computers

MDPI

Article

Procedural Modeling of Buildings Composed of Arbitrarily-Shaped Floor-Plans: Background, Progress, Contributions and Challenges of a Methodology Oriented to Cultural Heritage

Telmo Adão [1,2,*], Luís Pádua [2], Pedro Marques [2], Joaquim João Sousa [1,2], Emanuel Peres [1,2] and Luís Magalhães [3]

[1] Institute for Systems and Computer Engineering, Technology and Science (INESC-TEC—Formerly INESC Porto), 4200-465 Porto, Portugal; jjsousa@utad.pt (J.J.S.), eperes@utad.pt (E.P.)

[2] Department of Engineering, School of Sciences and Technology, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal; luispadua@utad.pt (L.P.), pedro.marques@utad.pt (P.M.)

[3] ALGORITMI Center, University of Minho, 4800-058 Guimarães, Portugal; lmagalhaes@dsi.uminho.pt

* Correspondence: telmoadao@utad.pt; Tel.: +351-259-35-07-62

Received: 14 March 2019; Accepted: 29 April 2019; Published: 11 May 2019

Abstract: Virtual models' production is of high pertinence in research and business fields such as architecture, archeology, or video games, whose requirements might range between expeditious virtual building generation for extensively populating computer-based synthesized environments and hypothesis testing through digital reconstructions. There are some known approaches to achieve the production/reconstruction of virtual models, namely digital settlements and buildings. Manual modeling requires highly-skilled manpower and a considerable amount of time to achieve the desired digital contents, in a process composed by many stages that are typically repeated over time. Both image-based and range scanning approaches are more suitable for digital preservation of well-conserved structures. However, they usually require trained human resources to prepare field operations and manipulate expensive equipment (e.g., 3D scanners) and advanced software tools (e.g., photogrammetric applications). To tackle the issues presented by previous approaches, a class of cost-effective, efficient, and scarce-data-tolerant techniques/methods, known as procedural modeling, has been developed aiming at the semi- or fully-automatic production of virtual environments composed of hollow buildings exclusively represented by outer façades or traversable buildings with interiors, either for expeditious generation or reconstruction. Despite the many achievements of the existing procedural modeling approaches, the production of virtual buildings with both interiors and exteriors composed by non-rectangular shapes (convex or concave n-gons) at the floor-plan level is still seldomly addressed. Therefore, a methodology (and respective system) capable of semi-automatically producing ontology-based traversable buildings composed of arbitrarily-shaped floor-plans has been proposed and continuously developed, and is under analysis in this paper, along with its contributions towards the accomplishment of other virtual reality (VR) and augmented reality (AR) projects/works oriented to digital applications for cultural heritage. Recent roof production-related enhancements resorting to the well-established straight skeleton approach are also addressed, as well as forthcoming challenges. The aim is to consolidate this procedural modeling methodology as a valuable computer graphics work and discuss its future directions.

Keywords: procedural modeling; procedural generation; building ontology; building modeling; virtual traversable building; ontology-based grammar; arbitrary shapes; fake-concave; virtual reality; augmented reality; cultural heritage; Roman architecture; Roman style; computer-aided design

1. Introduction

Virtual reconstruction refers to the faithful digital reproduction of real-world entities such as objects, settlements, and even biological systems. Its usefulness has been shown in a wide range of areas, including biomechanics [1], architecture [2], post-catastrophe event analysis [3], reconstructive surgery [4,5], Mixed Reality (MR) [6], education [7], and archeology [8–10], among others.

Approaches to achieve virtual reconstruction effectively can be classified into non-automatic and semi-/fully-automatic techniques, both depending on the information source. While the former essentially refers to manual modeling, the latter includes image-based, range-scanning, and procedural modeling techniques. Manual modeling relies on the usage of Computer-Aided Design (CAD) software tools and usually requires skilled labor, as well as a considerable amount of production time. Archeological virtual reconstruction demands a multidisciplinary articulation between several specialists such as modelers, designers, programmers, and archeologists or historians. Another couple of strategies capable of streamlining the modeling process using techniques to extract information from a particular source stand as alternative. On the one hand, image-based approaches—resorting to, for example, the pinhole model or structure from motion (SfM) techniques recurrently applied in photogrammetric processes—enable the estimation of 3D positions out of 2D digital imagery composed of single or multiple views, which can be obtained either from archives or surveying processes. While data scarcity associated with the single-view approach leads to the need for additional inputs—eventually provided by the user—the multiple views' approach requires proper image sets commonly obtained through surveying activities that often demand: (1) extensive analysis and preparation of the area of interest, when carried out terrestrially; and (2) specialized knowledge on manned or unmanned aerial platforms, when carried out by air. Besides, technical know-how to deal with proper data processing tools—such as photogrammetric software—cannot be neglected. On the other hand, range scanning resorts to usually expensive sensing devices such as Light Detection And Ranging (LIDAR), Synthetic Aperture Radar (SAR), and laser technology to obtain point cloud representations of the scanned surfaces. It involves the use of expensive equipment and technical knowledge to operate them, as well as to process data in specialized tools. Besides, the time consumption associated with both image-based and range scanning approaches, the effectiveness of works requiring virtual reconstruction of settlements—such as buildings and monuments—is directly related to those structures' conservation state. Still, within the semi-/fully-automatic approaches topic, procedural modeling has been studied and applied to generate virtual environments and structures based on rule sets, with a minimal amount of user interaction and time consumption. Particularly for archeology, it brings advantages in the reconstitution of missing or severely-damaged structures, since it enables rapidly obtaining glimpses of their hypothetical representations. Thus, procedural modeling turns out to be suitable not only as an exclusive or complementary virtual reconstruction strategy—according to available information—but also to iteratively test and refine hypotheses towards pertinent virtual models depicting convincing ancient structures [11]. Table 1 sums up the main characteristics associated with each approach, in a virtual reconstruction perspective.

Table 1. Comparison between different virtual reconstruction approaches—manual, image-based, range scanning, and procedural—by features, adequacy, requirements, and automation level.

Modeling Topic	Manual	Image-Based	Range Scanning	Procedural
Short definition	Traditional handmade modeling through the use of CAD software tools.	3D estimation supported by the pinhole model, triangulations, and SfM used in photogrammetry pipelines, among others.	Active sensor-based modeling acquiring surfaces' 3D morphology, (usually) accessed as point clouds.	Collection of semi-automatic processes and techniques with a reduced demand for user interaction.
Conservation adequacy	Both well-conserved (digital reproduction) and degraded structures (hypothesis proposal).	Well-conserved structures (e.g., aiming at digital heritage documentation and conservation) or proper photographic documentation prior to advanced deterioration.	Well-conserved structures (e.g., aiming at digital heritage documentation and conservation).	Both well-conserved (digital reproduction) and degraded structures (hypotheses proposal).
Possible requirements	Skilled labor, time, and/or numerous human resources; need to understand issues related to geometric construction and morphology modeling.	Additional inputs (single view); proper image sets (multiple views); eventual training on time-consuming surveying techniques; knowledge on specialized software (e.g., photogrammetry); computational power.	Expensive sensors or data access; eventual training on time-consuming surveying techniques; knowledge on specialized software (LIDAR/GIS); computational power; complementary RGB sensor for color acquisition.	Grammars and rules systems that require some isolation level (e.g., graphical tools); computational resources may impact expeditiousness; rules to address specific geometric construction and morphology modeling issues must be supported.
Potential for automation	Lowest.	Medium.	Medium.	Highest.

During the last few decades, the procedural modeling scientific community has proposed different methodologies that aim at the semi- or fully-automatic production of urban environments represented by virtual buildings' exteriors, as well as virtual buildings with interiors, either for expeditious generation or reconstruction. The virtual models outputted by those methodologies are of major importance in research and business fields like architecture, archeology, or video games, to satisfy different requirements, such as expeditious building generation for extensive virtual environment production and hypothesis testing based on virtual reconstructions.

Despite the many achievements of the existing procedural modeling approaches, most of them still lack support of the generation of virtual buildings with both exteriors and interiors composed of arbitrary shapes—non-rectangular shapes (convex or concave n-gons)—at the floor-plan level. Considering the limitations, new processes and techniques, capable of properly supporting some relevant tasks (e.g., digital preservation of cultural heritage or extensive virtual urban environments production), specifically those involving the generation/reconstruction of virtual buildings with irregular geometric requirements, were developed in a methodology that considers ontology-based grammar rules as input to derive coarse 3D building models externally constrained by arbitrary shapes and having internal divisions delimited by convex shapes [12]. Regarding the generation process, an adapted treemap approach is used to subdivide the building layout into floor-plan areas, ranging from rectangles to arbitrary shapes. Moreover, a method concerning internal room walls' adaptation is addressed. Then, a set of operations is performed, from the marking transitions' step to the walls' extrusion process, which provides the 3D aspect. Rather than validating architectural rules, this work aimed to provide a particular class of specialists—namely archeologists and multimedia professionals—the freedom to timely propose and iteratively refine virtual models, until the achievement of plausible hypotheses.

The proposed procedural modeling methodology [12] work started to be delineated in [13,14] and passed through a few enhancements [15,16] until more recent versions [12,17]. This methodology had its first implementation in the "Expeditious Reconstruction of Virtual Cultural Heritage Sites" (ERAS) research project (PTDC/EIA-EIA/114868/2009), which focused on the virtual reconstruction of Roman buildings [18]. Later, the "Adaptive Mixed Reality System for Archeology Sites" (MixAR) project (2014/038803, European fund for regional development)—co-developed with GEMA Digital and Technology Agency enterprise (Oporto, Portugal)—benefited from virtual models produced by the enhanced version of this procedural modeling methodology [16] to integrate them in Mixed Reality (MR) environments providing virtual visitations to the past [19–24]. Recent contributions were also given to a system combining Unmanned Aerial Systems (UAS)-based land surveying, photogrammetry, and procedural modeling of buildings to set up enhanced VR environments for cultural heritage [25]. Indeed, the methodology that is being analyzed in this paper played a primary role for the success of the presented projects/works. Meanwhile, some progress was made to improve buildings' roofs generation, using the straight skeleton approach. Future challenges and final remarks are also presented and discussed.

Overall, this paper's main goal is the consolidation of a previously-developed procedural modeling methodology [12] that is addressed in two principal moments: (1) solution review; and (2) documentation of development stages, along with significant contributions to other research projects/solutions. The underlying contributions include:

- chronological compilation of the proposed methodology progress and related works in a single documenting source;
- overview of the methodology contributions that allowed accomplishing other VR/AR works supporting archeology promotion, providing an insight towards procedural modeling's practical application;
- incorporation of grammar rules and geometric operations to support the production of straight skeleton-based roofs [26–28] over either a single building part or an elevated polygon clustering the layouts of a set of neighbor, although individual, building parts;

- presentation of an integrated digital environment built up with the support of the proposed methodology, which includes both manually-produced and procedurally-generated virtual models.

The following section addresses some important works that have been proposed using both non-procedural and procedural modeling approaches, including contributions to archeology, with a stronger focus on the latter, which is the main topic underlying this work.

2. Background on Settlements' Virtual Production/Reconstruction

This section is dedicated to briefly review non-automatic, image-based, and range scanning methods, as well as procedural modeling approaches, which have been leading to progress in the digital cultural heritage area, more specifically in the recovery and/or preservation of historically-relevant structures/settlements.

2.1. Non-Procedural Modeling Approaches

In spite of being quite demanding regarding both time and human resources, non-automatic (manual) modeling enables a step-by-step follow-up, usually from scratch, which provides full control and customization over the entire virtual models' production pipeline. While it supports many professional/research areas (e.g., architecture, engineering, and construction [29]), this modeling style has been providing contributions to cultural heritage recovery, observable in works such as the recreation of Bracara Augusta (Braga, Portugal) [30], the recovery of cultural heritage buildings located in Conímbriga (Portugal) [31], and the reconstitution of Lisbon (Portugal) prior to 1755's devastating earthquake [32]. Matini and Ono [33] documented the quality verification performed over virtual models that were produced to recover and preserve digitally cultural heritage buildings ruined after an earthquake that impacted the Citadel of Bam (Iran), in 2003. Their aim was to ensure representational accuracy and faithfulness, not only for visualization, but also for eventual research purposes. More results of these reconstruction activities can be found in [34]. Furthermore, among the works worthy to highlight is a project wherein a virtual model of the historical Vila Velha (located in Vila Real (Portugal)) was produced, considering the 13th Century as the temporal reference. It was developed based on data gathered during the excavations from a topographic survey, on-site evidence, discussions with an archeology specialist, analysis of similar structures, and documentation [35]. To recover monuments of the Volga Region (Republic of Tatarstan, Russia) dated from the 14th Century, a project uniting experts of several areas—such as historians, architects, 3D modelers, programmers—was carried out, focusing on the virtual reconstruction of several monuments at different conservation stages, from intact to vanished. Concerned with public participation, a promotional game was also developed [36]. In [37], a virtual reconstruction of the TT209 tomb (Luxor, Egypt) consisting of a two-step process was presented: firstly, several elements of that point of interest were collected, including photographs, videos, and measurements, and then, Blender (Blender Foundation, Amsterdam, Netherlands) was used to develop the set of models that was integrated in a game built for teaching purposes. In some recent works, manual modeling is still an option for missing structures. For example, in [38], the Segeberg castle (Kalkberg, Germany) was virtually reconstructed using AutoCAD (Autodesk, Inc., San Rafael, CA, USA).

