of L. While in the beginning of the intermediary points extension stage V''_j is initialized with V'_i and appended to the I list, each iteration assumes the following behavior:
  - a new V'' is placed resorting to the equation $V''_{j+1} = (V''_{j_x} + Sp\cos(\theta_i), V''_{j_y} + Sp\sin(\theta_i))$, wherein θ_i is obtained through $\arctan\left(\frac{V'_{(i+1)_y} - V'_{i_y}}{V'_{(i+1)_x} - V'_{i_x}}\right)$;

  - if the cumulative length of added points V'', including the contribution of the newly computed V''_{j+1}, surpasses the length of L, then V''_{j+1} is discarded, and the loop is terminated – coinciding with the moment that k is reached;
  - otherwise, the V''_{j+1} is added to the I list, V''_j is reassigned to V''_{j+1}, and the intermediary points extension stage continues for another iteration;
- in the end, the list I containing a set of valid V'' is returned as result of the border-based distribution method.

Figure 3 depicts graphically the aforementioned distribution approaches, for assets' positional booking purposes.

The topographic alignment of the virtual assets with the terrain can be achieved based on simple pivot-based height adjustment or resorting to physics-based constraints. The former uses ray-cast collisions with the terrain to determine the height (y value) at which objects should be placed. This method is particularly effective for objects with minimal ground contact or those with root-like structures, such as trees. In contrast, the physics-based approach allows objects to fall from a predefined height – higher than the maximum topographic y value – until they naturally reach equilibrium with the terrain. This technique is especially suitable for objects requiring larger contact supports, such as buildings.

3) PLACEHOLDERS POSITIONAL BOOKING

The PROMORE encompasses a strategy using placeholders to reserve space for different types of objects, including trees, buildings, farm machinery, and animals. Another purpose of this approach is to probe the integration coherency of the virtual assets to be incorporated, more specifically, through the assessment of collisions and aberrations resulting from the use of physics-related functionalities (see Section III-B2). If a collision or lack of stability is detected for a newly added placeholder, it is promptly removed, ensuring that all assets will be able to coexist properly spaced and free from overlap. For components that are mutually penetrable, such as virtual tree canopies, collision tests are omitted, as their merging reflects real-world behavior.

4) INCREASING VISUAL FAITHFULNESS

A step to increase the visual fidelity is performed after the distribution of placeholders, consisting in their replacement with more detailed assets that resemble their real-world counterparts. Finally, complementary assets with lower priority are placed to complete the virtual scene. These assets primarily consist of smaller objects with widespread distribution but lower visual impact from distant viewpoints, such as grass, bushes, and flowers.

5) POST-MODELING DATASETS ACQUISITION

Once the virtual scene is fully assembled, datasets are ready for acquisition. PROMORE encompasses functionalities for the user-manageable automatic extraction of annotated data through virtual camera flights that simulate RPAV operations, employing a multi-patch acquisition strategy for comprehensive coverage. While the user can select the data acquisition height, the preferred annotation mode, and filter the classes of objects to capture, for each patch, PROMORE resorts to virtual objects' bounding-box and visibility features, shadow toggling (on/off) and background subtraction methods to retrieve fine-tuned annotations. These virtual campaigns capture and process both RGB data and corresponding annotation sets across the terrain.

More specifically, two dataset acquisition modes are supported, based on: a) segmentation; and b) object detection. Both integrate illumination and shadow switching features to manage the influence of light-related conditions over the virtual environment.

The segmentation mode aims at creating semantic masks for groups of objects belonging to the same type. To that end, for each group to annotate, shadow and local light sources are turned off, except global illumination. Then, foreground and background imagery are set up with the objects of interest hidden and visible, respectively. Next, both images are subtracted to retain only the given group under annotation, whose pixels are transformed to become white, according to the mathematics fundamentals of Equation 4:

$$F(x, y) = \begin{cases} 1 & \text{if } |I(x, y) - B(x, y)| > T \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where:

- $F(x, y)$ is the binary foreground mask at position (x, y).
- $I(x, y)$ is the intensity of the current frame at position (x, y).
- $B(x, y)$ is the background model at position (x, y).
- T is the threshold value.

After minor morphological operations – to fill holes in the middle of the white blob, for example –, this mask image is stored, constituting the annotated data.

In object detection mode, the focus shifts to identifying and annotating multiple instances of each group of virtual objects. Similar to segmentation mode, for each instance to handle, local lights and shadow settings are turned off. To better manage computational load, the far corners of the bounding-box for the virtual object being processed are determined. Following this, a foreground/background imagery composition operation – previously described in the segmentation mode – is applied. The coordinates of the bounding-box are then converted from the world coordinate system to the screen reference, enabling the adequate cropping of both foreground and background images, which are subtracted to isolate the specific instance being annotated. The resulting image portion with a narrowed search space is then binarized, as defined by Equation 4, to fine-tune the size of the rectangle used to annotate the virtual object in focus. Finally, the annotation elements – i.e., rectangle and label identifying the virtual object type – are saved for an appropriate file structure and format.

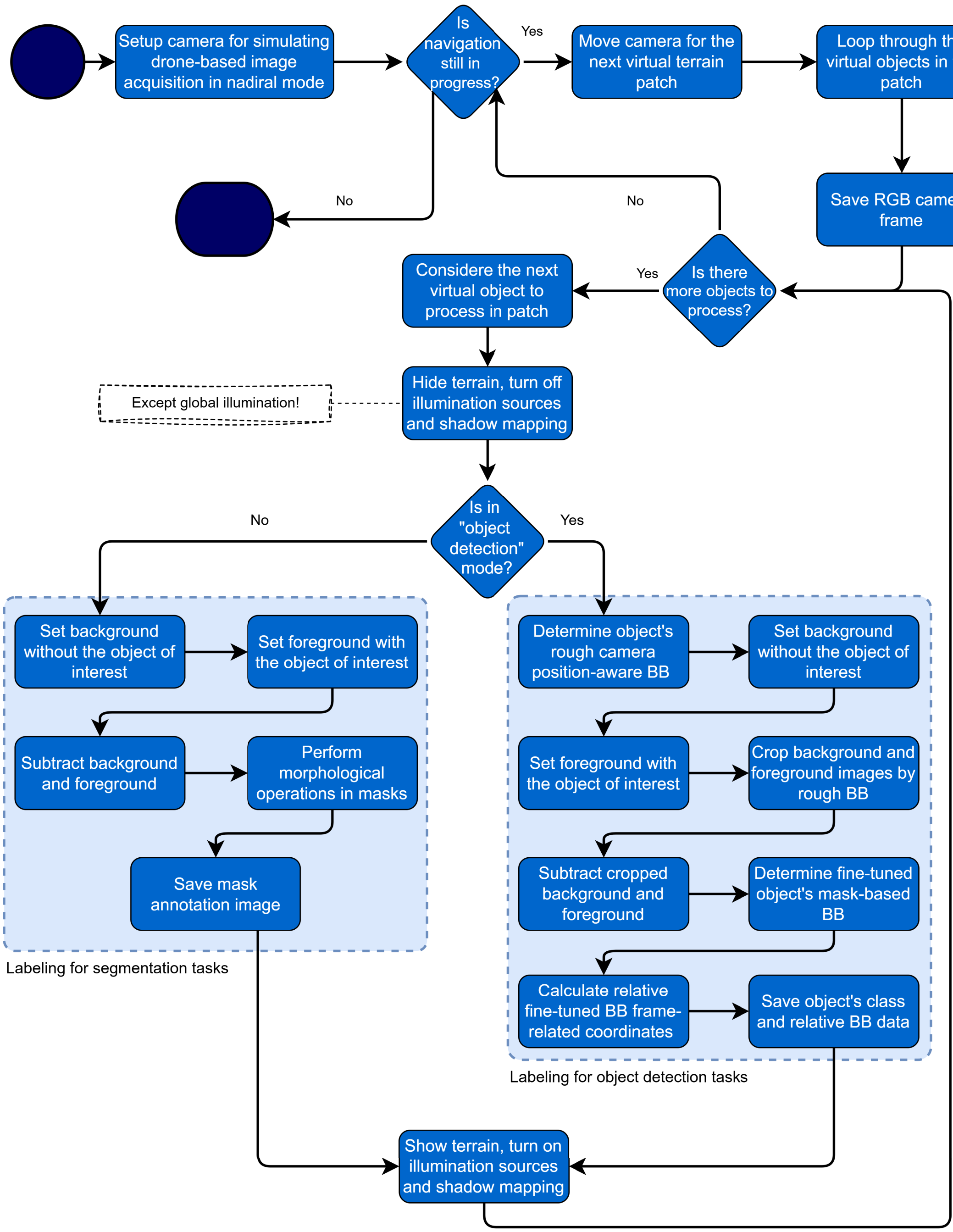


FIGURE 4. Datasets acquisition process, illustrating segmentation and object detection modes. BB stands for "bounding-box".

Regardless the mode, light and shadow settings are reset in the end of each annotation. Figure 4 graphically sums up the proposed methodology for both modes.

Next section will address the PROMORE implementation, based on the previously addressed specifications.

IV. PROMORE IMPLEMENTATION

This section is dedicated to the implementation of PROMORE, including critical aspects such as the involved hardware and software supporting tools, assets libraries, and the consolidation of the pipeline proposed in the previous section.

