Digital Applications in Archaeology and Cultural Heritage 1 (2014) 92–102

Contents lists available at ScienceDirect

Digital Applications in Archaeology and Cultural Heritage

journal homepage: www.elsevier.com/locate/daach

Reconstruction and generation of virtual heritage sites

N. Rodrigues [a,*], L. Magalhães [b,c], J. Moura [b,c], A. Chalmers [d]

[a] School of Technology and Management, Computer Science and Communication Research Centre, Polytechnic Institute of Leiria, Leiria, Portugal
[b] Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal
[c] Institute for Systems and Computer Engineering of Porto (INESC Porto), Porto, Portugal
[d] The Digital Laboratory, WMG, University of Warwick, UK

ARTICLE INFO

Article history:
Received 4 April 2013
Received in revised form
26 February 2014
Accepted 27 June 2014
Available online 5 July 2014

ABSTRACT

Traditionally procedural modelling techniques are commonly used to generate new structures and are presently established in several areas such as video games and computer animated movies. However they may also be used in heritage applications to efficiently produce models of non-existing worlds for which there is some kind of knowledge (e.g. floor plans, photographs) to support the reconstruction of realistic environments. Similarly they may also be used to support the generation of distinct possibilities that allow experts to draw some conclusions or conceive different hypotheses about lost worlds. The present paper shows the benefits and constraints that may arise from the use of such techniques in virtual heritage applications.

Furthermore, a whole method is proposed, for the reconstruction and generation of virtual heritage traversable house models, provided through the means of a grammar, demonstrated with the reconstruction and generation of several Roman houses from the heritage site of Conimbriga, Portugal.

1. Introduction

With the increasing demand for more complex and larger models in different fields, such as the design of virtual worlds, video games and computer animated movies the need to generate them automatically has become more necessary than ever. Manual tools are no longer sufficient to match this rising need and the impact that automatic tools may have within these fields is essential. Indeed, it is possible to eliminate most of the effort, associated with the creation of such environments, by providing tools that may generate "massive" 3D content automatically.

Still, there are also several other areas, such as archaeology, that may benefit from the use of procedural techniques and there are some related work addressing procedural heritage.

It is not the same to generate purely fictional structures and recreate existing structures. In order to assure the realism of the final models there is the need to account for many architectural concerns. Therefore, there are several aspects that have to be taken into consideration when dealing with the reconstruction and generation of virtual heritage worlds.

In this paper a method is presented which allows the specification of several features of a house, provided through textual specifications (using a grammar) allowing both the reconstruction and the generation of traversable houses. This feature is particularly useful, because it allows a user with no knowledge or experience with modelling tools, to construct his/her own models through a set of textual specifications.

Moreover the method also allows the use of incomplete specifications which allows the generation of several resembling models within an architectural style. The presented method is demonstrated with the generation of several Conimbriga's Roman houses (Conimbriga is a heritage roman site in Portugal) showing some dissimilarity amongst them.

The subsequent sections of this chapter start by presenting an overview of procedural modelling, and some projects concerning virtual heritage, as well as discussing several aspects concerning the use of procedural techniques in heritage applications.

Then, in the subsequent chapters, the several modules of the proposed method for the reconstruction and generation of heritage houses are described. After that, some results are presented and the major conclusions of this work enounced as well as some future work.

1.1. Related work

The introduction of this paper describes some features of procedural modelling techniques, showing their undeniable advantages in the generation of new worlds. Indeed, procedural modelling is the tool of choice, when addressing generation, but it may also prove to be useful to create virtual models representing

* Corresponding author.

http://dx.doi.org/10.1016/j.daach.2014.06.003
2212-0548/

structures lost in time, or even to create virtual models of existing structures (reconstruction). For these reasons it is important to first establish the significance of two different terms used throughout this paper: reconstruction and generation. The term "reconstruction" refers to the creation of models using automatic techniques representing existing structures or creation of models representing structures no longer existing but for each there is enough evidences (i.e. ruins, floor plans, photographs, etc.) to faithfully reconstruct them. The term "generation" refers to the generation of new worlds, i.e. automatic creation of fictional structures not intended to represent any ever-existing structure.

When considering urban procedural modelling, techniques have been applied in very different applications with different techniques and covering different aspects. They may centre the attention on the generation of individual houses (Martin, 2005; Rodrigues et al., 2008a, 2008b, 2009), on tall buildings and skyscrapers (Greuter et al., 2003; Parish and Müller, 2001; Park, 2005), or on large urban environments (Willmott et al., 2001; Lechner et al., 2003; Bostrom et al., 2004, applying urban planning principles to the design of virtual environments (Ingram et al., 1996), modelling from architectural rules (Fuchs, 2006; Luca et al., 2007; Rodrigues et al., 2008a, 2008b, 2008c), dealing with specific features as the generation of streets and roads (Parish and Müller, 2001; Chen et al., 2008), or on house specific house features such as windows (Charbonneau et al., 2006).