Image-based and range scanning modeling are both considered semi-automatic approaches due to the possibility of using advanced software tools capable of producing incredibly faithful virtual models out of data acquired from a monument, a building, or even an area of interest. Both approaches are widely adopted by the scientific/professional community linked to the virtual reconstruction field, especially if the structures/areas of interest are in a good conservation state and despite the dullness associated with surveying processes that are often required for faithful outcomes, technology and data access' (e.g., from satellites) typical expensiveness, the need for skills to handle specific equipment and software, as well as the usually required know-how regarding field operations. Within the image-based scope, Grossmann and Santos-Victor [39] proposed a 3D reconstruction method from a single or multiple images and user-provided geometric constraints. Eiffel Tower from a postcard was used for validation. The reconstruction of the archeological site of Sagalassos (Turkey) was addressed in [40], through an image correspondence method. By using image repositories, other authors [41] were able to reconstruct monuments (e.g., the Colosseum of Rome) and locations (e.g., St. Mark's Square), virtually. More recently, another method aiming at the 3D reconstruction of archeological excavations was proposed and validated in Boudelo Abbey (Belgium) [42], while the multi-image SfM-based technique resorting to aerial and terrestrial data collection was evaluated against 3D range-scanning and recognized as a high-quality result generator [43]. With the purpose of evaluating the surveying capabilities of consumer-grade UAS for cultural heritage reconstruction purposes, a test was performed with a DJI Phantom 2 (DJI Science and Technology Co. Ltd., Shenzhen, Guangdong, China) that consisted of collecting aerial images of the "Delizia Estense del Verginese" castle (Portomaggiore, Italy). The photogrammetric processing over collected data produced a virtual model with an average error less than 2 cm [44]. Dhonju et al. [45] also corroborated the previous conclusion, after carrying out some tests targeting temples and monuments located in Nepal. Pierdicca [46] documented the reconstruction of Palacio Tschudi (Chan, Peru) through photogrammetry, in which both UAS-based aerial and terrestrial surveys were carried out. Consumer-grade equipment was used—for example, a DJI S800 (DJI Science and Technology Co. Ltd., Shenzhen, Guangdong, China) performed the flights as UAS with an on-board Sony Alpha NEX-7 camera (Sony Corporation, Tokyo, Japan)—and the resulting models' validation resorted to sketches and local experts' evaluation. Range scanning has also been put into practice through a wide range of devices and methods capable of performing positional mapping, with proven benefits for archeology. For example, satellite-based radio detection and ranging (radar) technology was previously used to map ruins of the Maya civilization with interesting results, even in forested areas [47]. Another widely-known sensing method is LIDAR, which has been used not only for terrain's digital modeling, enabling the detection of archeological ruins [48], but also for digital reconstruction purposes [49]. Laser devices have long been used in the field of archeology as well, with applications in site modeling [50,51] or artifact reconstruction [52]. More recently, in [9], a methodology to survey and reconstruct archeological sites was proposed and tested in a group of Cham temples located at Mỹ Sơn (Vietnam). While laser scanning was applied to survey preserved structures, photographs were taken, mainly to support texturing completions. Based on exiting documentation and also on the monuments still standing, the missing structures were reconstructed with a sense of reality. There is also a review available addressing the application of laser scanning in cave surveying [53], in which several contributions for archeology/cultural heritage were highlighted. Rodríguez-Gonzàlvez et al. [54] proposed a mobile LIDAR system that combines multiple sensors—active sensors, inertial, and global navigation system—to survey archeological sites rapidly and produce geospatial 3D point clouds accurately. The Medieval Wall of Avila (Spain) was used as a case-study. Besides the individual integration of image-based and rage scanning, the fusion of approaches has been explored as well. For example, Remondino et al. [55] found such fusion useful to document castles in Trentino Province (Northern Italy) digitally. Among the wide set of technologies that were used, photogrammetry was the preferred for acquiring the exterior parts of the buildings of interest, as well as surrounding elements, while a laser scanner was employed to capture finer details in some selected interiors. A methodology gathering UAS-based photogrammetry and terrestrial laser scanning was later proposed by Xu [56]. It was tested in a historical monument of Fujian Province, China. In [57], a combination of LIDAR, ground, and airborne imaging was applied to build virtual models of historically-relevant settlements in Quintana Roo, Mexico. Figueiredo et al. [58] followed similar operational lines to carry out the reconstruction of the Milreu archeological site (Roman Vila located in Faro, Algarve, Portugal). Moreover, the Building Information Modeling (BIM) standard was integrated in their methodology for organization, sharing, and preservation purposes. Fernández-Palacios, Morabito, and Remondino [59], who proposed a pipeline for virtual models' visualization relying in VR technologies, also used this combination of approaches to produce the digital assets for their case study.

More works adopting non-procedural modeling strategies can be extensively consulted in other accessible sources (e.g., [60]). Meanwhile, the following subsection addresses an alternative to tackle previous approaches' limitations, known as procedural modeling.

2.2. Approaches Relying on or Including Procedural Modeling

When comparing to other modeling approaches, techniques relying on procedural modeling are usually much more efficient when considering both time and resource harnessing. Moreover, they seem adequate for situations characterized by information scarcity. In the specialized literature, two main approaches are reported: (1) the production/reconstruction of extensive virtual urban environments that are typically characterized by networks of streets, building clusters, and exterior building façades; and (2) production/reconstruction of floor-plans and virtual 3D traversable buildings (including interiors).

Regarding the first approach, Parish and Müller [61] introduced a procedural methodology based on Lindenmayer system (L-system) [62] that is able to produce road networks and buildings with different formats. Subsequent studies [63,64] focused on the production of buildings with highly-detailed façades using, respectively, a shape grammar and a few user inputs specifying high-level requirements. Greuter et al. [65] addressed the generation of buildings with random extrusions, taken from 2D shapes. Dikaiakou et al. [66] developed a partly-automatic method that combines several data sources regarding buildings such as the bases' geographic data, classification, and style-concordant components (e.g., doors and balconies) to model a part of old Nicosia (Chipre). A versatile approach that focuses on the production of virtual buildings through the manipulation of volumetric shapes was proposed by Müller et al. [67] and is known as computer graphics architecture (CGA) shape grammar. It was used in other works focusing on the digital recovery of cultural heritage as, for example, the Puuc buildings of Xkipché (Mexico) [68], Ancient Rome (320 AD) [10]—in which procedural models representing missing or severely-damaged buildings were incorporated with manual replicas of the conserved ones—, and Carcassone's old town (France) [69]. Furthermore, using CGA, but combined with statistics and fuzzy logic, Tepavćević and Stojaković [70] addressed Neo-Gothic chapels' appearance simulation, inspired by a set of sacral Roman Catholic buildings located in Vojvodina (Serbia). Their case study focused on the generation of Neo-Gothic chapels. Shortly after CGA's release, Lipp et al. [71] proposed an upgrade to [63,67] with interactivity and visual editing features that aim to suppress the need for a direct contact with text-based rules. The City Modeling Procedural Engine (CMPE) uses a set of input elements such as aerial photographs, vector, and raster maps or even text descriptions to produce extensive urban environments [72]. Regarding the ontology-based methods, the challenge of recovering the cultural heritage of ancient China was addressed by Liu et al. [73] by means of a city generator, a grammar for buildings' definitions, and a style checker to avoid generation inconsistencies, such as buildings upon streets. Another automatic modeling method based on constructive grammar for building Chinese Huizhou traditional vernacular houses was recently presented by Li et al. [74]. Layout generation and feature generation are the two steps composing that method. A Monte Carlo sampler was also adapted to increase computational efficiency. In [75], a system that uses the instructions sets stored in a spatial database to model urban elements (for example, buildings, roofs, or balconies) and also geographic features was proposed. Kelly and Wonka [76] worked in an interactive architectural modeling with procedural extrusions. In [77], a partitioning method was proposed to generate city blocks controlled by user specifications, relying on guidelines capable of ensuring geometric plausibility for the subsequent modeling of buildings. Later, other authors [78] implemented a visual, interactive, and intuitive system to perform copy and paste of rule sets for procedurally-modeled buildings. More recently, performance issues have been addressed [79,80] through the combination of shape-grammars' flexibility with the performance capabilities of the graphics processing unit (GPU) to generate large-scale virtual cities composed of building hulls, much faster than all previous works. A layer-based solution for modeling building façades was proposed by Ilcík et al. [81]. An approach named "cylindrical coordinate system" was proposed in [82], to address the generation of the conic parts that often constitute historical buildings. Supported operations to virtually build structures such as towers, barrel/crossed-vaults include attachments, cuts, splits, and intersections. More advances regarding the procedural generation of buildings addressing exterior façades or outdoor components can be found in [83–85].

Typically, the procedural modeling of extensive virtual urban environments only produces hollow buildings that lack interior rooms with passages between them. Thus, the interiors' related approach complements the first one by addressing the procedural modeling of traversable buildings composed of interior divisions, besides the exterior façades. One of the earliest solutions found addressing this topic consisted of a generative grammar to produce floor-plans of Palladian style villas [86]. Following a similar line, Rau-Chaplin et al. [87] proposed a shape grammar to generate floor-plans of basic rooms and simple geometry. These rooms belong to functional zones that are considered for the coherent placement of furniture. Later, Bradley [88] developed a set of 2D and 3D Constructive Solid Geometry (CGS) algorithms to generate buildings with interiors, and Martin [89] applied a Monte Carlo algorithm in a house generator that places and expands a room set until it reaches a state of balance, considering a contention area. Real-time generation of building interiors and infinite cities was presented by Hahn et al. [90]. Dahl and Rinde [91] proposed an algorithm for generating buildings confined by irregular polygons, which uses a Voronoi approach to subdivide interior rooms. Other methods for generation and reconstruction of traversable buildings were presented [92], namely for developing a multifaceted method capable of generating regular houses, which comply with the Portuguese general regulation for urban buildings (RGEU, from Portuguese Regulamento Geral das Edificações Urbanas [93]), and reconstructing ancient Roman houses (based on Vitruvius' rules [94]). More recent progress towards the semi-automatic virtual reconstruction of traversable Roman buildings can be found here [95,96]. Inspired by the way architects design floor-plans, Lopes et al. [97] proposed a grid-based growth method for procedural floor-plan generation. The squarified treemap approach [98] inspired Marson and Musse [99], as well as Mirahmadi and Shami [100] to divide building footprints into a given set of rectangular areas, considering the aspect ratio constraints that characterize the referenced method. Merrell et al. [101] used a Bayesian network for rooms' connectivity (i.e., for topology optimization), and for geometric optimization purposes regarding the layout, they utilized a greedy Metropolis algorithm. Trescak et al. [102] combined activity-based ontologies with shape grammars to guide the generation of virtual worlds. After the Generative Modeling Language proposal as a programmatic-based rule system that allows the definition of buildings with interiors for further generation [103], Tutenel et al. [104] developed a framework that allows traversable buildings' production, using several procedural modeling techniques for different generation stages, and Leblanc et al. [105] presented a programming tool that allows the creation of buildings through the manipulation of geometric elements (2D and 3D shapes). Santos et al. [106] worked on a tool that aims at the semi-automatic production of 3D buildings with interiors, furniture, and surroundings by considering several inputs such as scanned floor-plans and room areas. The feasibility of point grid room first structure generation (FPGRFSG) is a work authored by Bjelkesten [107] that aimed at the generation and evaluation of buildings for video games integration. In the assessment stage, participants were convinced that some of the generated structures, with non-traversable subdivisions, belonged to the video game environment under testing. Relying on a graph approach to design generation (GADG), a solution to enable the production of rectangular floor-plans that can be further improved or customized was proposed [108]. Recently, an ontology-based procedural modeling methodology, capable of generating virtual one-story buildings, including exteriors outlined by arbitrary shapes and interiors formed by convex polygons, was presented [12]. In this paper, the mentioned methodology [12] is briefly revisited for contextualization purposes, and then, the integration of the virtual buildings procedurally modeled as the methodology's outcomes for other projects (also carried out by most of this paper's authors) is addressed. Afterwards, improvements that enabled the production of buildings with straight skeleton-based roofs pursuing a more natural look are presented.

3. Procedural Modeling of Buildings Composed of Arbitrarily-Shaped Floor-Plans: Revisiting the Methodology

The procedural modeling methodology proposed in a previous work [12] is regulated by an ontology [14] relying on the City Geography Markup Language (CityGML) specification [109], which aims to constitute a knowledge-based skeleton to control the procedural operations, by establishing boundaries for the geometry generated for each ontology-based element (exterior aspect, windows, doors, room, roof, etc.). Essentially, it assigns buildings' conceptual parts with the geometrical aspects, while considering both as distinct entities. In practice, the referenced ontology oriented the design of a structured grammar with parameters, which aims to provide the procedural modeling process with inputs representing buildings' specifications—including rooms, types, dimensions, and connections—which are ceded in text files properly arranged according to the referenced grammar.

Regarding the methodology process, it is composed of four steps. The first step consists of the data input that is provided using an ontology-based grammar for specifying buildings. Then, each rule set resulting from the first step is verified in terms of typos and structure to check malformed grammars or inconsistencies related to the virtual building's definition. After the proper validation, third step takes place by performing the procedural modeling process to create the virtual building model, which in turn, is outputted in the fourth step. The next few subsections explain these steps with a little more detail. Figure 1 depicts the procedural modeling methodology architecture.

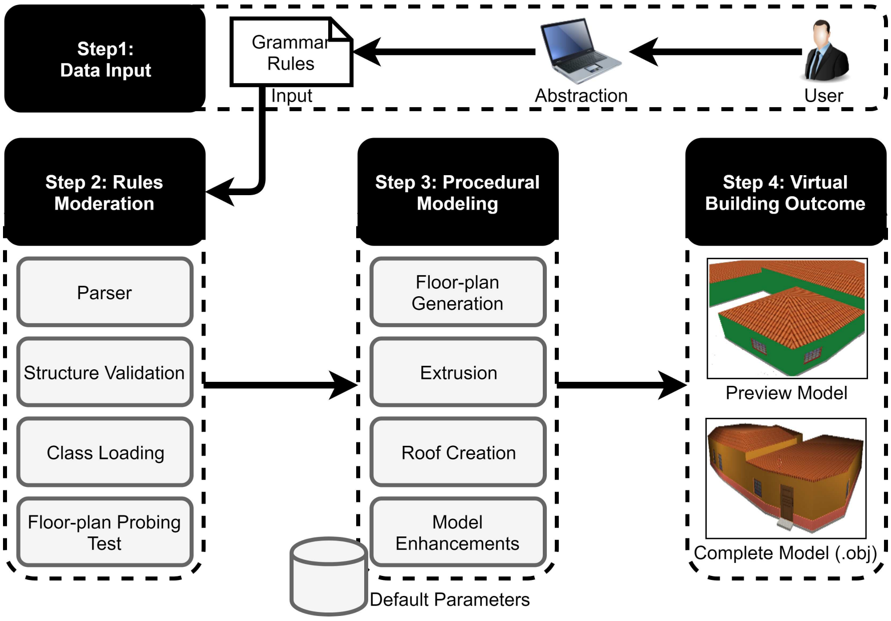


Figure 1. Architecture of the procedural modeling methodology, divided into four main steps [12]: (1) data input with the proper mechanisms for hiding the complexity that is typical in grammars—a graphical user interface (GUI) can provide such an abstraction, as will be explained later; (2) rules' moderation; (3) procedural modeling processing; and (4) virtual building model outcome. In the first step, the generation mode is selected among the two available options: deterministic (virtual buildings' generation through the user-based floor-plan parameterization) and stochastic (computer-based random generation of virtual buildings). The second step performs debugging for rules and geometric inconsistencies and configures the logic structure of the building, which is then transformed into a virtual model by the third step and finally outputted by the fourth and last step.

3.1. Step 1: Data Input

A set of definition rules contains a group of essential fields that aims to characterize a building and constitutes the main input for the procedural modeling methodology. It is presented in the form of an ontology-based grammar that was designed to define virtual buildings, considering the works of many other authors who had adopted grammars in their procedural modeling approaches [63,67,87,102,103,110,111].

The ontology-based grammar definition recovers two concepts: L-system [62] and treemap [112]. The L-system is used to iteratively produce transformations through the symbols that are context-free. However, transformations made to the symbols are "constructor functions" designed to operate considering ontology conventions. These transformations are also known as production rules and follow a logic sequence to achieve the building definition that comprises the number of vertical building parts and related floors, the divisions contained by each floor, and also the connections between these divisions. Considering those structures, a treemap of ontology-based objects is progressively set up.

The building axiom is the starting element, which in turn, is decomposed into building parts (corresponding to building's horizontal components/sections) and transitions (representative of doors and windows). Each building part is decomposed into floors (vertical stack of components) that are then subdivided into divisions (inner rooms). Divisions and transitions are final symbols, with no decomposition. More details can be found in [12,17]. The referenced production rules follow the format:

```
axiom->function(_parameters): {set_of_resulting_symbols},
```

Axiom is the symbol under processing; *function* is the ontology based operation; and *set_of_resulting_symbols* constitutes the resulting symbols that should be processed in later steps. The complete set of ontology-based grammar functions can be found in [12,17]. Deriving rules are set up either automatically through computer-based stochastic generation, addressed with more detail in Section 4.3, or specified/guided by user inputs, using, for instance, the interactive tool presented in Section 3.6.

3.2. Floor-Plan Subdivision Common Approach

In this subsection, the floor-plan division approach is introduced, since it is common to further steps (addressed in Sections 3.3 and 3.4) integrating the proposed methodology.