A. HARDWARE AND SOFTWARE DEVELOPMENT ENVIRONMENT

PROMORE was developed in a setup with the following specification:

- Processor: 11^{th} Generation Intel® Core™ i7-11800H @ 2.30GHz (Intel Co., CA, United States of America);

- Random Access Memory (RAM): 32GB @ 2933MHz SODIMM (Corsair Gaming, Inc., CA, United States of America);
- Graphic Card: Nvidia® GeForce RTX 3080 (laptop edition), with 16.0GB of VRAM GDDR6 (Nvidia Co., CA, United States of America);
- Storage: 1TB, 3500MB/R, 3300MB/W (Samsung Electronics Co., Ltd., Suwon, South Korea);
- Operating System: Windows 10 Home 64 Bit (Microsoft Co., WA, United States of America).

PROMORE was developed resorting to Unity Engine (Unity Software Inc., CA, United States of America), which served as the main implementation platform. It combines a powerful graphical user interface with a handy and functional scripting compiler supporting component-oriented programming. Moreover, Unity comprises a vast set of productivity features, such as audio, video, rendering, physics, among many other elements attachable to instances of type *GameObject* – a visualizable and interactive key entity managed by the framework. Supported by an exhaustive asset store [43] and compatible with Github [44], Unity enables to safely import plugins developed by the community. Having that said, the (internal or external) tools used for productivity inside Unity's environment that are worthy to highlight are the following:

- Unity Editor's Terrain [45], which offers many tools for manual or code-based manipulation, such as brushes for texturing and for inducing a height profile, as well as collision-sensitive components.
- Komietty's Github project on Unity-compatible Voronoi [46], which integrates utilitarian code to strip an terrain-matching area into parcels.

B. ONTOLOGY-BASED PARAMETERIZATION–CONDITIONING THE PROCEDURAL MODELING STOCHASTICITY

To align with the guiding ontology defined earlier (see Section III-A), PROMORE manages to articulate between a serializable/deserializable object-oriented (OO) class system and a JavaScript Object Notation (JSON) archive, which specify terrain features and parcel types declaring roles and probabilities of incorporation. Each parcel aggregates a list of ontological groups and/or objects representing detailed/final virtual assets, organized according to a hierarchical file and folder structure – referred to as the library-driven ontological data model –, loaded recursively to flexibly accommodate different configurations. The referred elements can be fully edited and parameterized through a graphical user interface (GUI), implemented with a standard Unity's library [47] and depicted in Figure 5.

When the environment settings are being specified by an user, for each newly declared parcel type, an extensive editable list of ontological groups and objects are loaded, fully reflecting the encapsulation established by the library of final assets. Such groups and/or objects can be added or removed at user's will. Moreover, each object and group can be parameterized to establish boundaries in PROMORE's action during the generation of the virtual rural environment. A comprehensive list of the main parameters that can be edited per ontology-based configuration is provided in Table 1, resorting to the OO class system that is managed by PROMORE in articulation with a respective JSON file for permanent data storage and retrieval.

It should be noted that the proposed GUI not only provides controls tailored to each parameter type (e.g., ranges are represented using pairs of sliders) but also addresses specific requirements to ensure parameterization coherence. For example, since `random_rate` and `random_absl` are mutually exclusive, adjusting the sliders for one automatically sets the sliders for the other to 0, and vice versa. Finally, all of the changes performed by the user can be saved upon request for future access, either for modeling purposes or for the redefinition of settings.

C. TERRAIN, PARCELING, ROLE ASSIGNMENT, AND ROAD NETWORK

As previously mentioned, Unity environment provides a *GameObject* for utilizing and manipulating virtual terrain features, equipped with a wide array of parameters and tools. By combining this component with scripting, specific parameters can initially be adjusted to configure the terrain's dimensions and surface oscillation. To perform terrain parcel splitting, two elements are required: a ground-level rectangular area and a Voronoi layout builder [46]. Based on the number of Voronoi seeds, defined by the user, the algorithm tessellates a set of closed-loop lines over the specified rectangular area, which is then divided into several polygons sharing common edges. These polygons are assigned roles, according to a given ontology-based specification. Valid examples of roles are forestry population, human habitation and viticultural exploration. The next step involves road highlighting. By integrating terrain features – specifically brushes texture and radius – and scripting, roads can be outlined based on the polygons' edges and the terrain's height at each point of passage.

D. VIRTUAL ASSETS–PLACEHOLDERS AND DETAILED MODELS

PROMORE supports a diverse range of assets designed for rural environments, encompassing two-stage prefabs relying in rough placeholders and handcrafted detailed models. The rough placeholders are created using a Unity's primitive or a combination of them, serving as basic representations. Additionally to booking scene space, these preliminary assets enable a series of coherence-checking operations, ensuring consistency and preparing the groundwork for integrating the final assets, which, in turn, represent detailed models and may originate from diverse sources. For instance, the ones associated to trees and vine plants featured in the current version of PROMORE are self-authored. In contrast, low vegetation models such as grass and bushes were acquired

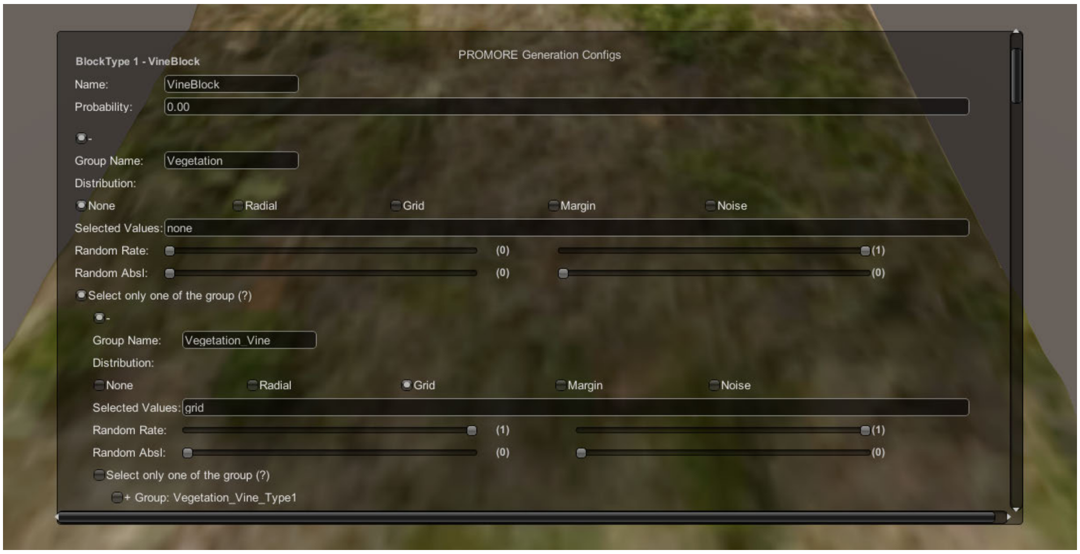


FIGURE 5. Configuration environment menu of PROMORE.

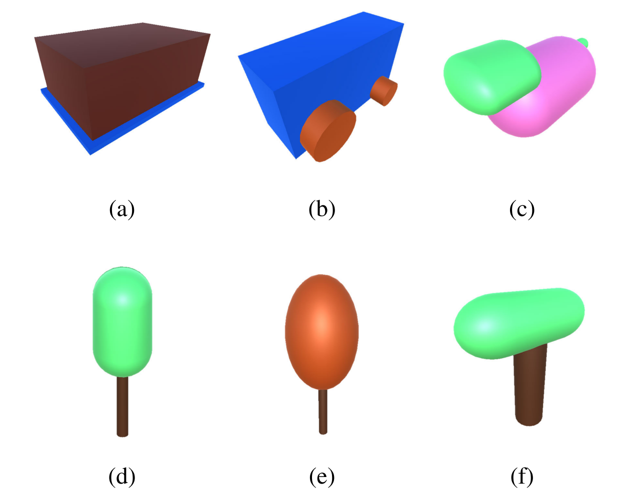


FIGURE 6. Placeholder prefabs: a) building; b) machinery; c) animal; d-e) trees; and f) vine plant.

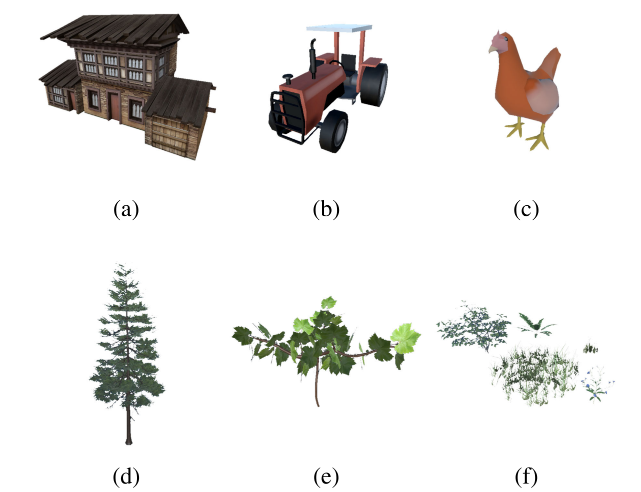


FIGURE 7. Examples of final prefabs: a) rural house (building); b) tractor (machinery); c) chicken (animal); d) pine (*Pinus pinaster* tree); e) vine plant; and f) low vegetation.

from CGTrader [48]. Similarly, models representing rural buildings, animals, and farm machinery were downloaded from the Unity Asset Store [49], [50], [51], where they were made freely available. The accommodation and organization of the final detailed assets within a folder/file system constitute a library that drives the configuration of ontology-based instances (see subsection IV-B). As one of the parameters that configures PROMORE's ontological data model, `priority` distinguishes the importance of the assets for the modeling activities. Lower priority assets can be seen as complementary final models – typically, with a widespread distribution and low visual impact at far perspectives – that, however, do not depend on prefabs preliminary placement.