As referenced in previous Section 3.1, each definition rules set includes a treemap encoding both the organization and occupation of the internal elements of a building—i.e., containers for arrangement purposes, building parts, or divisions—which guides the building's layout splitting process. For the sake of simplicity, this subsection will generically refer to the kind of structures encompassing the referenced components as the building layout sectioning element (BLSE). Essentially, the process follows a treemap subdivision (depicted in Figure 2) with some intersection and dimensioning operations to deal with the division of the referenced convex polygon. The concept relies on a simplified axis-aligned version of the oriented bounding boxes' subdivision proposed in [77]. Comparatively to CGA, splitting operations are carried out based on a treemap structure configured through an ontology-based grammar specification that arranges BLSEs, defining relative values for area occupation (weights) inside a container, instead of directly manipulating geometries.

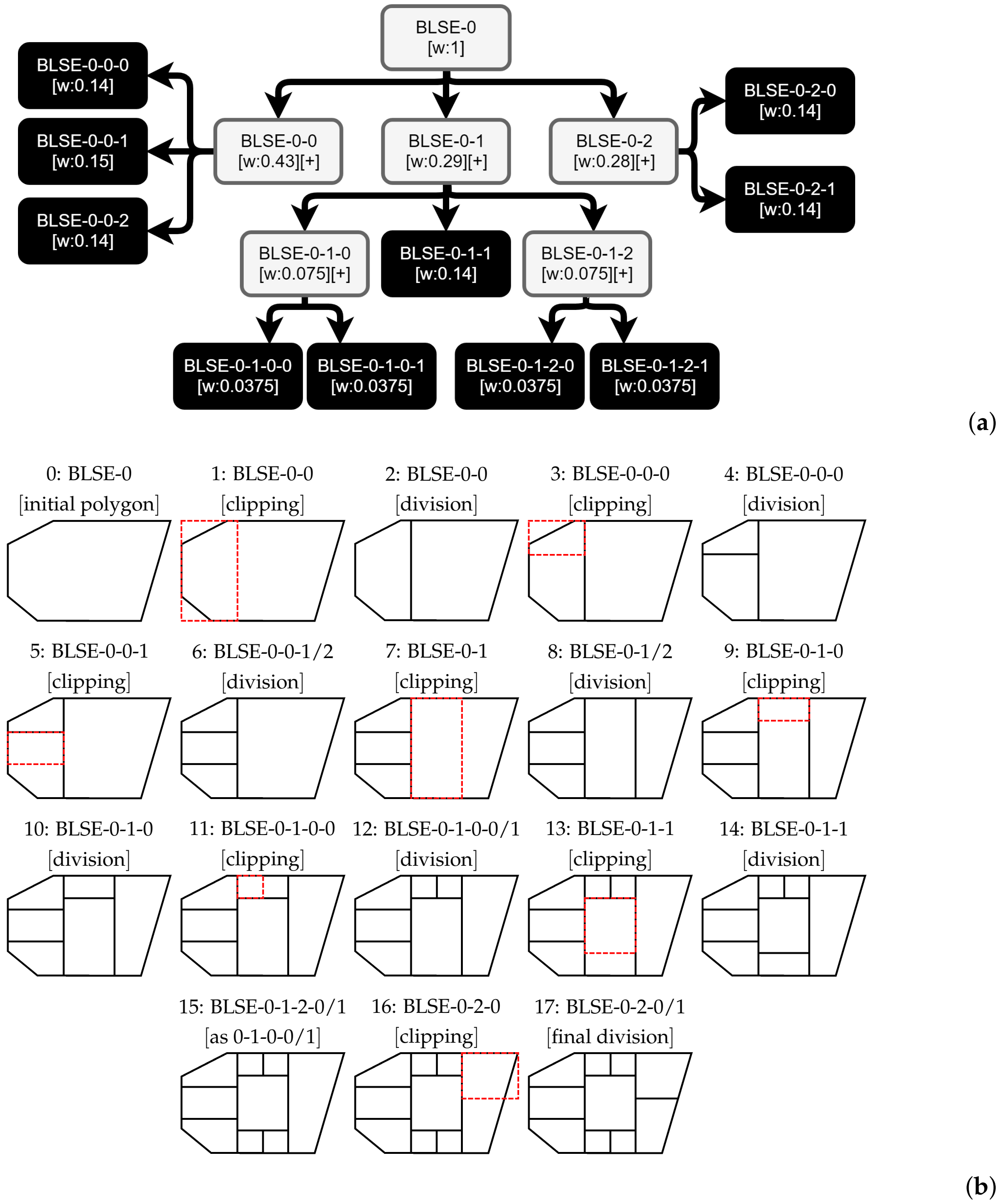


Figure 2. Treemap recursive subdivision process to set up building layout sectioning elements (BLSEs) inside a given layout. In (**a**), a generic tree of BLSEs is depicted, specifying a set of weights of occupation and structural relationships/compositions for a floor-plan initial definition. The numbers assigned next to BLSE identify not only the sectioning element, but also its depth (e.g., BLSE-0-1-0 refers to the first inner BLSE of the Sub-BLSE 1 that, in turn, is arranged inside main BLSE 0, coincident with the building's layout). In (**b**) a step-by-step processing of the previously referenced tree is presented, wherein a splitting task of a given layout, representing the building's polygon, takes place, assuming a recursive behavior for in-depth operations, to divide BLSEs inside BLSEs. This is done by strategically using intersecting rectangles of variable initialization and mutating sizes along the building's layout. Each one of these rectangles can increase or decrease in area, until the resulting intersection shape reaches an approximate value regarding the excepted area, complying with the respective weight of occupation (specified in an earlier step for the *BuildingPart* and *Division* structures).

In short, the footprint division is a recursive process that iterates along the building's contour and at many depths as required for arrangement purposes, ensuring weights' back-propagation updating from deeper to superficial BLSE layers, i.e., when final depth BLSE (*BuildingPart* or *Division* structures) adjusts the weights, a chain alteration occurs towards the first layer of BLSEs, to conserve sectioning process coherency and, of course, a cumulative weight of one. For each sub-BLSE within a container BLSE (representing building layout or any other deeper division process to arrange building parts and divisions inside of each other), an auxiliary rectangle is assigned with an initial geometry. Then, an intersection operation with the geometry of the container BLSE is applied. Weight specified to that container (see previous Section 3.1) is assessed to check if the process needs to be reapplied, using a larger or smaller rectangle, considering small dimension increments/decrements, in a loop that ends when sub-BLSE's area reaches a value closer to expected. Essentially, the transition of the intersection shape's area from bigger to smaller than required or otherwise defines the stop condition. This is particularly relevant to cover the cases where rectangular shapes intersect a building's constraint polygon, since the resulting geometries may not coincide with the original ones used for sectioning. Notwithstanding, Heron's formula is used in every situation to calculate the area of the resulting BLSE shapes through their internal triangles' decomposition.

The following set of mathematical/geometrical formalisms intends to synthetically detail the process, considering the 2D Cartesian coordinate system at the floor-plan level. Equation (1) refers to the placement of the initial rectangle, as previously mentioned:

$$\begin{cases} BLSE[i]_{rect.x} = BLSE[i-1]_{rect.x} + BLSE[i-1]_{rect.width} & \\ BLSE[i]_{rect.y} = BLSE[i-1]_{rect.y}, & \text{if x-axis division direction} \\ BLSE[i]_{rect.x} = BLSE[i-1]_{rect.x} & \\ BLSE[i]_{rect.y} = BLSE[i-1]_{rect.y} + BLSE[i-1]_{rect.height}, & \text{if y-axis division direction} \end{cases} \tag{1}$$

The initial rectangle *rect* created for an index *i* BLSE at a given depth assumes the initial position $(rect.x, rect.y)$, right next to the rectangle of the previous BLSE ($BLSE[i-1]$), both included in the same container BLSE. Such an operation takes the direction of an axis, like a sliding window that slices a certain floor-plan layout into BLSEs' shapes according to specified weights of occupation. The axis direction is inverted in each sectional depth, i.e., if a layout starts with the x-axis as the arrangement direction, initial BLSEs are placed in that sense, their eventual sub-BLSEs are divided following the y-axis direction, and so on. Auxiliary rectangles' dimensions are also set up, as shown in (2):

$$\begin{cases} BLSE[i]_{rect.width} = CBLSE_{mr.width} \times BLSE[i]_{relativeweight}, & \\ BLSE[i]_{rect.height} = CBLSE_{mr.height}, & \text{if x-axis division direction} \\ BLSE[i]_{rect.width} = CBLSE_{mr.width}, & \\ BLSE[i]_{rect.height} = CBLSE_{mr.height} \times BLSE[i]_{relativeweight}, & \text{if y-axis division direction} \end{cases} \tag{2}$$

The width (*rect.width*) and height (*rect.height*) of a rectangle initially shaping an index *i* BLSE are also defined according to both axis arrangement direction and occupation weight; attributed to or set up by back-propagation. As can be concluded, a given BLSE fits in a container, denoted as *CBLSE*, by using its minimum rectangle *mr* total width and, also, height, but multiplied by a weight-based factor (*relativeweight*), or the other way around, depending on arrangement direction. This is followed by an intersection operation, according to (3):

$$BLSE[i]_{shape} = BLSE[i]_{rect} \cap CBLSE[i]_{shape} \tag{3}$$

Eventually, full intersections will result in rectangular geometries for $BLSE[i]_{shape}$, but in those cases wherein this operation is carried out in the border of a building outlined by a non-rectangular shape, a polygon with variable edges is the likely outcome. To cover both situations, area is always calculated as the sum of the internal triangles regarding intersection shapes, using Heron's formula. Then, the resulting value is assessed to check if the weight of occupation of the respective BLSE geometry is larger or smaller than expected and, subsequently, a loop is set up to: (1) scale the BLSE's rectangle—i.e. $BLSE[i]_{rect.x|y|width|height}$—considering its previous dimension (Equation (4))using small increments/decrements; and (2) reapply intersections and area calculations formerly described, until a weight status transition—from smaller to larger or vice versa—stops the process.

$$BLSE[i]_{rect}(M) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The previous matrix definition $BLSE[i]_{rect}(M)$ is characterized by a scaling operation (by S_x, S_y) of the last rectangle defining an occupation area for a given BLSE, just before intersection with its container, considering short steps as, for example, 0.01 or 0.001, specifiable in the restriction rules. Equation (5) shows how this step is used to calculate the correct scale factor.

$$S_x = \frac{BLSE[i]_{rect.width}}{BLSE[i]_{rect.width} \pm step}, S_Y = \frac{BLSE[i]_{rect.height}}{BLSE[i]_{rect.height} \pm step} \tag{5}$$

It is noteworthy to inform that last index sub-BLSEs inside containers do not go through the same process, since the remaining space is directly allocated to them. The result of these operations over the complete set of BLSEs is a properly sectioned and arranged layout—still, without doors and windows.

Some building parts may be suppressed accordingly with the grammar-based definition rules, to achieve the desired building configuration (e.g., L/U-shape). This technique, which allows to specify disposable parts for further suppression based on a labeling approach, was designated as "fake-concave". When defining the rules, building parts can be marked as empty—height equals to 0—which means that they will be disposed after the floor-plan division. If there is an imposed definition to connect a pair of rooms belonging to different side-by-side *Building Part* containers, in which one of them is labeled to be discarded, such transition rule is ignored. This results in a geometric hole that changes the contour shape representing the constraints of the virtual building. "Fake- concave" technique allows to turn the delimiting geometry of a given floor-plan, initially assuming a convex form, into a concave shape, by cutting sections intersecting the starting perimeter.

After this contextualizing building's layout splitting process, next methodological steps using it can be addressed.

3.3. Step 2: Rules' Moderator

After data input, each grammar-based instruction is verified to check typos, i.e., unknown syntax and/or the structure (sequence conformity). Then, a structure of ontology-based class instances, representing a building's components, is loaded with the data extracted from the incoming rules, which takes place before the floor-plan probing responsible for ensuring weights distributions (Figure 3). While user rules must define a dimensioning label—e.g., small, medium, big—for each room, a related value specifying a growth factor is used as a reference from the restriction rules to guide the distribution process (e.g., small, medium, and big rooms grow with a rate of 1.5, 2, and 2.5, respectively, regarding a fixed value such as 0.001).

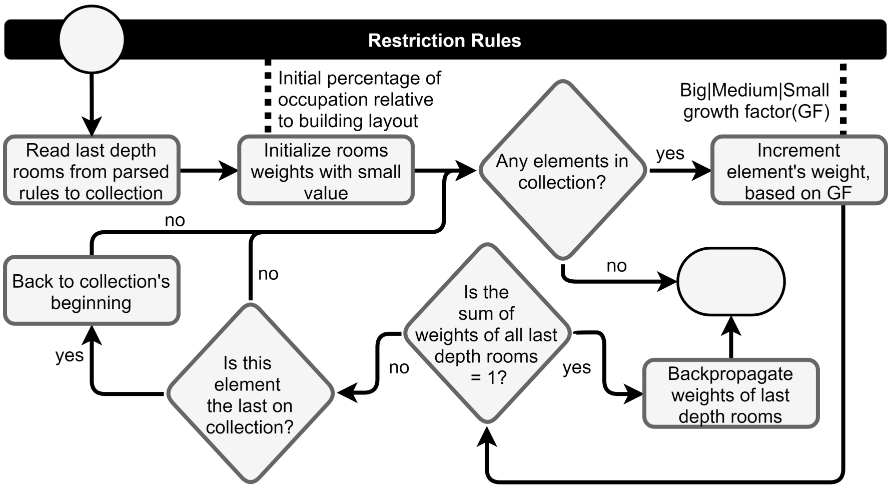

Restriction Rules
Read last depth rooms from parsed rules to collection
Initial percentage of occupation relative to building layout
Initialize rooms weights with small value
Any elements in collection?
yes
Big|Medium|Small growth factor(GF)
Increment element's weight, based on GF
no
Back to collection's beginning
no
no
Is the sum of weights of all last depth rooms = 1?
yes
Backpropagate weights of last depth rooms
yes
Is this element the last on collection?
no

Figure 3. Rooms' weights distribution: after the proper parsing of definition rules, the last depth rooms (related to the bottom nodes of a given treemap specification) are initialized with a small weight value (e.g., 0.001); for each one of these rooms, weight is incremented according to a growth factor imposed by restriction rules, in a loop that breaks when the sum of these rooms' weights reaches the value of one; back-propagation from deeper rooms to the most superficial containers is the last operation to be performed.

Still, within the mentioned probing context, containers/divisions undergo an arrangement process that, in practice, corresponds to the floor-plan's preliminary generation (detailed in previous Section 3.2, but with some additional weight adjustments to address the accommodation of transitions. More specifically, for a given wall portion shared by each pair of connecting rooms, geometric tests are performed to check if there is space to place a transition (door), whose dimensions comply with a percentage of the building's layout perimeter, established as a restriction rule. Transitions out of fit trigger a space transference process among divisions, named as weight borrowing, which seeks to solve those inconsistencies. When space collection reaches its limit, provided by a set of predefined, although editable, restriction rules, in a given couple of connected divisions, one of two events might happen: if the transition fits, it is considered valid, as exemplified in Figure 4; otherwise, borrowed space flows back to original lenders, and the testing transition is discarded to avoid malformed geometries in subsequent operations.

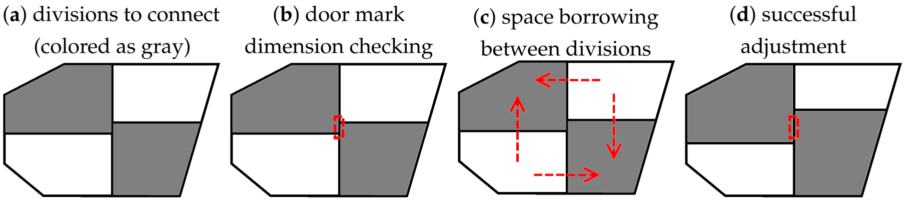

(**a**) divisions to connect (colored as gray)
(**b**) door mark dimension checking
(**c**) space borrowing between divisions
(**d**) successful adjustment

Figure 4. Space borrowing process for rooms requiring more wall of contact to place a transition: in (**a**), there is a couple of rooms that should be connected, whose areas are colored in gray; in (**b**), a transition is being submitted to a placement test; as it does not fit, an iterative space borrowing process takes place, as depicted in (**c**); finally, (**d**) denotes the result, in this case, a transition successfully placed between rooms flagged for connection.

Once again, in each space borrowing iteration, rooms' weights are back-propagated in a chain, overriding containers' weights with a partial contribution for an accumulated sum.

After ontology-based grammar rules' validation, weights' fine-tuning, as well as transitions inclusion/exclusion encompassing geometric consistency, effective floor-plan production takes place, as presented in the next step.

3.4. Step 3: Procedural Modeling Generation Process

After proper verification, validation and probing operations performed in the step addressed in previous subsection 3.3, a last execution of the process to split the building's layout—according to which is described in subsection 3.2—and to place validated transitions is carried out, resulting in the effective floor-plan that is eventually transformed by the "fake-concave" approach, if required. As a complement, a method to adapt the format of inner rooms was implemented, consisting in the direct manipulation of their outline segments—adding or deleting edges from 2D ground contours—accordingly to the provided definition rules. Such manipulation occurs with a strategic awareness that, in one hand, ensures rooms' geometric consistency and, on the other hand, avoids impacting on previously placed doors and windows. While adding a wall implies the incorporation of a new segment that is attached considering the midpoints of two consecutive room's segments with the larger lengths, walls' removal consists, precisely, in the opposite, i.e., deleting a point unifying a pair of consecutive room's segments, whose lengths are the shorter.

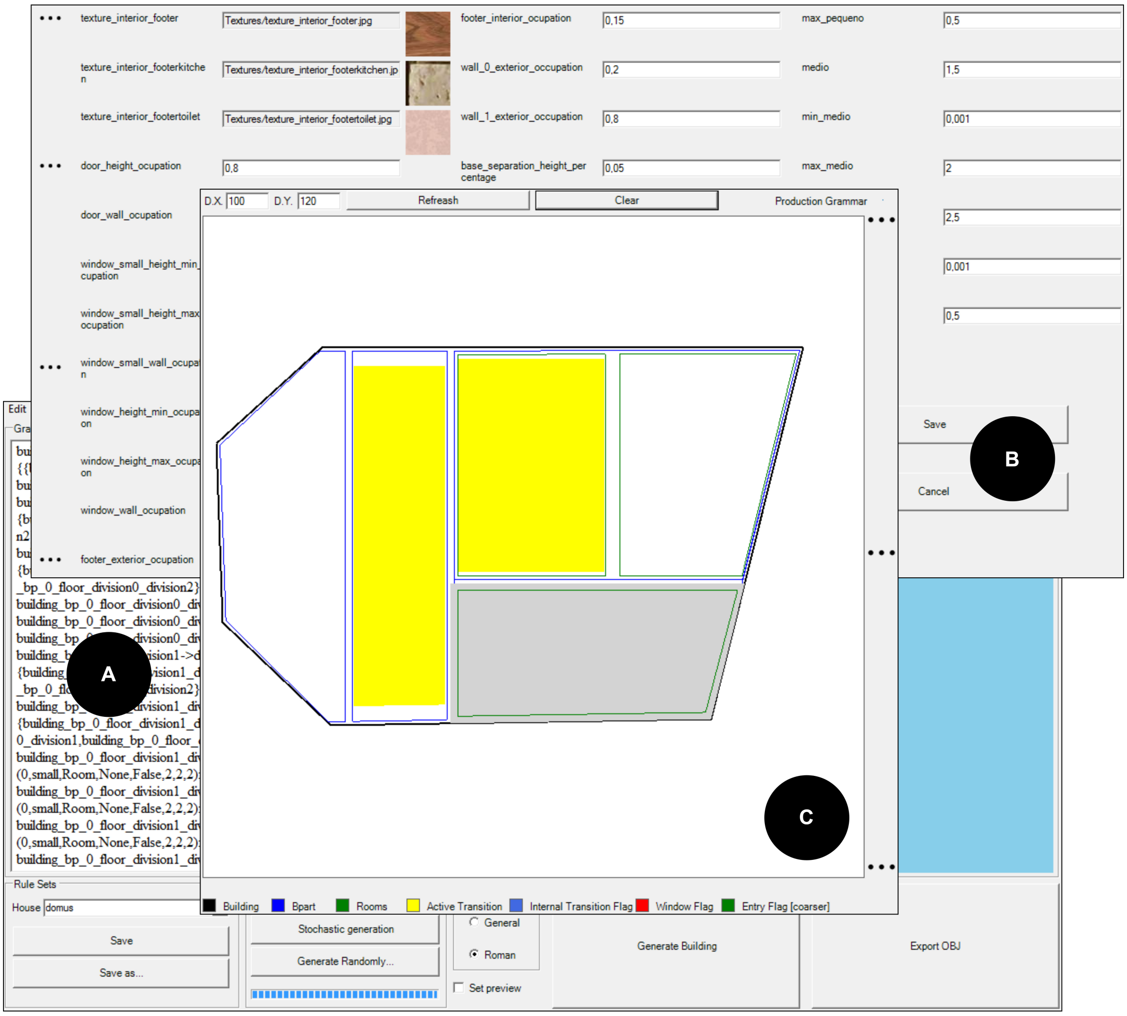
To accomplish the procedural modeling stage, the extrusion of the elements takes place to provide a 3D model definition. The process is highly based on the 2D contours of the previously built floor-plan. Thus, inner and outer walls are raised until a certain height, provided by input definition rules. Then, frames for doors and windows are created along with the roof. Some details such as roof skeleton coarsening, building base and division footers are also created to enhance the virtual model realism.

3.5. Step 4: Virtual Building Model Outcome

The final step regards the methodology outputs that result from the procedural modeling process (Step 3). There are two modes of accessing the output content, i.e., each resulting virtual model: (1) the preview mode provides a simplistic visualization of the generated virtual building model (without details), which is rendered directly in GUI; (2) the complete model mode allows the exporting of a generated virtual building (with full detail) to be later accessed using a professional CAD software for visualization or even manual editing purposes.

3.6. Bridging Users with the Methodology Through a Gui-Based Software

The ontology-based grammar constituting the rules' system of this procedural modeling methodology can be quite difficult and time-consuming to handle. Thereby, a GUI-based software wrapping this methodology has been developed and improved to provide, among other features, ontology-based grammar complexity isolation, support to simplistic previews over freshly-built virtual models, and exporting functionalities. To address the referenced complexity aspect, this software integrates an intuitive and interactive parametric tool to draw and parametrize floor-plans, allowing specialists such as archeologists and multimedia professionals to propose, test, and refine virtual buildings representing hypotheses, in a relatively dynamic process. This parametric tool is capable of generating grammatical rules from a short number of user interactions. Firstly, the building's outline contour must be defined along with a set of building parts that can be subdivided into different orientations for arrangement purposes. More specifically, if a building part has a horizontal orientation, user-defined partitions will result in vertically-arranged sub-parts and vice versa. To satisfy the grammar requirements, a roof type and a height need to be specified by the user for each building part, through a graphical prompt window. Heights equal to zero define respective building parts as disposable("fake-concave"), which ends up resulting in layouts with holes and/or non-convex configurations. Subsequently, through mouse clicking upon each building part, divisions (rooms or containers for structures of the same type) can be added and also subdivided to manage floor-plan organization, following the same previously-explained strategy. During a subdivision operation, the number of partitions to be accommodated inside a building part or a room needs to be specified, and then, for each partition, the following features must be parametrized through a graphical window prompt: desired number of delimiting walls, division type, number of windows, internal structure, and flag for entry doors (transitions interfacing the building exterior). This tool also supports the definition of the divisions' connectivity in a few simple steps, including: (1) right-clicking upon one of the available divisions, which prompts a context menu; (2) selecting the connectivity option from the triggered menu; and (3) clicking upon another available division to establish the connection rule. Lastly, it is important to stress out that parameters requested by the user along the whole floor-plan setup process are bound to operational contexts to prevent input errors and malformed grammatical rules, which could result in incoherent models. The implementation was carried out using Microsoft (Washington, United States) tools, the C# programming language, and the .NET and XNAframeworks. Figure 5 depicts a screen-shot of the working software.



texture_interior_footer Textures/texture_interior_footer.jpg
texture_interior_footerkitchen Textures/texture_interior_footerkitchen.jp
texture_interior_footertoilet Textures/texture_interior_footertoilet.jpg
door_height_ocupation 0,8
door_wall_ocupation
window_small_height_min_ cupation
window_small_height_max ocupation
window_small_wall_ocupa n
window_height_min_ocupa on
window_height_max_ocupa on
window_wall_ocupation
footer_exterior_ocupation
footer_interior_ocupation 0,15
wall_0_exterior_occupation 0,2
wall_1_exterior_occupation 0,8
base_separation_height_percentage 0,05
max_pequeno 0,5
medio 1,5
min_medio 0,001
max_medio 2
2,5
0,001
0,5
D.X. 100 D.Y. 120
House domus
○ General
● Roman
□ Set preview

Save
Cancel
Refreash
Clear
Production Grammar
Edit
Gra
Rule Sets
Save
Save as...
Stochastic generation
Generate Randomly...
Generate Building
Export OBJ
■ Building
■ Bpart
■ Rooms
■ Active Transition
■ Internal Transition Flag
■ Window Flag
■ Entry Flag [coarser]
_bp_0_floor_division0_division2}
building_bp_0_floor_division0_di
building_bp_0_floor_division0_di
building_bp_0_floor_division1->d
{building_bp_0_floor_division1_d
0_division1,building_bp_0_floor_
(0,small,Room,None,False,2,2,2)
A
B
C

Figure 5. Screenshot of the procedural modeling software: (**A**) is the main tool that enables (also, but not only) storing virtual models' grammar rules in text file format, to preview generated/produced models and to export them as wavefront (.obj) files; (**B**) is a menu interfacing with the restriction rules that allows extensive parametrization, from texture features to divisions' growth rates; (**C**) shows a graphical assistant that enables rapidly configuring floor-plans, according to user specifications. During each development step, grammar rules are automatically extracted (see right box), and at the end, they can be returned to the main tool that properly handles the 3D model production process.

The next section reviews the scientific progression of this methodology, by presenting its evolution regarding developmental stages and contributions for other projects/works.

4. Scientific Contributions and Development Stages of the Procedural Modeling Methodology

This procedural modeling methodology underwent several transformations since its very beginning in the ERAS project [18] until its current state [12]. Meanwhile, some other works benefiting from this methodology's capabilities were developed [20,25]. Both topics are addressed in the next subsections along with the methodology's latest developments.

4.1. Genesis: Virtual Reconstruction of Roman Buildings In Eras

The first developments regarding this procedural modeling methodology [12] started in the ERAS research project, which was an initiative to address the digital reconstruction of severely damaged ancient structures while ensuring reliable virtual representations generated expeditiously and without the need for extensive document analysis. Firstly, an ontology for traversable buildings was proposed to establish a common regulation structure [14], as previously explained in Section 3. In that specification, the following assumptions were made: (1) a building is composed of temporary building part containers (for floor-plan arrangement purposes) or final building parts; (2) each final building part contains at least one floor; (3) each floor is composed by divisions that can be temporary arrangement containers or final rooms; (4) transitions, such as doors and windows, ensure transitivity between floors and/or divisions and between the interior and exterior of the building; and (5) each building part has its own roof. The work in [13] carried out a deeper specification for the ERAS system to set up guidelines for implementation. Later, following most of these guidelines, ERAS's first tangible product was finally achieved with the support of a couple of constituting components: information extraction module and procedural modeling module [18].

Textual descriptions are converted into ontology-based data structures within a specific Roman architectural data model by the information extraction module. Parsing of Conímbriga's missing Roman buildings [113] descriptions is made based on more than 70 reusable grammatical rules developed resorting to Nooj [114]: a graphical editor that allows to identify text patterns and add notes to complement extracted information. Building parts, floors, divisions, transitions and other elements were mapped according to the aforementioned building ontology [14]. Geographic information system (GIS) provided additional data to complete the textual description about buildings' dimensions and locations [115]. The resulting internal organization of a building's divisions is mapped using a 2D matrix that contains each division's relative position on the building's floor-plan, as well as their connections. At the end of its operations, the information extraction module was expected to output an Extensible Markup Language (XML) file representing a building specification complying with the referenced ontology that also includes definitions regarding rooms' arrangement and transitions between them, in accordance with the already mentioned 2D matrix map.

The procedural modeling module (the first version of the methodology focused by this paper, whose implementation started in [15]) receives the XML file produced in the information extraction module and uses it to populate a structure of ontology-based class instances. Then, several steps were carried out under the responsibility of an architect class implemented in the module: floor-plan definition, wall extrusion, roof placement, and building's detailing. The first stage handles the building's restriction polygon definition, as well as the division of that area into several sub-areas, in an operation that follows a treemap approach to achieve the floor-plan. Afterwards, windows and doors are flagged considering the specification file and the rooms that interface with each other and with the buildings limits by an overlapping frontier (determined from the sub-areas resulting from previous operation). In the second stage, the walls are extruded considering the doors and windows previously flagged. Ground and ceilings are also placed in each room. Then, the geometry of the roof is created according to the type of roof defined for each building part. Flat, pyramidal, hipped, mansard, holed mansard, and none are the supported roof types. The final stage focuses on the enrichment of the building's virtual model by producing entrance's stairs, rooms' footers, frame portals for doors and windows, and other structures.

The whole process results in a virtual model representing the traversable building that can be observed using the modeling tool previewer. The model can also be exported into a wavefront file format (.obj) for portability and integrability with external CAD tools. To assess ancient buildings' reconstruction capabilities of ERAS system, a set of textual descriptions available on Conímbriga's website [113] were used. Representative, although hypothetical, models of the Skeleton house, Fountain house, Insula Aqueduct, and Via South stores (Conímbriga's ruins, Portugal) were successfully outputted. Some of them are depicted in Figure 6. As can be observed, roofs were not the most suitable to represent the desired covering structures for Roman houses. Although, this was the best compromise found between the recommendations of the consultant archeologist for the ERAS project and the capabilities of the procedural modeling tool version developed until that moment (more specifically, the second of three releases).