1) TWO-STAGE PREFABS FOR SETTING UP VIRTUAL SCENARIOS

Placeholders consist in groups of Unity primitives that are converted into prefabs, as exemplified in Figure 6. The main goal of these primary assets is to attain templates that could roughly match real-world counterparts. Placeholders prefabs utilization is twofold: (i) to constrain the sizes of the visually faithful assets modeled in external tools and which will take place in the more advanced stages of the PROMORE operation, avoiding discrepant or awkward dimensions in the final environment composition; and (ii) to anticipate collisions or spawning-related inconsistencies before final virtual objects' placement. To facilitate collision checking,

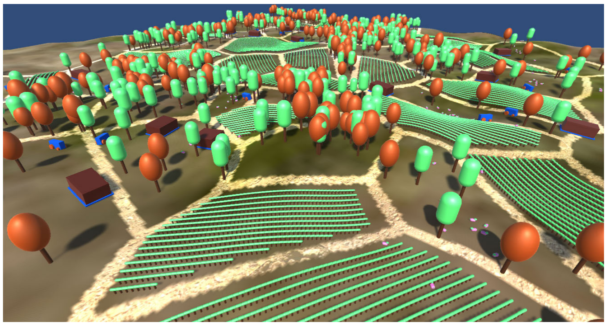
TABLE 1. Documentation summary of PROMORE's ontology-related classes and fields.

Class	Purpose	Fields
Entity (abstract)	Serves as a base class providing unique identification and ontology-related naming for derived classes.	`ID`: A unique identifier for each entity, generated using a GUID. `name`: A human-readable ontology-related name for the entity.
Object (inherited from *Entity*)	Represents a tangible item or asset with placement and transformation properties.	`path`: Path to the file representing the object. `distribution_pattern`: Spatial distribution pre-booking rule (random, radial, grid, border). `dist_alpha_variation`: Range for distribution/spacing in one of the axis ("alpha"). `dist_beta_variation`: Range for distribution/spacing in another axis ("beta"). `random_scale_variation`: Range for randomly varying the object's scale. `random_rotat_variation`: Range for randomly varying the object's rotation. `random_rate`: Percentage-related range of probability for integrating the object in a given terrain parcel – used for a pre-booked positions set and in exclusion of the `random_absl` parameter. `random_absl`: Absolute range determining the number of objects required for integration in a given terrain parcel – in exclusion of the `random_rate` parameter. `requires_physics`: If true, object placement occurs by physics-managed drop. `priority`: Establishes a processing precedence.
Group (inherited from *Entity*)	Represents a collection of objects or nested groups, enabling hierarchical organization.	`distribution_pattern`: Spatial distribution rule for randomly selected objects within the group. `random_rate`: Group/inner objects probability range for inclusion in a given terrain parcel - used in exclusion of the `random_absl` parameter. `random_absl`: Absolute value range for random inclusion of the group/inner objects in a given terrain parcel – used in exclusion of the `random_rate` parameter. `select_only_one_of_the_group`: If true, limits selection to one child group or object within the group. `childs`: Array of nested `Group` instances (obeying to the final assets library organization). `objects`: Array of `Object` instances directly contained within the group (obeying to the final asset library organization).
ParcelType (inherited from *Entity*)	Represents a type of parcel populating the *TerrainSet*, containing a set of groups and a placement probability.	`probability`: Likelihood of assignment to a terrain portion. `groups`: Array of `Group` instances associated with this parcel type.
TerrainSet (inherited from *Entity*)	Represents a terrain with dimensions, terrain-specific properties, and associated parcel types.	`voronoiSeeds`: Seeds number for generating a Voronoi-based layout. `width`: Width of the terrain in units. `height`: Height of the terrain in units. `min_height`: minimum y-axis value for terrain elevation. `max_height`: maximum y-axis value for terrain elevation. `parcel_types`: Array of `ParcelType` definitions.

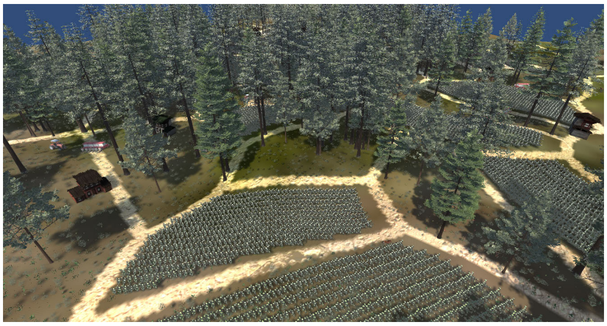
each prefab needs to be modeled integrating "bulk" collider component.

According to the ontology-based specification, some objects may resort to physics settings to ensure a more natural adjustment to the terrain's topography, after being dropped from a certain height. In case of stabilization failure, associated to a jitered behavior lasting more than 3 seconds, the newly added placeholder is removed from the scene. The same applies when an undesired collision with another object is detected. This verification process targets structures with large contact bases, such as buildings. To that end, it is important to note that in the ontology-based specification, the `requires_physics` parameter must be set to `true` for all objects requiring physics-managed drops.

In the more advanced steps of the procedural modeling pipeline, prefabs resulting from handmade virtual models with higher visual fidelity – exemplified in Figure 7 – replace their respective representative placeholders. To that end, a semantically-based association is considered. Such process highly contributes for the overall enhancement of realism of the scene. With influence over all of the final/detailed prefabs, the ontology-based specification provides additional



(a)



(b)

FIGURE 8. Virtual rural environment modeling stage transition: from a) placeholders population to b) replacement with higher priority final prefabs.

controlled randomness for scale and rotation operations, improving the overall realism of the resulting virtual environment.

Figure 8 depicts the progress of a virtual rural environment modeled by PROMORE, from the placeholders population stage to the replacement using the final detailed prefabs.

It should be noted that environmental aesthetics depend not only on the quality of the final assets – i.e., the most detailed ones – but also on Unity's lighting systems, which enhance the scene through global illumination, directional lighting, and context-aware shadows simulation.

While season-based representations are not explicitly addressed, PROMORE's library can be populated with virtual representations of vegetation in regard to the different periods of the year. Incorporating phenological development knowledge (e.g. inspired in BBCH-scale [41]) is recommendable for the sake of scientific accurateness, considering specie-specific characteristics. As in this version of PROMORE the primary focus is to showcase representative virtual environments generation capabilities and automatic datasets extraction, virtual assets simulating developed vegetation were encompassed to emphasize unique visual insights associated with advanced phenological stages essential for key RS-related monitoring tasks, such as yield estimation [52].

2) LIBRARY OF FINAL ASSETS–FLEXIBLE GROWTH SUPPORT AND ONTOLOGICAL EXPANSION

The file system comprising the detailed virtual objects forms PROMORE's final asset library, which is dynamically utilized to drive data loading for assisting the user in the configuration of the PROMORE generation settings. As such, users can define groups and objects according to desired land use while fine-tuning their stochastic properties. Furthermore, the library can be flexibly expanded through a straightforward, step-by-step workflow currently performed in the Unity Editor.

When adding a new detailed virtual object of higher priority to the library, it is essential to verify whether a representative placeholder already exists, as it plays a key role in ensuring coherence during PROMORE's preliminary asset placement. If not, a new one needs to be created using Unity primitives, with dimensions closely aligned to real-world measurements, left at the criterion of the human configurator. This placeholder must include one component named "bulk", set up to be collision-aware, for PROMORE's handling purposes. Then, the placeholder is stored as a prefab within a dedicated library for placeholders. As this placeholder serves as a metric-based reference for subsequent steps, its materials can be temporarily set to transparent to assist in aligning the detailed/final virtual model within the defined spatial limits. Afterwards, the final virtual model, either manually produced or procedurally generated, is selected and imported from an external repository and manually translated, rotated, and scaled to achieve a satisfactory alignment with the placeholder. Once aligned, another step involving cleaning up the final model must be carried out, with the aim of setting it free from the placeholder. At that point, the final model is ready to be stored in the library, expanding the pool of assets available for composing virtual environments, with repercussions to the parameterizable ontological data model. Figure 9 provides a visual insight regarding the described process.