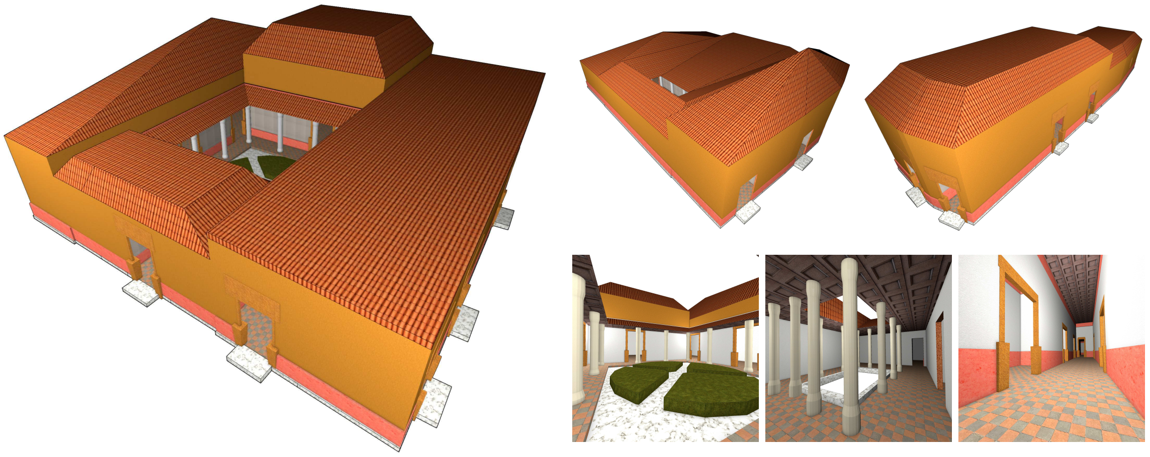


Figure 6. Reconstructed Roman buildings of Conímbriga, Portugal [18]: the first exposes Skeleton house; the second shows the reconstruction results of the Fountains house; the third presents Via South stores; finally, the last depicts the reconstruction results of the house designated by the Insula Aqueduct. These representations are just hypothetical models generated by an early version of the procedural modeling tool under analysis and do not intend to claim historical/architectural faithfulness.

Shortly after ERAS system implementation [15], minor adaptations were made for the procedural modeling part regarding data input format and validation, closing the first version implementation [16]. More specifically, a extended markup language for building definition (XML4BD) format was introduced to improve data input structuring and facilitate respective parsing/validation. At this development stage, supported shapes at the floor-plan level were not providing the desired flexibility, confining virtual buildings to be delimited by convex shapes. Besides, production was deterministic, and there was no way of specifying the number of walls to change the format of inner divisions. The next subsection explains how those problems were addressed and overcome to reach a more mature methodology.

4.2. Methodology Enhancements: Providing Shape Freedom at the Floor-Plan Level

Some enhancements were made to ERAS's procedural modeling version with the aim of making it capable of subdividing building layouts in geometries that were not confined to squares or convex forms. In the enhanced version, XML-based data input gave place to a rule system relying on an ontology-based grammar that aimed to allow the definition of virtual buildings in terms of working specifications for the implemented methodology. At the rules' definition time, building parts can be marked as empty (disposable).

Once the floor-plan definition is determined, a progressive process is triggered to produce the virtual building model. Firstly, the restriction polygon is forced to be convex; next, the floor-plan subdivision takes place (following the steps specified in Section 3) to split the virtual building polygon into smaller areas, according to ontology-based grammar specifications. If needed, some building parts are suppressed to comply with those specifications (or rules). This technique is designated by the term "fake-concave" and relies on a tree labeling approach to define disposable building parts. As result, it allows achieving buildings configured by non-convex shapes and geometric holes at the floor-plan level [12], as is shown in Figure 7.

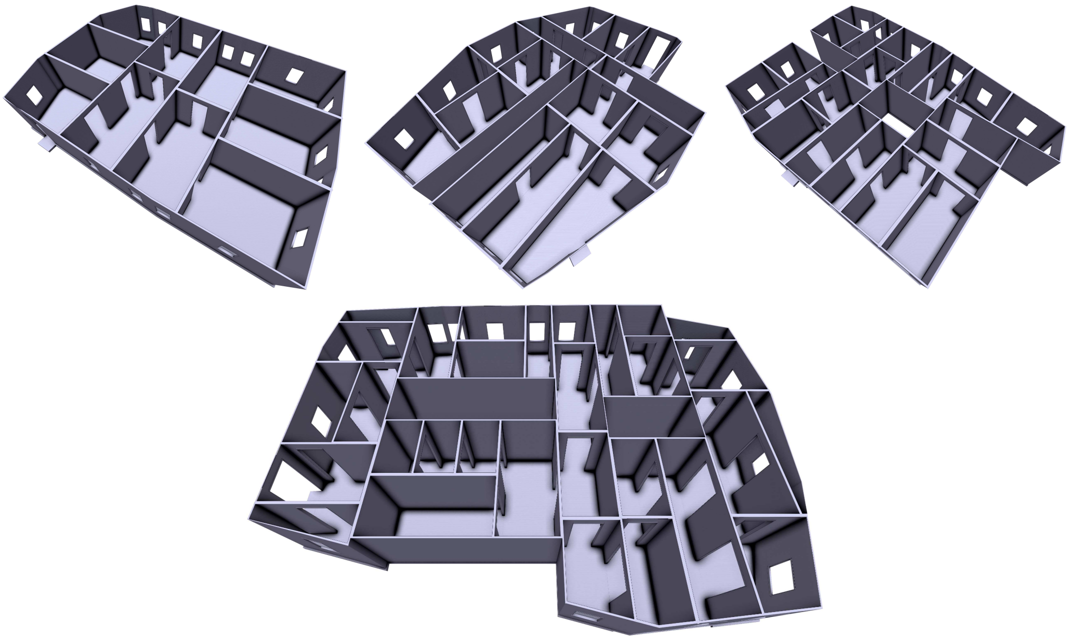


Figure 7. Floor-plans generated for testing purposes, after the procedural modeling methodology's subdivision capabilities enhancement [12].

Another step was introduced in the enhanced procedural modeling version (subsequent to ERAS project) to enable inner divisions' shape adaptation, as was already addressed in Section 3. This step accomplishes the geometric transformation of inner divisions, at the floor-plan level, to fulfil certain requirements of ontology-based grammar rules, more specifically the number of wall segments surrounding a given room. Therefore, some segments may be added or deleted from the rooms' shapes, after the floor-plan subdivision step and before the marking of transitions [12]. Afterwards, the process follows the regular flow until the complete virtual 3D model achievement, according to as was described previously, in Section 3.

The aforementioned developments focusing on the rule-based geometrical modeling of floor-plans contributed to overcome the previous (ERAS's) procedural modeling version's main limitation effectively, i.e., the production of buildings strictly confined to convex layouts. However, at this point, there was no way of producing buildings without exhaustive user parametrization. Thereby, a stochastic approach capable of producing virtual buildings automatically (or, at least, demanding much less user interaction) is addressed in the next subsection.

4.3. Methodology Enhancements: An Experimental Approach Towards Autonomous Building Generation

To overcome the need for exhaustive user parametrization required by this procedural modeling methodology, an experimental stochastic generation approach working as an higher abstraction layer was developed to provide autonomous computer-based building production. Specifically, two types of requests can be made with the proper input for random generation: on the one hand, a number of structures to be built can be provided to trigger the generation of stochastic-based virtual buildings; on the other hand, there is also the possibility of inputting a building configuration (L-shaped, U-shaped, or fully convex) along with a rooms' list to produce a single virtual building complying with imposed restrictions. Regardless the requested type, the proposed stochastic approach arranges the rooms (or divisions) by priorities, sets up the connected divisions' graph, and randomly arranges them in a treemap-like node structure. Therefore, this node structure is then converted into grammar rules that are used to produce a 3D virtual building model properly, using the procedural modeling methodology. The GUI-based software previously presented in Section 3 already implements this approach, enabling an expeditious generation of non-characterized virtual buildings derived from the main ontology [12].

In spite of the promising preliminary results of the experimental stochastic approach, the opportunity to have it work for other solutions has not shown up yet. Still, two other works in which the relevance of the procedural modeling methodology under analysis in this paper becomes clear are revisited in the next couple of subsections.

4.4. Methodology Contributions: Mixar Project and In Situ Virtual Reconstructions Seamlessly Aligned Upon Ruins

The importance of promoting general public participation in archeology was considered in MixAR, which is an MR project that enabled the development of a system, whose main goal is to manage the amount of reality and virtuality displayed in cultural heritage visitations. It is a solution capable of providing in situ visualization of reliable virtual reconstructions seamlessly integrated in the real scene, which can be freely explored by the visitor. To achieve it, a process that balances the blending level between reality and virtuality while the user is freely moving in an archeological site was developed. Thus, when the visitor is outside a virtual building (outdoor scene), an AR approach (in which the real world prevails) is used; if the visitor moves into a virtual building's interior, virtuality takes over the experience, placing the visitor inside a fully-synthesized environment. Transition occurs smoothly to provide a sensation of naturalness, as it was specified by [19].

More specifically, the MixAR system consists of a mobile application—compatible with Android-based tablet (Galaxy Tab PRO 8.4, Android 4.4 Kitkat)—that acquires and computes real-world context data using the tablet's global position system (GPS), inertial sensors, and camera to provide the proper MR experience, while the visitor moves within an archeological site. To that end, a CAD-based tracking approach was used to augment the real-world with ancient virtual buildings' hypotheses with satisfactory results [20]. The presented procedural modeling methodology was the key technology behind the production of such hypothesis. Essentially, virtual buildings were deterministically created using a parametric tool that assisted the user in the floor-plan design, which was used by the procedural modeling tool for further automatic 3D conversion.

Figure 8 provides an overview of the MixAR system, highlighting the three essential asset types required for MR visitations: procedurally-modeled augmentation models (produced using [12]), line models and respective visual aid elements (manually produced using Blender), and a configuration file to set up the CAD-based tracking approach. Such assets are submitted to the server database, making them accessible for being downloaded by visitors' mobile devices (e.g., smartphone or tablet). During visitation and after assets' download, a line model visual aid is shown in the display of the mobile device carried by the user; every time those lines match with the silhouette of the respective real-world structure (ruin or building in advanced degradation stage) that is being sought, i.e., successful tracking registration, the procedurally-modeled virtual building pops up as the augmentation model properly aligned and placed on the real-world target.

A recent study [116] pointed out that in spite of the absence of virtual shadow effects, the generated models were sufficiently convincing to induce acceptable levels of satisfaction among the testing participants, more specifically regarding presence and immersion. Respectively, 71 and 75 points were reached using the system usability scale (SUS) metrics.

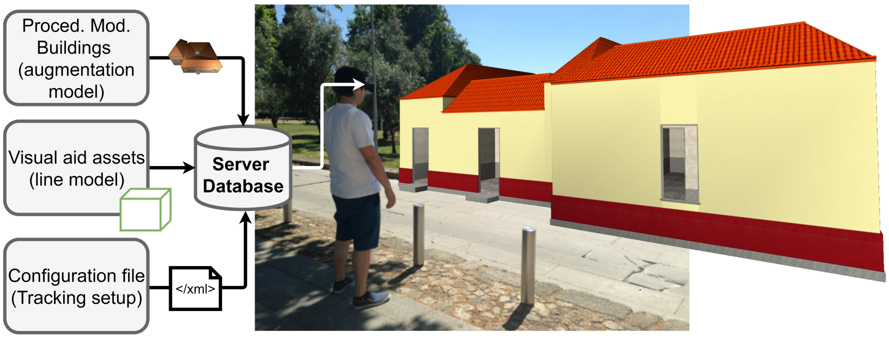

Proced. Mod. Buildings (augmentation model)
Visual aid assets (line model)
Configuration file (Tracking setup)
</xml>
Server Database

Figure 8. MixAR's system overview: procedural virtual buildings, line models, and respective visual aid elements, as well as a configuration file specifying a CAD-based tracking approach are set up and submitted to the MixAR server database. Those assets are then downloaded by the visitor's mobile device (on application start up) and used during the archeological MR experience proposed in MixAR [20]. The visual aid line model is firstly presented on the user device's display, and after successful tracking registration with the real-world structure, the assigned procedurally-modeled building is augmented in the scene.

4.5. Methodology Contributions: Recovering Lost Heritage in Faithful Virtual Environments Produced with Aerial-Based Photogrammetry

Acknowledging the adequacy of both UAS-based surveying and photogrammetry to produce accurate virtual reconstructions, as well as the potential of procedural modeling in generating reliable hypotheses concerning missing parts of history, a method articulating these approaches was proposed in [25]. It relies on a set of steps, including: aerial surveying and georeferencing of buildings' ruins within an archeological site, photogrammetric processing, production of procedural models representing missing buildings, and finally, seamless integration of all elements in a complete VR environment.

Firstly, a UAS flight mission gathered data about the site of interest. Then, using a photogrammetric process, data were converted to an accurate 3D model of the referenced place. By considering the geographic information that also results from the previous process, ruins were manually outlined for georeferencing purposes. To tackle ruins' missing information, virtual building models were produced using the procedural modeling approach in revisitation [12], more specifically through the respective interactive GUI-based parametric tool. Scaling, translation, and rotation operations do not have to be a concern of the user, who shall only ensure the virtual models' coherence by avoiding, for example, disparities between buildings' area extension and height or divisions mistakenly connected. The referenced geometric transformations were delegated to a final component responsible for setting up the full VR environment. In particular, it carried out (automatic) translation, rotation, and scaling operations, which aim to match procedurally-modeled buildings with the respective identified ruins. Such component was implemented in a fully-functional application developed in Unity (Unity Technologies, San Francisco, California, United States) for demonstration purposes. Figure 9 portrays the result of a virtual reconstruction combining photogrammetry and procedural modeling approaches, carried out considering the iconic Vila Velha ruins (Vila Real, Portugal) as a case-study.

Acknowledging the need to enhance this methodology's roof production step, a straight skeleton approach was implemented and will be presented in the following subsection.

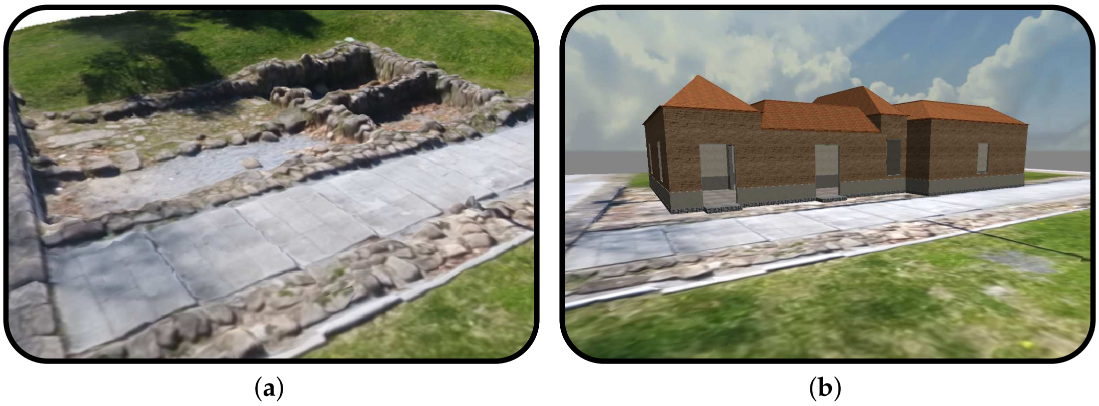


Figure 9. Vila Velha ruins' reconstruction case study [25]: (**a**) depicts a 3D model portion focusing on the Vila Velha ruins, which resulted from aerial imagery acquisition and photogrammetric processing; (**b**) shows a hypothetical virtual building (produced with the procedural modeling tool [12]) accurately and automatically placed upon the Vila Velha's virtual ruins' representation. The latter stands as a hypothetical outcome produced by the procedural modeling tool under analysis and does not intend to claim historical/architectural faithfulness.

4.6. Methodology Enhancements: Moving Towards Convincing Roofs Generation with the Straight Skeleton Approach

Roofs have been an issue of this procedural modeling methodology inasmuch as, until now, each virtual building's part only supported an isolated top cover, i.e., there was no way of merging the roofs to become a single one, while producing virtual buildings composed of two (or more) main parts arranged next to each other. Consequently, some of the outcomes looked like individual and unnaturally-stitched building parts' compositions when, in fact, they belonged to the same containing virtual building. In recent developments, that issue was addressed with the straight skeleton algorithm [26–28], which seems to be a suitable approach for producing visually-reliable roofed buildings, as was done before with good results [66,110]. The original approach has been adapted for this methodology taking into consideration the geometric independence of the building parts flagged to share a common roof, and as long as they have some overlapping walls, an auxiliary intersection shape is created (based on ear clipping triangulation) as a common roof layout, wherein the straight skeleton is built up.

In terms of process, the intention of merging two buildings must be specified in the ontology-based rules (detailed in [12,17]) that define a building. Thereby, the "HipGlued" keyword needs to be present in every building part whose roof merging is required, as is shown in the following snippet:

```
...
bpart0->buildingpart(HipGlued,10):{...}
...
bpart1->buildingpart(HipGlued,10):{...}
...
bpart2->buildingpart(Hip,10):{...}
...
```

In the example above, there are three parts defining a building: *bpart0*, *bpart1*, and *bpart2*. For *bpart0* and *bpart1*, it is defined that they should share the same roof, since they both have the "HipGlued" parameter for roof type definition. On the other hand, *bpart2* does not share its roof with any other building part. Instead, an isolated hip cover will be created upon it. It is noteworthy to clarify that transitions among building parts that do not share the same roof are still supported. For instance, divisions within *bpart0* and/or *bpart1* may have passage doors for rooms belonging to *bpart2*, unless eventual geometric operations carried out during floor-plan division restrict the placement of such transitions (e.g., by the lack of adjacent wall).

After defining the rules and submitting them to procedural modeling, the set of processes run regularly (as described in Section 3) until the roof production stage, which is now capable of applying the straight skeleton algorithm whenever it is demanded by rules' specifications. For the already documented roof types, the process follows as reported in [12,17]. On the other hand, for all the building parts labeled to hold straight skeleton roof (through the introduced "HipGlued" parameter), a geometric test is carried out to check which of them are in spatial contact with others, at the ground level. Relying on that test, groups of contacting building parts are set up. For each one of these groups, ear-clipping-based polygons (loaded and processed with the complete list of vertices belonging to the ground-level shapes of the building parts integrating a given group) are also built. Further operations involve the application of the straight skeleton algorithm that returns the required 3D roofs to cover the groups of contacting building parts. Regarding the freshly mentioned algorithm itself, essentially, it can be defined as a shrinking process using the edges formed by an initial polygon, in which split events may occur whenever a vertex collides with an edge. The resulting straight skeleton is formed by the set of curves configured by vertices displacement. Figure 10 presents four generated straight skeleton roofs, for validation purposes and a better understanding of the described algorithm goal. During parametrization, roof compositions (belonging to different building parts) were wittingly set with the "HipGlued" parameter to ensure a fully-merged generation.

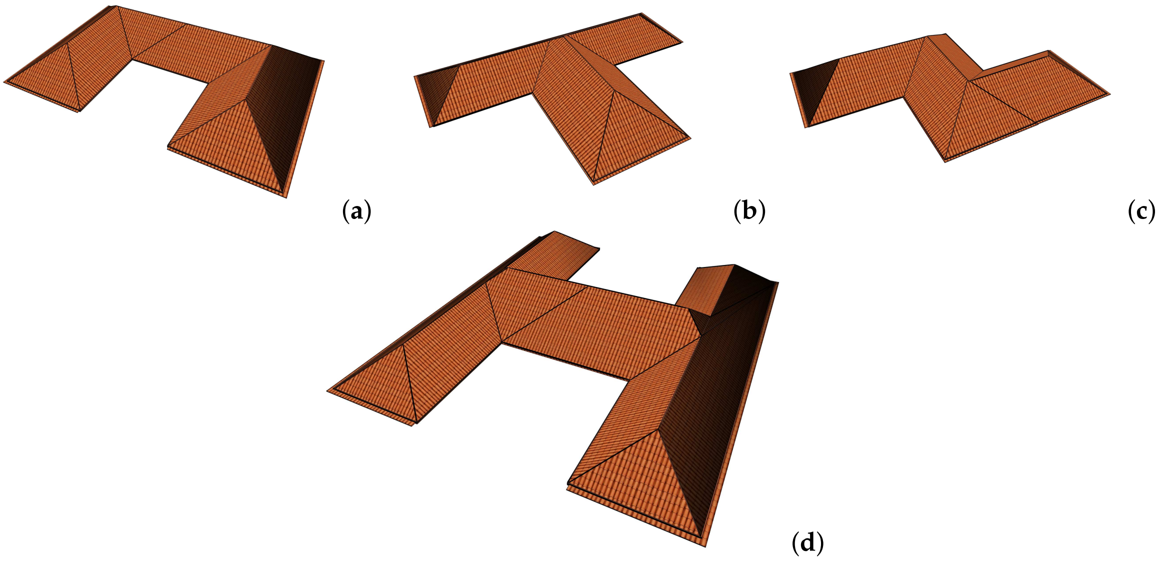


Figure 10. Roofs generated with the methodology's straight skeleton implementation: (**a**) consists of a U-shaped roof that connects three roof components; (**b**) is a T-shaped roof composed by a couple of roof components; (**c**,**d**) respectively depict S- and H-shaped, both of them connecting three roof components.

In the end, the virtual building enhancements step takes place to produce details in the building, in accordance with what was mentioned in Section 3.

4.7. Computational Performance

An evaluation considering average load per CPU core, RAM usage, time, and number of generated vertices was carried out, focusing the impact of the methodology's implementation over computational resources. It took place in a computer whose specifications include an Intel^{R} Core^{TM} i7 CPU (Model M620) running at 2.67 GHz, as well as 4 GB of RAM capacity.

Specifically, and as presented in Table 2, tests encompassed the production of five virtual buildings groups, each one varying the number of both building parts and divisions, in a sequential and progressive increment.

Table 2. Performance tests' schedule, gradually incrementing the number of building parts per generation group and the number of rooms per building part, totaling 25 generations.

Building Parts	Rooms' Range
1	[3–7]
2	[5–9]
3	[7–11]
4	[9–13]
5	[11–15]

Figure 11 depicts these tests' results. Average CPU usage stabilized between 37 and 61% during the tests. The interval of RAM consumption was about [40, 45] MB. It is also remarkable that the most complex structure, with 25,728 vertices, took no longer than 125 ms to be generated.

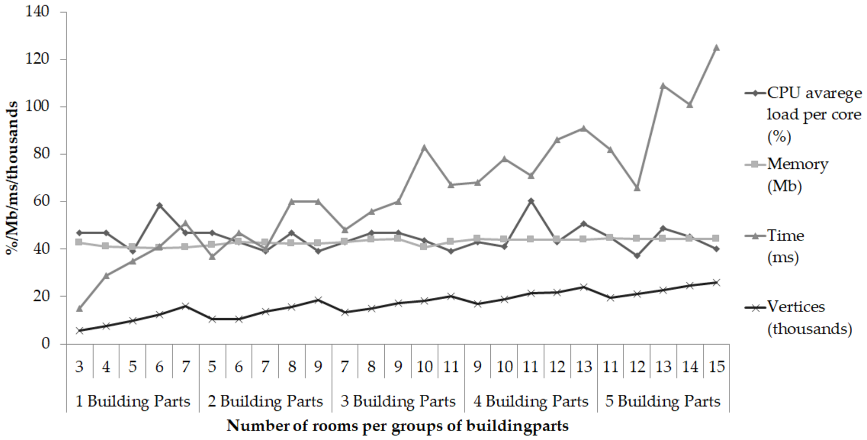


Figure 11. Chart depicting the performance tests performed on five groups of virtual buildings with a gradual increment of building parts per group and divisions per generation [12,17].

Next, the final section addresses future challenges and conclusions, wherein forthcoming directions regarding the authors' procedural modeling research line are discussed.

5. Conclusions and Future Challenges

The procedural modeling approach proposed by [12] to deal with the production of traversable buildings composed of floor-plans was revisited in this paper, and its relevance for the accomplishment of other research project was highlighted. The first version of this procedural modeling approach resulted from the ERAS project, which aimed at the virtual reconstruction of Roman buildings based on textual descriptions [14,16,18]. Later, an enhanced version was integrated in the building's production process of MixAR, which refers to a cultural heritage MR project for providing visualization over virtual reconstructions seamlessly aligned upon ruins [19–22]. Recently, another work for cultural heritage aiming at ruins' virtual reconstructions within full VR environments produced using UAS surveys and photogrammetry [25] also benefited from this procedural modeling approach. Along the way, a doctoral thesis [17] and two Master's dissertations [23,24] were also accomplished. Meanwhile, recent enhancements to enable straight skeleton-based roofs' production were implemented, allowing the optional merging of two or more building coverings into a single one with the aim of improving realism. A table summarizing the addressed methodology's progress is presented in Table 3.

Table 3. Summary of the progress made during the development of the revisited procedural modeling methodology [12], from the preliminary to final version.

Version Topic	Preliminary Version	Enhanced Version	Final Version
Rules' system and format	Extensive and complex grammar-based rules following the L-system logical sequence; it also includes several (detachable) restriction rules.	XML4BD, also inspired by L-system (hierarchic organization enables building treemaps defining buildings' structures).	Ontology-based grammar rules, inspired by L-system; GUI hiding grammar complexity.
Restriction rules	Semantic mapping of structures to categories corresponding to general ontology elements (generalization).	Complementary set of default parameters defining building components' textures and dimensions.	Essentially, the same set of parameters of previous version, but accessible and editable through GUI.
Rules' compiling and pre-testing	None.	XML4BD structure validation, parsing, and fitting into ontology-based classes, growth, and connectivity pre-test.	Adapted from the previous version to deal with the ontology-based grammar.
Supported shapes	Rectangles of variable size for different building parts.	Convex delimitations constraining buildings, with an impact on border rooms.	"Fake-concave" delimitations constraining buildings; convexly-arranged constraining walls for inner rooms.
Floor plan generation strategy	Rectangular divisions based on the pure treemap approach, considering relative occupation weights.	Rooms' weights balanced distribution combined with a treemap approach adapted to work with convex shapes that are triangulated to control expansions, through Heron's formula.	Flagging approach enabling one to consider or discard building parts to support "fake-concave" building delimitations; rooms' shaping through user-defined grammar parameters.
Basic roof types	"Hip" type with a central descending hole (openings for gardens and water collection purposes, typically Roman).	Support to "Flat", regular "Hip", "Mansard", "None" and "Pyramid" types.	Roof type based on the straight skeleton approach.
Special structures' support	Water collection system, *peristylium*, without garden and columns, derived from the general to the Roman ontology.	Garden and completions: doors and windows frames.	Mainly addressing roofs' enhancement (flaps' and edges' thickening operations).
Contributions and related works	Buildings ontology [14]; virtual reconstruction for archeological sites' approach proposal [13]; textual extraction through natural language processing [115]; coarse generation of ancient Roman houses [15].	Combination of natural language processing with procedural modeling for ERAS [18]; convex shapes support [16]; contributions for MixAR [20–22] and related master theses [23,24].	Doctoral thesis [17] and book [12], documenting this procedural modeling methodology (including support to non-convex shapes outlining buildings and rooms' shaping); a combination of this methodology with photogrammetry to set up enhanced VR environments [25].

A small village composed of both manually-produced and procedurally-generated virtual models simulating the reconstruction of a fictitious archeological site with Roman style characteristics is presented in Figure 12, to exemplify an integrated digital environment built up with the support of the proposed methodology. While former models were acquired from online platforms—CGTrader (Vilnius, Lithuania) and Free3D (TurboSquid, Inc., New Orleans, Louisiana, United States of America)—latter ones were integrated without additional restructuring tasks. Only translations, rotations, and scaling operations were used to place and resize both types properly.

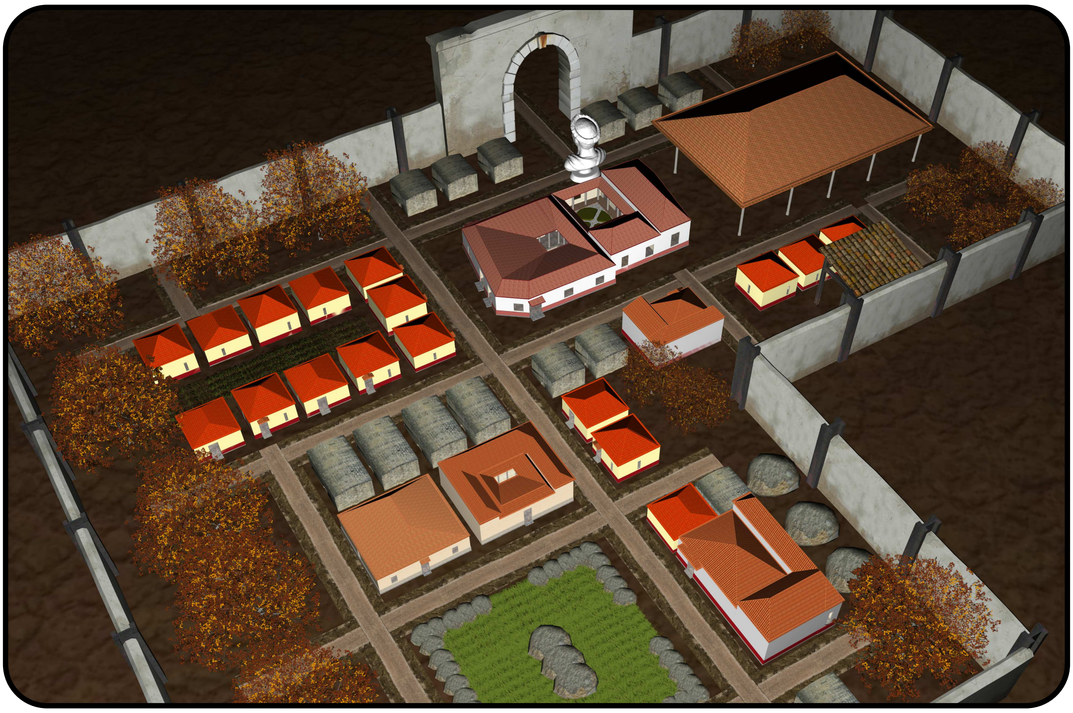


Figure 12. Synthesized village partially styled with Roman features, built up combining manually-modeled and procedurally-generated virtual elements: arched entrance, delimiting walls, grass, rocks, statues, streets, trees, and warehouses were retrieved from online platforms (CGTrader and Free3D); remaining (traversable) elements, namely habitation buildings topped with different roof types (widespread within the walled area), greater *domus*, and large covering structure supported by columns (both nearby the village's entrance) were generated through the GUI-based software that was developed to implement the proposed methodology. Historical/architectural faithfulness is not claimed.

It is important to highlight that this methodology aims to supply specialists such as archeologists and multimedia professionals (AR/VR) with tools to enable the iterative and free-style development of coarse virtual buildings towards reliable hypotheses, rather than restraining options through architectonic-style rules' imposition. In most of the projects that benefited from the revisited procedural modeling methodology, a trial-and-error approach to reach the virtual buildings was followed. For example, while in ERAS, some of the proposed hypotheses relied on the supervision of an experienced archeologist, the virtual models generated for MixAR were evaluated by AR/VR developers, through online documentation support.