E. INTERACTIVE DATASETS ACQUISITION

At the moment that a virtual rural environment is ready, PROMORE offers a navigation mode that allows the user to control the camera in nadiral mode, mimicking RPAV's image acquisition settings during remote sensing missions. Using the *WASD* keys, the user can move the camera north, south, east, and west to position the viewport optimally. By pressing the *K* key, a screenshot is taken along with a series of annotated data. Moreover, segmentation and object detection modes can be selected, by pressing *M* key, which toggles among both. Additionally, the user can cyclically iterate through various heights (distances from the terrain in nadir mode – 50, 75, 100, 125, 150, 200 m) to refine the elevation for shots, ensuring comprehensive coverage and accurate data acquisition, by using the *H* key. As for the selection of specific virtual object types to capture, currently, PROMORE supports the interpretation of a text file that, to take effect, must be placed within the root of the assets library folder. Such file must be filled with a comma-separated list of complete or partial class names corresponding to generated assets of interest for which processing is required during the dataset extraction stage. In the absence of such filtering

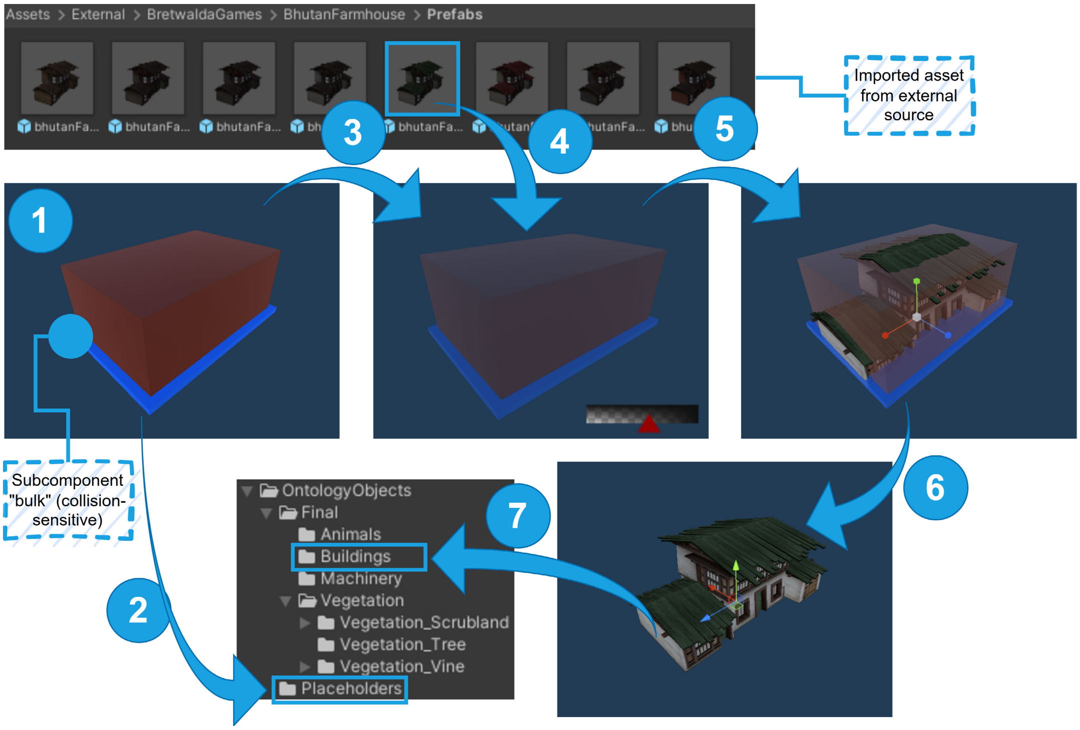


FIGURE 9. **PROMORE's virtual models integration process: (1) create a placeholder with real-world dimensions, (2) store it as a prefab, (3) make the placeholder transparent for alignment, (4) import the final 3D model, (5) adjust the model's position, rotation, and scale to fit, (6) clean up the model for independent use, (7) save it in the final prefab library.**

guidelines, all virtual objects associated to the library-driven ontological data are considered eligible for undergoing through the dataset extraction process. For a complete and automatic RPAV flight mission-like acquisition, the user can press *P* key.

To acquire annotated data for each terrain patch during manual or automatic RPAV-like flights, all of the consolidated (final) virtual objects are iterated and grouped based on names that completely or partially match the filtering criteria specified in the aforementioned text file located at the root of the ontology-driven library. This grouping process involves splitting the text file content by commas to obtain filtering tokens that are then compared with the names of the virtual objects to identify matches and map them into an associative dictionary. Subsequently, another operation is performed over the filtered objects to identify the ones within the camera frustum, making them eligible to undergo through the patch-wise annotation process, according to the selected mode. Regardless of the active mode, the first step involves capturing and storing a plain RGB image of the frame.

For the semantic-oriented segmentation mode, with light sources and shadows deactivated and the terrain temporarily occluded by layer reassignment, two images are captured for each object group to process: one with the objects of interest (foreground) and another without them (background). By performing background subtraction, masks for the objects within a given group are generated and stored along with the clean RGB image.

As for the object detection mode, beyond acquiring the plain RGB frame, an instance-based bounding-box detection is conducted for each virtual asset to analyze. Therefore, considering a single iteration, light sources and shadows are disabled, and the terrain is temporarily occluded via layer reassignment. Using the object's bounding box — whose accuracy is influenced by the camera's position relative to that object — the image is cropped. Foreground and background images are then generated and subtracted, as in the segmentation mode. Subsequently, the region of interest (ROI) can be fine-tuned within a smaller search space of non-black pixels. Finally, the object's class information is stored, along with YOLO-compatible ROI coordinates, considering necessary adjustments for the cropping operations performed relative to the full frame.

For both modes, shadowing and terrain occlusion settings are restored before processing the next object or group of them.

Figure 10 illustrates the results of an annotation operation conducted in segmentation mode for three object types.

The next section, focuses in the tests and results made to PROMORE, covering functional effectiveness and performance.

V. TEST AND RESULTS

To evaluate the PROMORE solution, a series of performance and functional tests were carried out. To that end, an asset library was set up for PROMORE, with several prefab

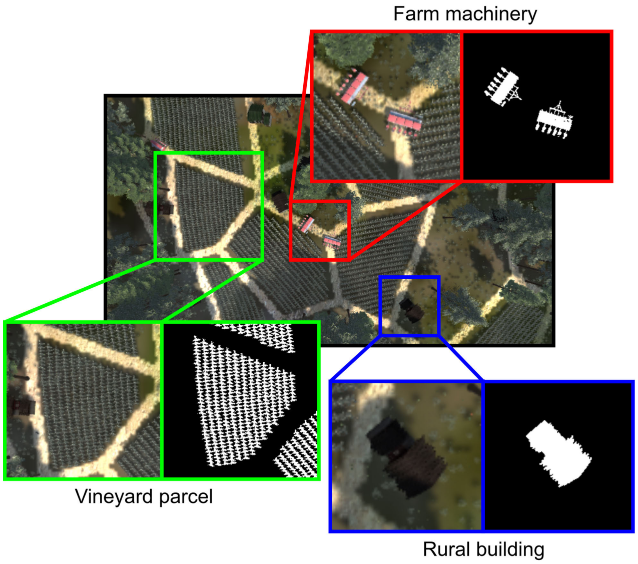


FIGURE 10. Example of imagery acquisition for three object types: farm machinery, rural buildings and vine plants. Clean images and segmentation mode-based masks are collected for each group of elements within the camera frustum.

instances, which are generally characterized in Table 2. Then, an ontology-based specification was set up, assigning the following probabilities to parcel roles: 30% for forestry areas, 40% for vineyard crops, and 30% for human occupation. These parcels were parameterized according to the following: forestry areas were associated with groups of trees; vineyard parcels were linked to vine groups; and human-occupied parcels were assigned with buildings, animals, machinery and a smaller proportion of trees compared to forestry areas. Transversely, all of the parcel types were configured to encompass lower priority assets, more specifically, low vegetation. An overview of this specification is provided in Table 3, serving as the common baseline for 4 replicas, each one redefining Voronoi seeds to 20, 40, 60, and 80 points, with the goal of varying landscapes representations within a virtual terrain area of 512 $\times$ 512 Unity units. Such replicas were made undergo through to the PROMORE generation process, producing a set of 4 respective rural-like virtual environments, whose characteristics are summarized in Table 4.

From these results, it can be inferred that the parceling stage is very fast, lasting less than 0.5 s in all cases. The operation time for the road decal step shows a correlation with the number of generated parcels. The placement of lower detail objects is, on average, more time-consuming than the placement of detailed models, likely due to the time required to check building stabilization after being dropped on the terrain (3s), which induces additional delay. Nonetheless, PROMORE did not take longer than 94 s to fully generate a virtual rural environment with 50 parcels (blocks) and a total of 88,160 processed prefabs. One should note that, despite the ontology-based specification defining random probability intervals for generating different virtual object types (Table 3), some final counts fell below the minimum expected values due to anti-overlapping collision operations (as explained in Section IV-D1). For instance, as shown in the first data row of Table 4, across the 8 generated blocks, only a single *Building* instance was produced, whereas the minimum expected was two, more specifically, one for each of the designated human occupation parcels – derived from the 30% assignment ratio for human occupation, as defined at the beginning of this section, which results in ≈ 2 of these parcels.

In Figure 11, visual results of a virtual rural environment generated by PROMORE are presented, showcasing primarily parcels allocated for human habitation and farming machinery anchorage, as well as land designated for viticultural activities.

To evaluate the functionality of both dataset acquisition modes, another ontology-based configuration was defined, setting 60 Voronoi seeds in the terrain settings while retaining the previously described parcel specifications. Using this configuration, PROMORE was requested to generate a rural environment. The resulting virtual artifact was then used to acquire datasets compliant with segmentation and object detection purposes. To that end, automatic virtual camera-based nadiral flights were conducted at two distinct altitudes: 50 m and 125 m.

Regarding the extraction of segmentation-compliant datasets, a visual result of a sample of this process is illustrated in Figure 12, which demonstrates some classes annotated across a set of terrain parcels, densely populated with vineyard representations, but also, integrating other elements such as varied tree species and rural buildings.