As a short discussion note, it is noteworthy to mention that many procedural modeling solutions for the production of virtual buildings with interiors have been proposed (e.g., [90,99,101,117,118]) with remarkable supporting processes (e.g., computer-generated floor-plans) and results. Although, addressed buildings' geometries mostly focus squared/rectangular shapes. Others approaches [103,105] relying on CSG algorithms and generative modeling languages have been proposed to produce complex structures that, besides requiring programming skills, are usually very time consuming and oriented to rectangular/squared-based geometries. The work developed by Dahl and Rinde [91] admittedly addresses shapes' arbitrariness on traversable buildings. However, the interior divisions' generation carried out by the algorithm does not seem to be appropriate when rooms' generation control (for example geometric constraints and rooms' arrangements) is required. Dahl and Rinde [91] also reported the lack of support to produce geometric holes modeling building limits. The visual distinction between room types is hamper by the absence of texturing features.

Despite the accomplishment of the proposed objectives for the procedural modeling work documented in [12], there is still much space for improvement, namely in the following aspects: support to multiple floors, integration of pure concave geometries, reuse/re-adaptation of division's dead zones, i.e., space gaps resulting from inner and outer walls' manipulation, and automatic inclusion of furniture. Moreover, proper interfaces must be extended to allow collaborative work among the community of archeologists and related professionals, in particular towards proposals and refinements of virtual reconstructions.

As part of future work as well, efforts for the integration of more effective automatisms for virtual models' procedural production/generation will be developed, taking advantage of promising emerging machine learning solutions. Among the approaches are adversarial networks (GANs), which have the interesting ability to synthesize outputs within a given context, based on a specific problem-oriented dataset. Strategies to build ontology-based GANs and studies to assess the quality of floor-plans and 3D meshes generated through this kind of approach must be carried out. Enabling high-level user-defined restrictions on GANs' action is a challenge that should also be addressed.

Author Contributions: Conceptualization, T.A. and L.M.; methodology, T.A.; software, T.A. and L.P.; validation, T.A., L.P. and P.M.; formal analysis, L.M. and J.J.S.; investigation T.A. and P.M.; resources, L.M. and E.P.; data curation, T.A. and E.P.; writing—original draft preparation, T.A. and E.P.; writing—review and editing, T.A., L.M. and E.P.; visualization, T.A. and J.J.S; supervision, L.M. and E.P.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. De Santis, R.; Gloria, A.; Viglione, S.; Maietta, S.; Nappi, F.; Ambrosio, L.; Ronca, D. 3D laser scanning in conjunction with surface texturing to evaluate shift and reduction of the tibiofemoral contact area after meniscectomy. *J. Mech. Behav. Biomed. Mater.* **2018**, *88*, 41–47.
2. Brusaporci, S. *Handbook of Research on Emerging Digital Tools for Architectural Surveying, Modeling, and Representation*; IGI Global: Hershey, PA, USA, 2015.
3. Achille, C.; Adami, A.; Chiarini, S.; Cremonesi, S.; Fassi, F.; Fregonese, L.; Taffurelli, L. UAV-based photogrammetry and integrated technologies for architectural applications?Methodological strategies for the after-quake survey of vertical structures in Mantua (Italy). *Sensors* **2015**, *15*, 15520–15539.
4. Gellrich, N.C.; Schramm, A.; Hammer, B.; Rojas, S.; Cufi, D.; Lagrèze, W.; Schmelzeisen, R. Computer-assisted secondary reconstruction of unilateral posttraumatic orbital deformity. *Plast. Reconstr. Surg.* **2002**, *110*, 1417–1429.
5. Tepper, O.M.; Sorice, S.; Hershman, G.N.; Saadeh, P.; Levine, J.P.; Hirsch, D. Use of virtual 3-dimensional surgery in post-traumatic craniomaxillofacial reconstruction. *J. Oral Maxillofac. Surg.* **2011**, *69*, 733–741.
6. Sing, K.H.; Xie, W. Garden: A mixed reality experience combining virtual reality and 3D reconstruction. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; pp. 180–183.
7. Augusto, I.; Monteiro, D.; Girard-Dias, W.; dos Santos, T.O.; Belmonte, S.L.R.; de Oliveira, J.P.; Mauad, H.; da Silva Pacheco, M.; Lenz, D.; Bittencourt, A.S.; et al. Virtual Reconstruction and Three-Dimensional Printing of Blood Cells as a Tool in Cell Biology Education. *PLoS ONE* **2016**, *11*, e0161184.
8. Zollikofer, C.P.; de León, M.S.P.; Lieberman, D.E.; Guy, F.; Pilbeam, D.; Likius, A.; Mackaye, H.T.; Vignaud, P.; Brunet, M. Virtual cranial reconstruction of Sahelanthropus tchadensis. *Nature* **2005**, *434*, 755.
9. Guidi, G.; Russo, M.; Angheleddu, D. 3D survey and virtual reconstruction of archeological sites. *Digit. Appl. Archaeol. Cult. Herit.* **2014**, *1*, 55–69.

10. Dylla, K.; Frischer, B.; Müller, P.; Ulmer, A.; Haegler, S. Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques. In Proceedings of the 37th Proceedings of the CAA Conference, Williamsburg, VA, USA, 22–26 March 2010; pp. 62–66.
11. Magalhães, L.G.; Adão, T.; Peres, E. Reconstructing the Past: Providing an Enhanced Perceptual Experience. In *Handbook of Research on Technological Developments for Cultural Heritage and eTourism Applications*; IGI Global: Hershey, PA, USA, 2018; pp. 140–172.
12. Adão, T.; Magalhães, L.; Peres, E. *Ontology-Based Procedural Modeling of Traversable Buildings Composed by Arbitrary Shapes*, 1 ed.; *Springer Briefs in Computer Science*; Springer International Publishing: Cham, Switzerland, 2016; Volume 1.
13. Adão, T.; Magalhães, L.; Bessa, M.; Barreira, J.; Melo, M.; Gonçalves, M.; Sousa, J.; Peres, E. Proposal of an Information System for a Semi-automatic Virtual Reconstruction of Archeological Sites. *Procedia Technol.* **2012**, *5*, 566–574.
14. Adão, T.; Magalhães, L.; Bessa, M.; Coelho, A.; Sousa, A.; Rodrigues, N.; Rodrigues, R.; Pereira, F.; Moura, J.; Reis, L.P. ERAS—An Ontology-Based Tool for the Expeditious Reconstruction of Virtual Cultural Heritage Sites. In Proceedings of the Actas do 20° Encontro Português de Computação Gráfica, Viana do Castelo, Portugal, 24–26 October 2012.
15. Adão, T.; Magalhães, L.; Peres, E. Semi-automatic virtual reconstruction of ancient Roman houses. In Proceedings of the 2013 8th Iberian Conference on Information Systems and Technologies (CISTI), Lisboa, Portugal, 19–22 June 2013; pp. 1–6.
16. Adão, T.; Magalhães, L.; Peres, E.; Pereira, F. Procedural Generation of Traversable Buildings Outlined by Arbitrary Convex Shapes. *Procedia Technol.* **2014**, *16*, 310–321.
17. Adão, T. Ontology-Based Procedural Modeling of Traversable Buildings Composed by Arbitrary Shapes. Ph.D. Thesis, University of Trás-os-Montes e Alto Douro, Vila Real, Portugal, 2015.
18. Adão, T.; Batista, R.; Peres, E.; Magalhães, L.G.; António, C. Reconstructing traversable buildings for archeology with ERAS. In Proceedings of the Virtual and Networked Organizations: Emergent Technologies and Tools—ViNOrg '13, Póvoa de Varzim, Portugal, 20–22 November 2013.
19. Magalhães, L.G.; Sousa, J.J.; Bento, R.; Adão, T.; Pereira, F.; Filipe, V.; Peres, E. Proposal of an Information System for an Adaptive Mixed Reality System for Archaeological Sites. *Procedia Technol.* **2014**, *16*, 499–507.
20. Narciso, D.; Pádua, L.; Adão, T.; Peres, E.; Magalhães, L. MixAR Mobile Prototype: Visualizing Virtually Reconstructed Ancient Structures In Situ. *Procedia Comput. Sci.* **2015**, *64*, 852–861.
21. Pádua, L.; Narciso, D.; Adão, T.; Cunha, A.; Peres, E.; Magalhães, L. Cost-effective and lightweight mobile units for MixAR: A comparative trial among different setups. *Procedia Comput. Sci.* **2015**, *64*, 870–878.
22. Pádua, L.; Adão, T.; Narciso, D.; Cunha, A.; Magalhães, L.; Peres, E. Towards Modern Cost-effective and Lightweight Augmented Reality Setups. *Int. J. Web Portals (IJWP)* **2015**, *7*, 33–59.
23. Pádua, L.F.M. ULTRA-light and Low-Cost Mobile Augmented Reality Units. Master's Thesis, Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal, 2015.
24. Narciso, D.G. Técnicas de tracking sem marcas em aaplicações de realidade mista. Master's Thesis, Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal, 2016.
25. Adão, T.; Pádua, L.; Hruška, J.; Peres, E.; Sousa, J.J.; Morais, R.; Magalhães, L.G. Bringing together UAS-based land surveying and procedural modeling of buildings to set up enhanced VR environments for cultural heritage. In Proceedings of the EPCGi'2017—24th Portuguese Meeting of Computer Graphics and Interaction, Guimaraes, Portugal, 12–13 October 2017; IEEE Xplore: Guimarães, Portugal, 2017.
26. Felkel, P.; Obdrzalek, S. Straight skeleton implementation. In Proceedings of the Spring Conference on Computer Graphics, Budmerice, Slovakia, 12–14 May 1998.
27. Eppstein, D.; Erickson, J. Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions. In Proceedings of the Fourteenth Annual Symposium on Computational Geometry, Minneapolis, MN, USA, 7–10 June 1998; ACM: New York, NY, USA, 1998; pp. 58–67.
28. Laycock, R.G.; Day, A. Automatically generating large urban environments based on the footprint data of buildings. In Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, Seattle, WA, USA, 16–20 June 2003; pp. 346–351.
29. Belle, I. The architecture, engineering and construction industry and blockchain technology. *Digit. Cult.* **2017**, *2017*, 279–284.

30. Bernardes, P. Arqueologia Urbana e Ambientes Virtuais: Um Sistema Para Bracara Augusta. Master's Thesis, Universidade do Minho, Braga, Portugal, 2002.
31. Gonçalves, A.J.M. Reconstrução de ambientes históricos utilizando VRML: O caso do Fórum Flaviano de Conimbriga. Master's Thesis, Universidade de Coimbra, Coimbra, Portugal, 2002.
32. da Câmara, A.G.; Murteira, H.; Rodrigues, P. City and spectacle: A vision of pre-earthquake Lisbon. In Proceedings of the 2009 15th International Conference on Virtual Systems and Multimedia, Vienna, Austria, 9–12 September 2009; pp. 239–243.
33. Matini, M.R.; Ono, K. *Accuracy Verification of Manual 3D CG Reconstruction: Case Study of Destroyed Architectural Heritage, Bam Citadel*; Digital Heritage; Ioannides, M., Fellner, D., Georgopoulos, A., Hadjimitsis, D.G., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 432–440.
34. Andaroodi, E.; Matini, M.R.; Ono, K. Post Disaster Virtual Revival: 3D CG Manual Reconstruction of a World Heritage Site in Danger. In *Virtual Reality-Human Computer Interaction*; InTech: Vienna, Austria, 2012.
35. Lebres, F. Reconstrução fidedigna da vila velha no séc. XIII. Master's Thesis, Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal, 2011.
36. Razuvalova, E.; Nizamutdinov, A. Virtual reconstruction of cultural and historical monuments of the Middle Volga. *Procedia Comput. Sci.* **2015**, *75*, 129–136.
37. Molinero-Polo, M.; Hernández, C.; Méndez-Rodríguez, D.; Naranjo, T.; Díaz, Y.; Pérez-Ruiz, S.; Acebo, A.; Molins-Ruano, P.; Jurado, F.; Rodríguez, P.; et al. Bringing Egyptology to the classroom: Virtual reconstruction of the TT 209 site. In Proceedings of the 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), Madrid, Spain, 24–26 February 2016; pp. 306–310.
38. Deggim, S.; Kersten, T.; Lindstaedt, M.; Hinrichsen, N. The Return Of The Siegesburg–3D-Reconstruction of a Disappeared and Forgotten Monument. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 209.
39. Grossmann, E.; Santos-Victor, J. Least-squares 3D reconstruction from one or more views and geometric clues. *Comput. Vis. Image Underst.* **2005**, *99*, 151–174.
40. Pollefeys, M.; Van Gool, L.; Akkermans, I.; De Becker, D.; Demuynck, K. A Guided Tour to Virtual Sagalassos. In Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage, Glyfada, Greece, 28–30 November 2001; ACM: New York, NY, USA, 2001; pp. 213–218.
41. Agarwal, S.; Furukawa, Y.; Snavely, N.; Simon, I.; Curless, B.; Seitz, S.M.; Szeliski, R. Building Rome in a Day. *Commun. ACM* **2011**, *54*, 105–112.
42. De Reu, J.; De Smedt, P.; Herremans, D.; Van Meirvenne, M.; Laloo, P.; De Clercq, W. On introducing an image-based 3D reconstruction method in archaeological excavation practice. *J. Archaeol. Sci.* **2014**, *41*, 251–262.
43. Koutsoudis, A.; Vidmar, B.; Ioannakis, G.; Arnaoutoglou, F.; Pavlidis, G.; Chamzas, C. Multi-image 3D reconstruction data evaluation. *J. Cult. Herit.* **2014**, *15*, 73–79.
44. Bolognesi, M.; Furini, A.; Russo, V.; Pellegrinelli, A.; Russo, P. Testing The Low-Cost RPAS Potential In 3d Cultural Heritage Reconstruction. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, doi:10.5194/isprsarchives-XL-5-W4-229-2015.
45. Dhonju, H.; Xiao, W.; Sarhosis, V.; Mills, J.; Wilkinson, S.; Wang, Z.; Thapa, L.; Panday, U. Feasibility study of low-cost image-based heritage documentation in Nepal. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 237.
46. Pierdicca, R. Mapping Chimu's settlements for conservation purposes using UAV and close range photogrammetry. The virtual reconstruction of Palacio Tschudi, Chan Chan, Peru. *Digit. Appl. Archeol. Cult. Herit.* **2018**, *8*, 27–34.
47. Lira, J.; López, P.; Rodríguez, A. Detection of Maya's archaeological sites using high resolution radar images. *Int. J. Remote Sens.* **2005**, *26*, 1245–1260.
48. Challis, K. Airborne laser altimetry in alluviated landscapes. *Archaeol. Prospect.* **2006**, *13*, 103–127.
49. Rodrigues, N.C.S. Visualização de superfícies dinâmicas. Master's Thesis, Universidade Nova de Lisboa, Lisbon, Portugal, 2005.
50. Allen, P.K.; Troccoli, A.; Smith, B.; Murray, S.; Stamos, I.; Leordeanu, M. New methods for digital modeling of historic sites. *IEEE Comput. Graph. Appl.* **2003**, *23*, 32–41.
51. Beraldin, S.E.H.J.A.; Gonzo, L.; Whiting, E.; Jemtrud, M.; Valzanoe, V. A Hierarchical 3d Reconstruction Approach for Documenting Complex Heritage Sites. In Proceedings of the XX CIPA International Symposium, Torino, Italy, 26 September–1 October 2005.

52. Kampel, M.; Mara, H.; Sablatnig, R. Automated investigation of archaeological vessels. In Proceedings of the 14th European Signal Processing Conference, Florence, Italy, 4–8 September 2006; pp. 1–5.
53. Mohammed Oludare, I.; Pradhan, B. A decade of modern cave surveying with terrestrial laser scanning: A review of sensors, method and application development. *Int. J. Speleol.* **2016**, *45*, 8.
54. Rodríguez-Gonzálvez, P.; Jiménez Fernández-Palacios, B.; Muñoz-Nieto, N.L.; Arias-Sanchez, P.; Gonzalez-Aguilera, D. Mobile LiDAR system: New possibilities for the documentation and dissemination of large cultural heritage sites. *Remote Sens.* **2017**, *9*, 189.
55. Remondino, F.; El-Hakim, S.; Girardi, S.; Rizzi, A.; Benedetti, S.; Gonzo, L. 3D Virtual reconstruction and visualization of complex architectures-The 3D-ARCH project. In Proceedings of the 3rd International Workshop, 3D-ARCH 2009, Trento, Italy, 25–28 February 2009.
56. Xu, Z.; Wu, L.; Shen, Y.; Li, F.; Wang, Q.; Wang, R. Tridimensional Reconstruction Applied to Cultural Heritage with the Use of Camera-Equipped UAV and Terrestrial Laser Scanner. *Remote Sens.* **2014**, *6*, 10413–10434.
57. Hess, M.; Petrovic, V.; Meyer, D.; Rissolo, D.; Kuester, F. Fusion of multimodal three-dimensional data for comprehensive digital documentation of cultural heritage sites. *Digit. Herit.* **2015**, *2*, 595–602.
58. Figueiredo, M.; Bernardes, J.P.; Rodrigues, J.I.; Gonçalves, C. A Framework Supported by Modeling and Virtual/Augmented Reality for the Preservation and Dynamization of Archeological-Historical Sites. In *Handbook of Research on Technological Developments for Cultural Heritage and eTourism Applications*; IGI Global: Hershey, PA, USA, 2018; pp. 215–227.
59. Fernández-Palacios, B.J.; Morabito, D.; Remondino, F. Access to complex reality-based 3D models using virtual reality solutions. *J. Cult. Herit.* **2017**, *23*, 40–48.
60. Stanco, F.; Gallo, G. (Eds.) *Proceedings of ArcheoFOSS: Free, Libre and Open Source Software e Open Format Nei Processi Di Ricerca Archeologica: VIII Edizione, Catania 2013*; Archaeopress: Oxford, UK, 2016.
61. Parish, Y.I.H.; Müller, P. Procedural modeling of cities. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; ACM: New York, NY, USA, 2001; pp. 301–308.
62. Lindenmayer, A. Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs. *J. Theor. Biol.* **1968**, *18*, 300–315.
63. Wonka, P.; Wimmer, M.; Sillion, F.; Ribarsky, W. Instant architecture. *ACM Trans. Graph.* **2003**, *22*, 669–677.
64. Finkenzeller, D.; Bender, J.; Schmitt, A. Feature-based decomposition of façades. In Proceedings of the Virtual Concept, Biarritz, France, 8–10 November 2005.
65. Greuter, S.; Parker, J.; Stewart, N.; Leach, G. Real-time Procedural Generation of 'Pseudo Infinite' Cities. In Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, Melbourne, VIC, Australia, 11–14 February 2003; ACM: New York, NY, USA, 2003.
66. Dikaiakou, M.; Efthymiou, A.; Chrysanthou, Y. Modelling the Walled City of Nicosia. In Proceedings of the 4th International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage, Brighton, UK, 5–7 November 2003; Eurographics Association: Aire-la-Ville, Switzerland, 2003; pp. 61–70.
67. Müller, P.; Wonka, P.; Haegler, S.; Ulmer, A.; Van Gool, L. Procedural modeling of buildings. *ACM Trans. Graph.* **2006**, *25*, 614–623.
68. Muller, P.; Vereenooghe, T.; Wonka, P.; Paap, I.; Van Gool, L. Procedural 3D reconstruction of Puuc buildings in Xkipché;. In Proceedings of the 7th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage, Nicosia, Cyprus, 30 October–4 November 2006; Eurographics Association: Aire-la-Ville, Switzerland, 2006; pp. 139–146.
69. Besuievsky, G.; Patow, G. Procedural modeling historical buildings for serious games. *Virtual Archaeol. Rev.* **2013**, *4*, 160–166.
70. Tepavčević, B.; Stojaković, V. Procedural modeling in architecture based on statistical and fuzzy inference. *Autom. Constr.* **2013**, *35*, 329–337.
71. Lipp, M.; Wonka, P.; Wimmer, M. Interactive Visual Editing of Grammars for Procedural Architecture. *ACM Trans. Graph.* **2008**, *27*, 102:1–102:10.
72. Carrozzino, M.; Tecchia, F.; Bergamasco, M. Urban procedural modeling for real-time rendering. In Proceedings of the 3rd ISPRS International Workshop 3D-ARCH, Trento, Italy, 25–28 February 2009.
73. Liu, Y.; Xu, C.; Zhang, Q.; Pan, Y. The Smart Architect: Scalable Ontology-Based Modeling of Ancient Chinese Architectures. *Intell. Syst. IEEE* **2008**, *23*, 49–56.

74. Li, S.L.; Li, L.; Cao, M.W.; Cao, L.; Jia, W.; Liu, X.P. Rapid Modeling of Chinese Huizhou Traditional Vernacular Houses. *IEEE Access* **2017**, *5*, 20668–20683.
75. Silva, P.B.; Coelho, A. Procedural Modeling for Realistic Virtual Worlds Development. *J. Virtual Worlds Res.* **2011**, *4*, doi:10.4101/jvwr.v4i1.2109.
76. Kelly, T.; Wonka, P. Interactive Architectural Modeling with Procedural Extrusions. *ACM Trans. Graph.* **2011**, *30*, 14:1–14:15.
77. Vanegas, C.A.; Kelly, T.; Weber, B.; Halatsch, J.; Aliaga, D.G.; Müller, P. Procedural generation of parcels in urban modeling. *Comput. Graph. Forum* **2012**, *31*, 681–690.
78. Barroso, S.; Besuievsky, G.; Patow, G. Visual copy & paste for procedurally modeled buildings by ruleset rewriting. *Comput. Graph.* **2013**, *37*, 238–246.
79. Steinberger, M.; Kenzel, M.; Kainz, B.; Müller, J.; Peter, W.; Schmalstieg, D. Parallel generation of architecture on the GPU. *Comput. Graph. Forum* **2014**, *33*, 73–82.
80. Steinberger, M.; Kenzel, M.; Kainz, B.; Wonka, P.; Schmalstieg, D. On-the-fly generation and rendering of infinite cities on the GPU. *Comput. Graph. Forum* **2014**, *33*, 105–114.
81. Ilcík, M.; Musialski, P.; Auzinger, T.; Wimmer, M. Layer-Based Procedural Design of Façades. *Comput. Graph. Forum* **2015**, *34*, 205–216.
82. Edelsbrunner, J.; Havemann, S.; Sourin, A.; Fellner, D.W. Procedural Modeling of Round Building Geometry. In Proceedings of the 2016 International Conference on Cyberworlds (CW), Chongqing, China, 28–30 September 2016; pp. 81–88.
83. Koenig, R.; Miao, Y.; Knecht, K.; Buš, P.; Mei-Chih, C. Interactive Urban Synthesis. In Proceedings of the Computer-Aided Architectural Design: Future Trajectories: 17th International Conference, CAAD Futures 2017, Istanbul, Turkey, 10–14 July 2017; Volume 724, p. 23.
84. Rogla, O.; Pelechano, N.; Patow, G. Procedural Semantic Cities. 2017. Available online: https://pdfs.semanticscholar.org/730e/0f0f364777d3c35f22f3571ab52e0a137694.pdf (accessed on 14 March 2019).
85. Jesus, D.; Patow, G.; Coelho, A.; Sousa, A.A. Generalized selections for direct control in procedural buildings. *Comput. Graph.* **2018**, *72*, 106–121.
86. Stiny, G.; Mitchell, W.J. The Palladian Grammar. *Environ. Plan. B Plan. Des.* **1978**, *5*, 5–18.
87. Rau-Chaplin, A.; MacKay-Lyons, B.; Spierenburg, P. The LaHave House Project: Towards and Automated Architectural Design Service. In Proceedings of the International Conference on Computer Aided Design (CADEX-96), Palo Alto, CA, USA, 6–8 November 1996; pp. 62–66.
88. Bradley, B. Towards the Procedural Generation of Urban Building Interiors. Master's Thesis, University of Hull, Hull, UK, 2005.
89. Martin, J. Procedural house generation: A method for dynamically generating floor-plans. In Proceedings of the Symposium on Interactive 3D Graphics and Games, Redwood City, CA, USA, 14–17 March 2006.
90. Hahn, E.; Bose, P.; Whitehead, A. Lazy Generation of Building Interiors in Realtime. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, Ottawa, ON, Canada, 7–10 May 2006; pp. 2441–2444.
91. Dahl, A.; Rinde, L. Procedural Generation of Indoor Environments. Master's Thesis, Charmers University of Technology, Gothenburg, Sweden, 2008.
92. Rodrigues, N.; Dionísio, M.; Gonçalves, A.; Magalhães, L.G.; Moura, J.P.; Chalmers, A. Rule-based Generation of Houses. *Comput. Graph. Geom.* **2008**, *10*, 49–65.
93. RGEU. Regulamento Geral das Edificações Urbanas, Decreto no. 38382. 2008. Available online: http://www.pgdlisboa.pt/leis/lei_mostra_articulado.php?nid=1217&tabela=leis (accessed on 14 March 2019)
94. Maciel, M. *Vitrúvio—Tratado De Arquitetura*; Ist Press: Palo Alto, CA, USA, 2006.
95. Rodrigues, N.; Magalhães, L.; Moura, J.; Chalmers, A. Reconstruction and generation of virtual heritage sites. *Digit. Appl. Archaeol. Cult. Herit.* **2014**, *1*, 92–102.
96. Gonçalves, A.; Rodrigues, N.; Hipólito-Correia, V. Conimbriga as paradigm of cultural heritage (virtual) recriation. In Proceedings of the 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal, 21–24 June 2017; pp. 1–6.
97. Lopes, R.; Tim, T.; Smelik, R.M.; Jan de Kraker, K.; Bidarra, R. A Constrained Growth Method for Procedural Floor Plan Generation. In Proceedings of the GAMEON'10, Leicester, UK, 17–19 November 2010.
98. Bruls, M.; Huizing, K.; van Wijk, J. Squarified Treemaps. In Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, Vienna, Austria, 26–28 May 1999; pp. 33–42.

99. Marson, F.; Musse, S.R. Automatic real-time generation of floor-plans based on squarified treemaps algorithm. *Int. J. Comput. Games Technol.* **2010**, *2010*, 7:1–7:10.
100. Mirahmadi, M.; Shami, A. A Novel Algorithm for Real-time Procedural Generation of Building Floor Plans. *arXiv* **2012**, arXiv:1211.5842.
101. Merrell, P.; Schkufza, E.; Koltun, V. Computer-generated residential building layouts. *ACM Trans. Graph.* **2010**, *29*, 181:1–181:12.
102. Trescak, T.; Esteva, M.; Rodriguez, I. A Virtual World Grammar for automatic generation of virtual worlds. *Vis. Comput.* **2010**, *26*, 521–531.
103. Hohmann, B.; Havemann, S.; Krispel, U.; Fellner, D. A GML shape grammar for semantically enriched 3D building models. *Comput. Graph.* **2010**, *34*, 322–334.
104. Tutenel, T.; Smelik, R.; Lopes, R.; de Kraker, K.; Bidarra, R. Generating Consistent Buildings: A Semantic Approach for Integrating Procedural Techniques. *Comput. Intell. AI Games, IEEE Trans.* **2011**, *3*, 274–288.
105. Leblanc, L.; Houle, J.; Poulin, P. Component-based Modeling of Complete Buildings. In Proceedings of the Graphics Interface, Waterloo, ON, Canada, 25–27 May 2011; pp. 87–94.
106. Santos, D.S.; Dionísio, M.; Rodrigues, N.; Pereira, A. Efficient Creation of 3D Models from Building's Floor Plans. *Int. J. Interact. Worlds* **2011**, *2011*, 1–30.
107. Arvola Bjelkesten, K. Feasibility of Point Grid Room First Structure Generation: A bottom-up approach. Master's Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 2017.
108. Wang, X.Y.; Yang, Y.; Zhang, K. Customization and generation of floor-plans based on graph transformations. *Autom. Constr.* **2018**, *94*, 405–416.
109. Gröger, G.; Kolbe, T.H.; Nagel, C.; Häfele, K.H. *OGC City Geography Markup Language(CityGML) Encoding Standard*; Technical Report; Open Geospatial Consortium; 2012. Available online: https://mediatum.ub.tum.de/doc/1145731/file.pdf (accessed on 14 March 2019).
110. Larive, M.; Gaildrat, V. Wall Grammar for Building Generation. In Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, Kuala Lumpur, Malaysia, 29 November–2 December 2006; ACM: New York, NY, USA, 2006; pp. 429–437.
111. Rodrigues, N.; Magalhães, L.G.; Moura, J.P.; Chalmers, A. Automatic reconstruction of virtual heritage sites. In Proceedings of the 9th International Conference on Virtual Reality, Archaeology and Cultural Heritage, Braga, Portugal, 2–5 December 2008; pp. 39–46.
112. Johnson, B.; Shneiderman, B. Treemaps: A space-filling approach to the visualization of hierarchical information structures. In Proceedings of the Visualization, San Diego, CA, USA, 22–25 October 1991; pp. 284–291.
113. Correia, V. Conimbriga Ruínas—Museu Monográfico. 2011. Available online: http://www.conimbriga.pt/portugues/ruinas0.html (accessed on 14 March 2019).
114. Bontcheva, K.; Cunningham, H.; Tablan, V.; Maynard, D.; Hamza, O. Using GATE as an Environment for Teaching NLP. In Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics-Volume 1. Association for Computational Linguistics, Philadelphia, PA, USA, 7 June 2002; pp. 54–62.
115. Baptista, R.; Reis, L.P.; Magalhães, L.; Rodrigues, R.; Coelho, A.; Sousa, A. Extracção de informação de texto formal para a modelação expedita de edifícios monumentais. In Proceedings of the Actas do 20*o* Encontro Português de Computação Gráfica, Viana do Castelo, Portugal, 24–26 October 2012.
116. Adão, T.; Pádua, L.; Narciso, D.; Sousa, J.J.; Peres, E.; Agrellos, L.; Magalhães, L. MixAR—A multi-tracking mixed reality system to visualize virtual ancient buildings aligned upon ruins. *J. Inf. Technol. Res. (JITR)* **2019**, *12*, 4–32.
117. Martin, J. Algorithmic Beauty of Buildings Methods for Procedural Building Generation. Computer Science Honors Theses. 2005. Available online: https://digitalcommons.trinity.edu/compsci_honors/4 (accessed on 10 May 2019).
118. Rodrigues, N. Rule-Based Generation of Virtual Traversable Architectural-Period Houses. Ph.D. Thesis, University of Trás-os-Montes e Alto Douro, Vila Real, Portugal, 2010.