Table 5 presents quantitative data for instances of *Animals*, *Buildings*, *Machinery*, *Tree*, and *Vine*, detailing the number of objects per group, the area of the generated masks in pixels, and the acquisition times. The results highlight that closer proximity during dataset acquisition enhances detail as expected and, therefore, increases the number of patches, which in turn leads to longer processing times. A roughly proportional relationship is observed between the number of patches and processing duration. However, when the object count is relatively low (below 90), it significantly influences processing time, surpassing the impact of mask size. Conversely, as the number of objects grows (exceeding 300), the mask size becomes the dominant factor, exerting a greater influence on processing time than the object count.

Figure 13 showcases partial results from a YOLO-oriented dataset exported from PROMORE, designed for object detection tasks. The ROIs and labels for *Animal*, *Machinery*, *Building*, and *Vine* instances are displayed in the Roboflow tool [53], following the successful loading of YOLO-compatible annotation files.

The results presented in Table 6 for the YOLO dataset indicate that increasing the altitude by a factor of 2.5 leads

TABLE 2. Characterization of the virtual models used in PROMORE's asset library.

Asset Family	N. Original Instances	N. Mean Triangles (Mesh)	N. Mean Vertices (Mesh)	Textures/Normals Mean Complexity
Vine	4	49818	124090	88 MB
Tree	4	193224	404521	66 MB
Building	8	2578	4972	27 MB
Animal	3	1239	1274	788 KB (Single color pallet)
Machinery	2	5229	6279	10 MB
Low vegetation	9	546	544	4 MB

"N." stands for "Number of".

TABLE 3. Ontology-based specification proposal for testing PROMORE.

Asset Family	Vine Parcels		Forestry Parcels		Human Occupation Parcels		Priority	S. Mode
	Dist.	Rand.	Dist.	Rand.	Dist.	Rand.		
Vine	Grid	$R_N(3,5) \times R_N(1.5,2)^*$	-	-	-	-	0	Single
Tree	-	-	Noise, grid**	$R_N(5,30)$	Noise	$R_N(5,10)$	0	Multiple
Building	-	-	-	-	Border	$R_N(1,3)$	0	Multiple
Machinery	-	-	-	-	Noise	$R_N(1,2)$	0	Multiple
Animal	-	-	-	-	Radial	$R_N(10,20)$	0	Multiple
Low Vegetation	Noise, grid**	$R_R(0.6,1)^{**}$	Noise, grid**	$R_R(0.6,1)^*$	Noise, grid**	$R_R(0.6,1)^{***}$	1	Multiple

General notes: "Dist." is the column wherein is identified the distribution(s) strategy(ies) for objects' spreading/placement (among noise-, grid-, radial- and/or border-based). "Rand." specifies the column wherein random functions (R_N or R_R) are defined towards the absolute or relative instantiation of assets. "R_N" represents the function that randomly picks an absolute number of virtual elements from a given interval (minimum, maximum). "R_R" consists in a function for selecting an aleatory ratio value from a given interval ($Interval_{min} >= 0$, $Interval_{max} <= 1$), which is then used in articulation with an absolute reference (e.g. 1000 generated points). "S. Mode" stands for "Selection Mode" and determines whether a virtual object to be integrated should belong to an exclusive sub-type (Single) or can be selected from a range of available sub-types (Multiple); for example, while a single vineyard parcel is typically populated by a fixed variety of vine plants, in the case of trees, multiple species can coexist within the same shared terrain portion.

* Random values assigned to spacing positional references within rows, as well as to define inter-row distances in a parcel, based on a grid-based distribution approach.

** Whenever multiple distribution approaches are available for instantiating a given virtual asset family in a specific moment and iteration, only one is randomly selected and applied.

*** Random value defining the cover rate of virtual assets over, using either a grid-based approach with positional references spaced at 1.5×1.5 Unity units or a noise-based dispersion with 1,000 points projected onto the *xz*-plane, while ensuring compliance with the parcel's polygonal boundaries.

TABLE 4. PROMORE performance tests made with different Voronoi seed numbers. Columns 2 to 7 conserve virtual objects' generation order.

Voronoi Seeds	N. Blocks	N. Vine Plants	N. Trees	N. Buildings	N. Machines	N. Animals	N. Low Vegetation Plants	T. Parceling (s)	T. Road Decal (s)	T. Lower-Detail Population (s)	T. Higher-Detail Population (s)
20	8	9878	97	1	1	13	50202	0.26	1.29	4.91	5.39
40	20	10783	236	8	4	36	64154	0.37	2.35	16.55	13.96
60	33	11906	362	14	12	103	78317	0,52	3,24	37,28	21,78
80	50	12795	568	18	13	163	74603	0,37	3,98	57,04	33,09

"N." stands for "Number of"; "T." means "Time for".

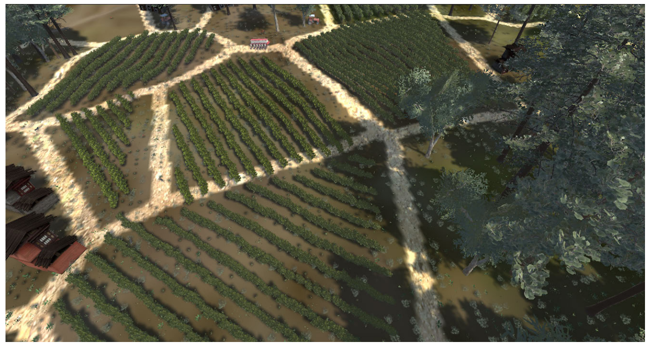
TABLE 5. PROMORE's performance in generating masks, considering both 50 m and 125 m of altitude.

Instance name	Amount	Altitude = 50 m, patches = 45		Altitude = 125 m, patches = 8	
		Mask (#px)	Time (s)	Mask (#px)	Time (s)
Animal	82	2,759	3.6	262	2.05
Building	9	103,729	2.6	10,765	1.94
Machinery	10	23,259	3.04	1,634	1.58
Tree	310	3,717,774	19.63	622,033	5.5
Vine	7,031	2,997,250	10.04	586,205	3.21

to an approximate 3.4-fold increase in processing time. This suggests that the instance-wise nature of YOLO's annotation significantly affects computational efficiency of PROMORE in exporting datasets. Furthermore, as expected, there is a clear correlation between the number of instances within a category or class group and the processing time, with larger groups requiring more time to calculate the ROIs for each instance.

A short discussion regarding the performances of PROMORE will be provided in the next section.

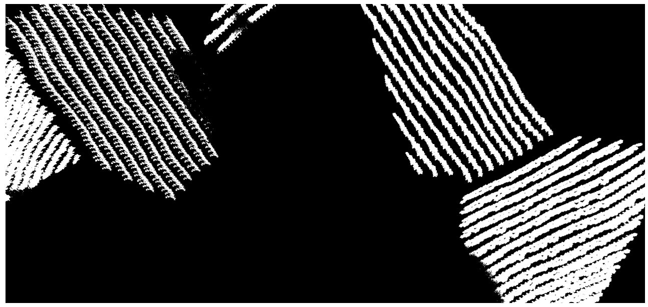
(a)



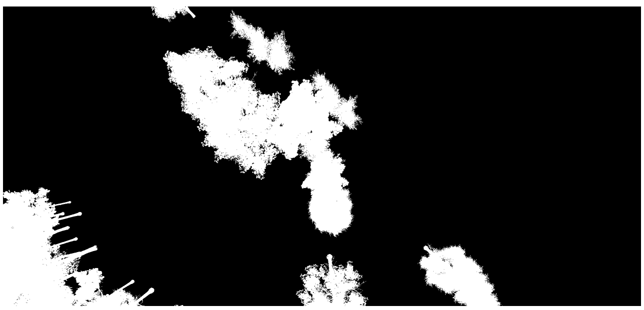
(b)

FIGURE 11. Virtual rural environment generated by PROMORE: a) focuses parcels with roles related to human habitation/storage purposes; and b) showcases a set of anthropogenic vine parcels.

(a)



(b)



(c)

(d)

FIGURE 12. Segmentation-based dataset extraction considering a set of terrain parcels generated by PROMORE: a) shows the original image; b) presents the masks for vine plants; c) displays the masks for trees, including *Abies alba* and *Pinus pinaster* species; and d) highlights the masks for rural buildings.

TABLE 6. PROMORE's performance in generating YOLO-based annotations, considering both 45 m and 125 m of altitude.

Instance name	Amount	Altitude = 50 m, patches = 45 Time (s)	Altitude = 125 m, patches = 8 Time (s)
Animal	82	1.49	6.6
Building	9	0.37	0.64
Machinery	10	0.36	1.06
Tree	310	81.02	41.04
Vine	7,031	124.54	649.73

VI. DISCUSSION

Creating DL datasets for RS is a labor-intensive, time-consuming and error-prone activity, involving extensive field acquisition campaigns, burdensome post-mission processing, and dull manual labeling and annotation procedures. Besides, data augmentation is a widely adopted practice typically conducted prior to the model training stage, aiming to achieve improved inference capabilities. However, whether implemented through traditional methods or GAN-based approaches, this practice heavily relies on pre-existing

(a) (b)

FIGURE 13. YOLO-based annotations automatically extracted from PROMORE and imported to Roboflow [53], for visualization purposes: a) is showing a human exploration parcel; and b) is focusing the partial areas of a couple of vineyard parcels.

examples – i.e., imagery from the training dataset – thereby introducing a potential risk of oversampling [8].

In this context, procedural modeling may constitute a powerful alternative as a methodology to generate rule-based faithful virtual environments, with benefits for data variability. One of the valid strategies for regulating/guiding procedural modeling is through the use of ontologies, which establish context-aware relationships between several entities describing a given system (e.g. a rural landscape). Along with operational guidelines, either implemented and/or provided by user input, ontologies stand as key-features for ensuring the systematic generation of coherent and meaningful virtual environments. Building on these considerations, PROMORE was developed as a novel solution for generating ontology-driven virtual environments representing rural landscapes. It is capable of modeling terrains, defining and slicing parcels, placing virtual objects (e.g., vegetation), and enhancing scene's overall realism to create environments that closely resemble real-world rural settings. In addition, PROMORE fully automates the export of RS-ready datasets tailored for DL tasks. More specifically, it incorporates functionalities for simulating autonomous RPAV flights while simultaneously performing the acquisition and annotation of data regarding the generated virtual environment under analysis, supporting operation modes for segmentation and object detection.

PROMORE's efficiency is noteworthy: it can generate a rural environment comprising 50 parcels and 88,000 prefabs in approximately 94 s on a laptop like the one described in section IV-A. For segmentation datasets, PROMORE can process a simulated flight at 50 m altitude in less than 40 s, capturing data from 7442 objects across five groups, with masks containing nearly 6.84M pixels. In the context of object detection, even in a challenging scenario involving a simulated flight at 125 m altitude, PROMORE processes 7,442 instances and generates YOLO-compatible annotations in less than 12 minutes. Therefore, for segmentation purposes, under the aforementioned conditions and considering the worst-case scenarios presented, PROMORE is capable of generating, annotating, and exporting datasets for approximately 645 virtual rural environments per day (≈29,025 patches). As for YOLO-compatible dataset generation, this number reduces to a still impressive 106 environments (≈848 patches).

Considering the existing literature, it seems that PROMORE addresses a gap in the field of synthetic environment generation in a unique manner. While various studies have developed frameworks for creating large-scale synthetic datasets in different contexts, none offer the same combination of features as PROMORE, particularly its ontology-driven generation process tailored to rural, vine-dense environments and its focus on addressing RS challenges. For instance, Fedorova et al. [35] introduced a framework for generating building-related synthetic datasets utilizing custom-implemented pipelines for 3D reconstruction, with an emphasis on architecture. In the scope of general urban landscapes, Khan et al. [32] developed an automated modeling tool designed for segmentation-based datasets, incorporating manageable variables such as weather and lighting. Despite acknowledging the potential for applying this solution in RS, the focus remained elsewhere, without exploring rural or agricultural contexts. Meta-Sim2 [34], while addressing aerial and terrestrial perspectives through unsupervised learning methods and scene structure extraction via graphs, focuses on urban environments and car-driving contexts. In the realm of autonomous driving, Cosy [36] targeted the automatic generation of ground-truth annotations, specifically from an in-car perspective, further emphasizing the urban and vehicular focus. Aiming at terrestrial robotics systems, Nunes et al. [37] proposed a solution for generating RGB images, point clouds, and depth maps, focusing on alleviating the manual efforts of dataset production. While similar in its goal to streamline dataset creation, it does not cater to the specific needs of RS tasks in rural environments. Adão et al. [31] addressed the extraction datasets from a nadiral perspective, but considering rural fire detection purposes, resorting to photogrammetric-based strategies combined with particle systems for simulating flames and smoke. Therefore, PROMORE aims to contribute to the existing literature by combining ontology-based procedural generation of moderately complex rural environments with a focus on automated and accurate RS-compatible dataset extraction.

As a final note, through the diversification of virtual objects, respective combinations and arrangements, as well as background configurations, PROMORE stands as a solution for massively generating/increasing training data for DL challenges, while potentially tackling the oversampling issues highlighted in [8].

The next section ends this paper with a summary of the work done, final remarks, and proposals for future research directions.

VII. CONCLUSION AND FUTURE WORK

This paper proposed PROMORE, a procedural modeling framework specialized in producing rural-like virtual environments. The stochasticity of its generation process can be constrained and customized through user-based parameterization, more specifically, through a GUI that allows to specify ontology-driven rules. In addition, PROMORE's asset library, designed to flexibly accommodate new virtual objects, ensures scalability and ability to generate more comprehensive and diverse virtual landscapes.

Beyond its modeling capabilities, PROMORE addresses critical challenges in DL supervised learning for RS, particularly the burdensome processes of data acquisition and annotation. Supported by the proposed ontology, PROMORE automates the efficient extraction of datasets from generated rural environments, mitigating the reliance on manual labeling and mask crafting, which are often time-intensive and prone to error. This datasets gathering process harnesses the capabilities of the Unity framework, enabling the precise manipulation of environmental factors such as lighting, shadows, and object visualization toggling, towards accurate and reliable annotations. To perform imagery acquisition for setting up RS-oriented datasets, PROMORE offers functionalities to simulate nadir-view RPAV flights by autonomously inducing grid-based positional translations in the scene's virtual camera, with its frustum's view plane parallel to, but oriented towards, the virtual terrain, covering the entire generated landscape at user-defined altitudes. Unlike traditional data augmentation techniques, which are inherently constrained by the characteristics of existing imagery, PROMORE provides an approach to dataset diversification by procedurally generating entirely new scenes, contributing for tackling oversampling issues [8].

While PROMORE holds significant potential for advancing RS applications, particularly in contexts where public datasets are scarce or unavailable, there is always room for improvements. Therefore, future work will address several key enhancements with impact in the capabilities of the proposed framework. One promising direction is the inclusion of more complex geometric patterns, such as mountainous terrains, non-convex parcels, and curved distributions of objects, which would enhance the realism and applicability of the generated virtual environments. Extending PROMORE's features to support the procedural generation of individual virtual objects – complementing the current functionalities for to import them from external sources – is another key goal for future development. This enhancement aims to enable flexible representations of the continuous development of vegetation species across season-related periods, ensuring compliance with phenological growth standards (e.g. the BBCH scale [41]) at a specie-specific level. Minor missing GUIs will be implemented as well, for example, to let the user create filters for restraining the extraction of datasets to certain virtual objects of interest, instead of using a text file for that purpose.

Another noteworthy possibility involves integrating natural language processing capabilities by connecting PROMORE to a conversational large language model (LLM). This integration would allow users to define generation guidelines through requests based on human communication, which the LLM could translate into ontology-driven specifications, streamlining the parameterization process and reducing user effort. Additionally, validating the transferability of models trained on synthetic data to real-world scenarios will be crucial. This involves assessing whether synthetic datasets generated by PROMORE can produce robust inference models for real environments or, at least, serve as reliable initial sketches for active learning pipelines.

ACKNOWLEDGMENT

The authors would like to acknowledge the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia-FCT), under the projects UIDB/04033/2020, LA/P/0126/2020, GreenBotics (ref. PTDC/EEI-ROB/2459/2021), UIDB/00048/2020 and the PhD grant with reference 2021.06492.BD.

REFERENCES

[1] H. Navarro-Hellín, R. Torres-Sánchez, F. Soto-Valles, C. Albaladejo-Pérez, J. A. López-Riquelme, and R. Domingo-Miguel, "A wireless sensors architecture for efficient irrigation water management," *Agricult. Water Manage.*, vol. 151, pp. 64–74, Mar. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378377414003424

[2] A. C. Teixeira, R. Morais, J. J. Sousa, E. Peres, and A. Cunha, "A deep learning approach for automatic counting of bedbugs and grape moth," *Proc. Comput. Sci.*, vol. 219, pp. 145–152, Jan. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923002831

[3] L. Pádua, J. Vanko, J. Hruška, T. Adão, J. J. Sousa, E. Peres, and R. Morais, "UAS, sensors, and data processing in agroforestry: A review towards practical applications," *Int. J. Remote Sens.*, vol. 38, nos. 8–10, pp. 2349–2391, May 2017.

[4] L. P. Osco, J. M. Junior, A. P. M. Ramos, L. A. de Castro Jorge, S. N. Fatholahi, J. de Andrade Silva, E. T. Matsubara, H. Pistori, W. N. Gonçalves, and J. Li, "A review on deep learning in UAV remote sensing," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 102, Jul. 2021, Art. no. 102456. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030324342100163X

[5] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang, "UAVid: A semantic segmentation dataset for UAV imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 165, pp. 108–119, Jul. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271620301295

[6] J. Steier, M. Goebel, and D. Iwaszczuk, "Is your training data really ground truth? A quality assessment of manual annotation for individual tree crown delineation," *Remote Sens.*, vol. 16, no. 15, p. 2786, Jul. 2024.

[7] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[8] A. S. Tarawneh, A. B. Hassanat, G. A. Altarawneh, and A. Almuhaimeed, "Stop oversampling for class imbalance learning: A review," *IEEE Access*, vol. 10, pp. 47643–47660, 2022.

[9] T. Adão, L. Magalhães, and E. Peres, *Ontology-Based Procedural Modelling of Traversable Buildings Composed by Arbitrary Shapes* (Springer Briefs in Computer Science). Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-42372-2.

[10] T. Adão, L. Pádua, P. Marques, J. J. Sousa, E. Peres, and L. Magalhães, "Procedural modeling of buildings composed of arbitrarily-shaped floor-plans: Background, progress, contributions and challenges of a methodology oriented to cultural heritage," *Computers*, vol. 8, no. 2, p. 38, May 2019. [Online]. Available: https://www.mdpi.com/2073-431X/8/2/38

[11] Y. I. H. Parish and P. Müller, "Procedural modeling of cities," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.* New York, NY, USA, Aug. 2001, pp. 301–308, doi: 10.1145/383259.383292.

[12] M. Saldana, "An integrated approach to the procedural modeling of ancient cities and buildings," *Digit. Scholarship Humanities*, vol. 30, no. suppl 1, pp. i148–i163, Dec. 2015, doi: 10.1093/llc/fqv013.

[13] R. M. Smelik, T. Tutenel, K. J. de Kraker, and R. Bidarra, "A declarative approach to procedural modeling of virtual worlds," *Comput. Graph.*, vol. 35, no. 2, pp. 352–363, Apr. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849310001809

[14] M. Nitsche, C. Ashmore, W. Hankinson, R. Fitzpatrick, J. Kelly, and K. Margenau, "Designing procedural game spaces: A case study," *Proc. FuturePlay*, vol. 2006, pp. 1–7, Oct. 2006.

[15] L. Lee and N. Pavlov, "Procedural fracturing and debris generation for *Kung-Fu Panda*," in *Proc. ACM SIGGRAPH Talks*, Aug. 2008, p. 1.

[16] J. Berssenbrügge, J. Stöcklein, A. Koza, and I. Gräßler, "Procedural generation of vegetation for a virtual test track," in *Proc. 34th Comput. Inf. Eng. Conf.*, Aug. 2014, p. 1, doi: 10.1115/detc2014-34891.

[17] J. Guo, S. Xu, D.-M. Yan, Z. Cheng, M. Jaeger, and X. Zhang, "Realistic procedural plant modeling from multiple view images," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 2, pp. 1372–1384, Feb. 2020.

[18] A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D. S. Ebert, J. P. Lewis, K. Perlin, and M. Zwicker, "A survey of procedural noise functions," *Comput. Graph. Forum*, vol. 29, no. 8, pp. 2579–2600, Dec. 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2010.01827.x

[19] L. van Aanholt and R. Bidarra, "Declarative procedural generation of architecture with semantic architectural profiles," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2020, pp. 351–358.

[20] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool, "Procedural modeling of buildings," in *Proc. ACM SIGGRAPH Papers*. New York, NY, USA, 2006, pp. 614–623, doi: 10.1145/1179352.1141931.

[21] Y. Liu, C. Xu, Q. Zhang, and Y. Pan, "Ontology based semantic modeling for Chinese ancient architectures," in *Proc. 18th Conf. Innov. Appl. Artif. Intell.*, Jul. 2006, pp. 1808–1813.

[22] M. Bassier, L. Mattheuwsen, and M. Vergauwen, "BIM reconstruction: Automated procedural modeling from point cloud data," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 2, pp. 53–60, Nov. 2019. [Online]. Available: https://isprs-archives.copernicus.org/articles/XLII-2-W17/53/2019/

[23] M. Yalcinkaya and D. Arditi, "Building information modeling (BIM) and the construction management body of knowledge," in *Product Lifecycle Management for Society*, A. Bernard, L. Rivest, and D. Dutta, Eds., Berlin, Germany: Springer, 2013, pp. 619–629.

[24] G. Gröger and L. Plümer, "CityGML—Interoperable semantic 3D city models," *ISPRS J. Photogramm. Remote Sens.*, vol. 71, pp. 12–33, Jul. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271612000779

[25] Š. Kohek and D. Strnad, "Interactive large-scale procedural forest construction and visualization based on particle flow simulation," *Comput. Graph. Forum*, vol. 37, no. 1, pp. 389–402, Feb. 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13304

[26] C. Gasch, J. M. Sotoca, M. Chover, I. Remolar, and C. Rebollo, "Procedural modeling of plant ecosystems maximizing vegetation cover," *Multimedia Tools Appl.*, vol. 81, no. 12, pp. 16195–16217, Mar. 2022.

[27] X. Yang and D. Delparte, "A procedural modeling approach for ecosystem services and geodesign visualization in old town Pocatello, Idaho," *Land*, vol. 11, no. 8, p. 1228, Aug. 2022. [Online]. Available: https://www.mdpi.com/2073-445X/11/8/1228

[28] T. Niese, S. Pirk, M. Albrecht, B. Beneš, and O. Deußen, "Procedural urban forestry," *ACM Trans. Graph.*, vol. 41, no. 2, pp. 1–18, Mar. 2022.

[29] W. Li, "Procedural modeling of the great barrier reef," in *Advances in Visual Computing*, G. Bebis, V. Athitsos, T. Yan, M. Lau, F. Li, C. Shi, X. Yuan, C. Mousas, and G. Bruder, Eds., Cham, Switzerland: Springer, 2021, pp. 381–391.

[30] J. Guo, H. Jiang, B. Benes, O. Deussen, X. Zhang, D. Lischinski, and H. Huang, "Inverse procedural modeling of branching structures by inferring L-systems," *ACM Trans. Graph.*, vol. 39, no. 5, pp. 1–13, Jun. 2020, doi: 10.1145/3394105.

[31] T. Adão, T. M. Pinho, L. Pádua, N. P. Santos, A. Sousa, J. J. Sousa, and E. Peres, "Using virtual scenarios to produce machine learnable environments for wildfire detection and segmentation," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 9–15, Aug. 2019. [Online]. Available: https://isprs-archives.copernicus.org/articles/XLII-3-W8/9/2019/

[32] S. Khan, B. Phan, R. Salay, and K. Czarnecki, "ProcSy: Procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jan. 2019, pp. 88–96.

[33] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 833–851.

[34] J. Devaranjan, A. Kar, and S. Fidler, "Meta-sim2: Unsupervised learning of scene structure for synthetic data generation," 2020, *arXiv:2008.09092*.

[35] S. Fedorova, A. Tono, M. S. Nigam, J. Zhang, A. Ahmadnia, C. Bolognesi, and D. Michels, "Synthetic data generation pipeline for geometric deep learning in architecture," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 2021, pp. 337–344, Jun. 2021, doi: 10.5194/isprs-archives-xliii-b2-2021-337-2021.

[36] N. Bhandari, "Procedural synthetic data for self-driving cars using 3D graphics," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2018.

[37] R. Nunes, J. F. Ferreira, and P. Peixoto, "Procedural generation of synthetic forest environments to train machine learning algorithms," in *Proc. ICRA Workshop Innov. Forestry Robot., Res. Ind. Adoption*, 2022, pp. 1–8. [Online]. Available: https://openreview.net/forum?id=rpzgjNCe4G9

[38] R. Varghese and M. Sambath, "YOLOv8: A novel object detection algorithm with enhanced performance and robustness," in *Proc. Int. Conf. Adv. Data Eng. Intell. Comput. Syst. (ADICS)*, Apr. 2024, pp. 1–6.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015, *arXiv:1505.04597*.

[40] W. Abdulla. (2017). *Mask R-CNN for Object Detection and Instance Segmentation on Keras and Tensorflow*. [Online]. Available: https://github.com/matterport/Mask_RCNN

[41] U. Meier. (1997). *Growth Stages of Mono-and Dicotyledonous Plants*. [Online]. Available: https://library.wur.nl/WebQuery/titel/962304

[42] G. Voronoi, "Nouvelles applications design paramètres continus à la théorie design formes quadratiques. Premier mémoire. Sur quelques propriétés design formes quadratiques positives parfaites," *J. für Die Reine Und Angew. Math.*, vol. 1908, no. 133, pp. 97–102, Jan. 1908, doi: 10.1515/crll.1908.133.97.

[43] Unity. *Asset Store*. Accessed: Jul. 21, 2024. [Online]. Available: https://assetstore.unity.com/

[44] GitHub. *Build and Ship Software on a Single, Collaborative Platform*. Accessed: Jul. 21, 2024. [Online]. Available: https://github.com/

[45] Unity. *Documentation | Terrain*. Accessed: Jul. 21, 2024. [Online]. Available: https://docs.unity3d.com/Manual/script-Terrain.html

[46] GitHub. *Komietty/Unity-Voronoi*. Accessed: Jul. 21, 2024. [Online]. Available: https://github.com/komietty/unity-voronoi

[47] Unity. *Documentation | Gui*. Accessed: Jul. 21, 2024. [Online]. Available: https://docs.unity3d.com/ScriptReference/GUI.html

[48] C. Marketplace. *UE4—Forest Foliage Low-Poly 3D Model*. Accessed: Jul. 21, 2024. [Online]. Available: https://www.cgtrader.com/3d-models/plant/grass/ue4-forest-foliage

[49] B. Games. *Unity Asset Store | Bhutan Farmhouse*. Accessed: Jul. 21, 2024. [Online]. Available: https://assetstore.unity.com/packages/3d/environments/urban/bhutanfarmhouse-19360

[50] ithappy. *Unity Asset Store | Animals Free*. Accessed: Jul. 21, 2024. [Online]. Available: https://assetstore.unity.com/packages/3d/characters/animals/animals-free-260727

[51] Rokay3D. *Unity Asset Store | Farm Machinery: Low Poly Tractor and Planter*. Accessed: Jul. 21, 2024. [Online]. Available: https://assetstore.unity.com/packages/3d/vehicles/land/farm-machinery-low-poly-tractor-and-planter-94533

[52] P. López-García, J. F. Ortega, E. P. Pérez-Álvarez, M. A. Moreno, J. M. Ramírez, D. S. Intrigliolo, and R. Ballesteros, ''Yield estimations in a vineyard based on high-resolution spatial imagery acquired by a UAV,'' *Biosyst. Eng.*, vol. 224, pp. 227–245, Dec. 2022.

[53] Roboflow. *Everything You Need to Build and Deploy Computer Vision Applications*. Accessed: Jul. 21, 2024. [Online]. Available: https://roboflow.com/

TELMO ADÃO received the B.Sc. and M.Sc. degrees in informatics from University of Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal, and the Ph.D. degree from UTAD, in 2015, specializing in computer graphics with a focus on procedural modeling. He is currently an Assistant Researcher with the UTAD, where he works on the Vine and Wine project under the Portuguese Recovery and Resilience Plan (PRR). With more than ten years of research and development experience, he has developed expertise in several other areas, including virtual/augmented/mixed reality, human–computer interaction, navigation systems (outdoors), image processing, computer vision, deep learning, remote sensing, and precision agriculture. He has authored more than 90 research publications in specialized conferences and journals, most of which are indexed in Scopus and/or Web of Science. He supervised several B.Sc. and M.Sc. projects, with nine successful conclusions. He has been a Guest Lecturer with the University of Minho for the past eight years, teaching subjects mostly related to computer programming. Formerly, he served as a Research and Development Project Coordinator with the Center for Computer Graphics (CCG/zgdv), associated with the Computer Vision Interaction and Graphics (CVIG) Department, wherein he successfully supervised more than ten research and development projects and earned funding from four recovery and resilience plan (RRP–Portugal) agendas/pacts and an INTERREG/POCTEP application. He also taught at the Polytechnic Institute of Braganca, Portugal, and UTAD, covering topics, such as computer programming, computer graphics, 3D modeling, game development, and web applications.

JOÃO CERQUEIRA received the B.Sc. degree in communication and multimedia and the M.Sc. degree in multimedia from the University of Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal, in 2019 and 2024, respectively, where he is currently pursuing the M.Sc. degree in computer engineering. His primary research interests include computer graphics, particularly 3D modeling, artificial intelligence, and graphics design.

MIGUEL ADÃO received the B.Sc. degree in multimedia from the Polytechnic Institute of Braganca (IPB), Portugal, which resulted from an internship carried out in the company Gema Digital, Porto, Portugal, wherein he engaged in various 3D modeling and animation projects. During his academic journey, besides accumulating experience in game engines, he also contacted with augmented reality, more specifically, applied to special need individuals. With a more professional orientation, he maintains a portfolio of relevant work on platforms, such as ArtStation and Sketchfab.

NUNO SILVA received the M.Sc. and Ph.D. degrees in electrotechnical and computer engineering from the University of Trás-os-Montes e Alto Douro (UTAD), in 2010 and 2023, respectively. Currently, he is a Senior Technician with UTAD, associated with a project related to smart agriculture/viticulture, being responsible to design, develop, deploy, and maintain networks of IoT devices for various purposes. He also integrates the Centre for the Research and Technology of Agro-Environmental and Biological Sciences (CITAB) as a Ph.D. Collaborator. Formerly, he worked on projects in other areas of interest, with a greater focus on the development of electronic devices used in smart prostheses, being this the main subject of his Ph.D.

DAVID PASCOAL received the B.Sc. degree in agricultural engineering from the Polytechnic Institute of Viseu, and the master's degree in agricultural engineering. He is currently pursuing the Ph.D. degree in emerging technologies applied to agroforestry systems (TechAgro) with UTAD, focusing on the application of the Internet of Things (IoT) and precision viticulture to develop innovative solutions for the viticulture sector. Later, at the University of Trás-os-Montes and Alto Douro (UTAD), he successfully completed a master's program in Enology and Viticulture. Throughout his career, he has gained significant practical experience in viticulture and enology, working with six companies in the sector through internships and full-time roles. He is also affiliated with the Centre for the Research and Technology of Agro-Environmental and Biological Sciences (CITAB) as a non-Ph.D. collaborator.

LUÍS G. MAGALHÃES received the B.Sc. and M.Sc. degree in informatics from the University of Minho and the Ph.D. degree in computer science from the University of Trás-os-Montes e Alto Douro. Currently, he is an Associate Professor of Habilitation with the University of Minho. He is also a Senior Researcher with the ALGORITMI Centre. Since 2016, he has been the Scientific Coordinator of the Computer Vision, Interaction, and Graphics Research and Development Department, Computer Graphics Centre. He is the author or co-author of more than 130 scientific publications in international journals and conference proceedings. His research interests include the use of computer vision, augmented reality, and computer graphics. He is the Scientific Coordinator and a Research Member in projects related to automatic optical inspection, expeditious modeling of virtual environments, immersive 3D environments, and mixed reality systems. Since 2020, he has been a member of the Board of Director of the Computer Graphics Centre.

TIAGO BARROS received the M.Sc. degree in electrical and computer engineering from the University of Coimbra, Portugal, where he is currently pursuing the Ph.D. degree with the Institute of Systems and Robotics. His research interests include deep learning, robotic perception, and remote sensing, areas to which he already contributed with publications in relevant venues, such as the International Conference on Intelligent Robots and Systems (IROS), the IEEE Intelligent Transportation Systems Conference (ITSC), IEEE Robotics and Automation Letters (RA-L), and *Computers and Electronics in Agriculture* (Elsevier). His work primarily focuses on the intersection of robotics and remote sensing applications.

CRISTIANO PREMEBIDA (Senior Member, IEEE) is currently an Assistant Professor with the Department of Electrical and Computer Engineering (DEEC), University of Coimbra (UC), Portugal, where he is part of the Institute of Systems and Robotics (ISR-UC). He works on multimodal and multisensory perception for robotics and autonomous systems applications, developing calibration strategies and probability-prediction approaches to increase the robustness of deep models. Recently, he has been coordinating the EU Horizon MSCA-DN project AIGreenBots fusion. His main research interests include robotic perception, machine learning, autonomous vehicles, autonomous robots, agricultural robotics, and sensor. He is a member of the IEEE ITS and RAS Societies, where he has been serving as an Associate Editor for the IEEE conferences IROS, ITSC, and IV.

URBANO J. NUNES (Life Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Coimbra (UC), Portugal, in 1995. He is currently a Full Professor with the Department of Electrical and Computer Engineering, UC, and a Senior Researcher with the Institute of Systems and Robotics, where he is the Coordinator of the Human-Centered Mobile Robotics Laboratory. He has been involved with/responsible for several national/international funded projects in the areas of human-centered mobile robotics and intelligent vehicles. He has been active in the organization of conferences, such as the General Chair of the 2010 IEEE Intelligent Transportation Systems Conference, the General Chair of the 2012 IEEE/RSJ Intelligent Robots and Systems, and the General Chair of the 2017 IEEE International Symposium on Robot and Human Interactive Communication. He is a Senior Editor of IEEE Transactions on Intelligent Vehicles.

EMANUEL PERES received the degree in electrical engineering and the Ph.D. degree in engineering sciences from University of Trás-os-Montes e Alto Douro (UTAD), Portugal, in 2003 and 2011, respectively. He is currently an Associate Professor with the Department of Engineering, School of Science and Technology, UTAD. He is also involved in digital reality applications and cybersecurity. He has an H-index of 22 on Scopus and 20 on Web of Science, with 2589 citations in Scopus and 2029 citations in Web of Science. He is a Co-Investigator with the DATI–Digital Agriculture Technologies for Irrigation Efficiency, funded by the PRIMA initiative, focusing on optimizing irrigation management through innovative digital agriculture technologies. He is also a Key Member of the Vine and Wine PT project, part of the Recovery and Resilience Plan (PRR), which aims to foster digital transformation in Portugal's wine sector, with a total funding of C7.8 million for UTAD. Additionally, he contributes to other significant research initiatives that leverage cutting-edge the IoT and AI solutions for sustainable agricultural practices. He has published 106 articles indexed in Scopus and 81 in Web of Science. He is actively involved in several high-impact research projects. His research interests include decision support systems for digital agriculture, remote monitoring using remotely piloted aerial vehicles (RPAV), the Internet of Things (IoT), big data, analytics, and artificial intelligence. He serves on the Editorial Boards of some journals, such as *Remote Sensing*, *Frontiers in Plant Science*, and *Drones*.

RAUL MORAIS received the bachelor's degree in electrical engineering from University of Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal, in 1993, specializing in electronics, instrumentation, and computer science, the M.Sc. degree in industrial electronics from the University of Minho, in 1998, the Ph.D. degree in electrical and computer engineering, in 2004, with a focus on microelectronics, and the Habilitation degree in electrical and computer engineering from UTAD, in 2009. He is currently a Full Professor with the Engineering Department, School of Sciences and Technology, UTAD. His areas of expertise include sensors and sensor interfaces in CMOS microelectronics, energy harvesting techniques for small, unattended electronic devices, and wireless sensor networks. He also focuses on the Internet of Things (IoT) within the context of precision agriculture/viticulture. Additionally, he has a keen interest in biomedical implantable devices, particularly biotelemetry systems for vibration micro-generators that produce electric energy within smart prosthetics. Currently, he is a Senior Researcher with the Centre for the Research and Technology of Agro-Environmental and Biological Sciences (CITAB) and a Collaborator with the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC). He is involved in 11 research projects, including four international ones. He serves as UTAD's coordinator for the PRR Agenda Mobilizadora Vine and Wine and leads the POCTEP/INTERREG VA project ''SIAPD'', which supports pest and disease control in agriculture.