Likewise, they may focus on modern architecture (Bessa et al., 2005; Coelho et al., 2007; Martin, 2005; Rodrigues et al., 2008a, 2008b) or on heritage architecture, such as Roman (Müller et al., 2005; Rodrigues et al., 2007, 2008c) or Mayan architecture (Müller et al., 2006a). Indeed, procedural techniques have been used to create heritage models representing ancient structures. For example, in Müller et al. (2005), the authors wrote a set of rules to generate a virtual model of Pompei using CityEngine, a system introduced by Parish and Müller (2001). Later, using the same system, the authors of "Procedural 3D Reconstruction of Puuc Buildings in Xkipche" (Müller et al., 2006a) generated Puuc-style buildings, similar to the ones that may be found in the ancient Mayan site of Xkipché, in Mexico. This system was also used with other examples shown in Müller et al. (2006b).

In structures showing more complex geometries, the aforementioned techniques (i.e. the definition of sets of rules to describe the physical geometry of physical structures) may prove to be, in several cases, harder to achieve (or take more time) than when using traditional methods (manual modelling) or even be impossible to realise. Indeed, an experienced modeller may produce 3D models very efficiently when given adequate data (e.g. measurements, photographs, etc.) about a physical structure. Unfortunately, manual modelling does not allow the efficient reproduction of large virtual environments showing diversity (e.g. creation of different houses corresponding to a similar architectural style).

Other projects, such as "Rome Reborn" (Frischer et al., 2008), use both reconstruction, which currently is done manually, and generation. One of the goals of this project was "to create a 3D digital representation illustrating the development of the city of Rome in antiquity from the first settlement in the late bronze age to the depopulation of the ancient city in the 6th century AD". Hence some of the buildings present in the digital model are manually modelled, based on archaeological evidence (such as excavations, studies and ancient literary sources). Other buildings are procedurally modelled from a digitisation of the Plastico di Roma Antica, "a large plaster-of-Paris model of imperial Rome (16×17 m^2) created in the last century" (Guidi et al., 2005a, 2005b, 2008; Kimberly et al., 2009), by replacing the scan data from the digitisation with geometrically simplified forms, and the modelling of these simplified form faces with detailed architectural features (e.g. doors, windows, balconies). Another interesting feature about "Rome Reborn" is that the aforementioned CityEngine, the urban modelling software from Procedural Inc., was used in the procedural modelled buildings (Kimberly et al., 2009).

1.2. Procedural heritage

When addressing reconstruction or even considering a procedural technique for reconstruction it is important to think in which areas it may be applied. So, one may think at most obvious areas such as archaeology, architecture, etc. This raises some questions concerning the ability to represent or reconstruct structures using procedural methods since, for example, in areas such as archaeology one main key point to recreate the past is the realism, which means that it is crucial to create features with a higher level of detail. Therefore, in some situations, it is advisable to use manually modelled objects rather than to try to use some algorithm which would resemble the real structures that are pretended to be represented. Producing these realistic models may be achieved either by designing them from known similar objects (e.g. artefacts, columns, windows, etc.) or even by replicating existing ones. Independently of how the models are created, the fact is that often there is evidence which indicate that most probably, within a certain architectural style, similar structures have identical features.

Considering the aforementioned constraints it is important to reason how procedural modelling techniques may be useful when dealing with areas such as archaeology. One way is to efficiently produce models of non-existing worlds for which there is some kind of knowledge (e.g. floor plans, photographs) to support the reconstruction of realistic environments. Other is to support the generation of distinct possibilities to allow experts to draw some conclusions or conceive different hypotheses about lost worlds. For example in several Roman heritage sites (e.g. Conimbriga, Portugal) some parts of the city are yet uncovered or may even be lost. By taking as an example similar Roman cities it is possible to determine what may exist in most of the site and, this way, produce new models resembling existing structures.

The idea of generating structures similar to existing ones is not something new and not even confined to heritage. In fact, it is something that is pursued by several authors and may be referred as inverse procedural modelling. For example, in Aliaga et al. (2007), the authors describe an interactive system for the creation of new buildings in the style of others or for the modification of existing buildings. The idea is to create geometric models from photographs, divide the building into their basic external features (e.g. floors, windows, etc.) and create a grammar that captures the repetition patterns of these features. Then it allows the user to design building configurations from building blocks and finally to divide these building configurations into the several features, using the repetition patterns from the grammar. One of the most interesting aspects of this approach is the idea itself and its potential use in architecture. Nevertheless, there is still a great deal of interaction needed from the user, starting from the initial photograph mapping until the design of the final building blocks (the interaction in this last step does make a lot of sense if the idea is for the user to control new geometries). Another aspect to notice about this approach is that only facades are created.

The next chapter describes a method which uses procedural modelling techniques and yet maintains the realism of the virtual models, by using a hybrid approach.

2. Method architecture

Within this chapter the main features of the method for the reconstruction and generation of architectural-period traversable

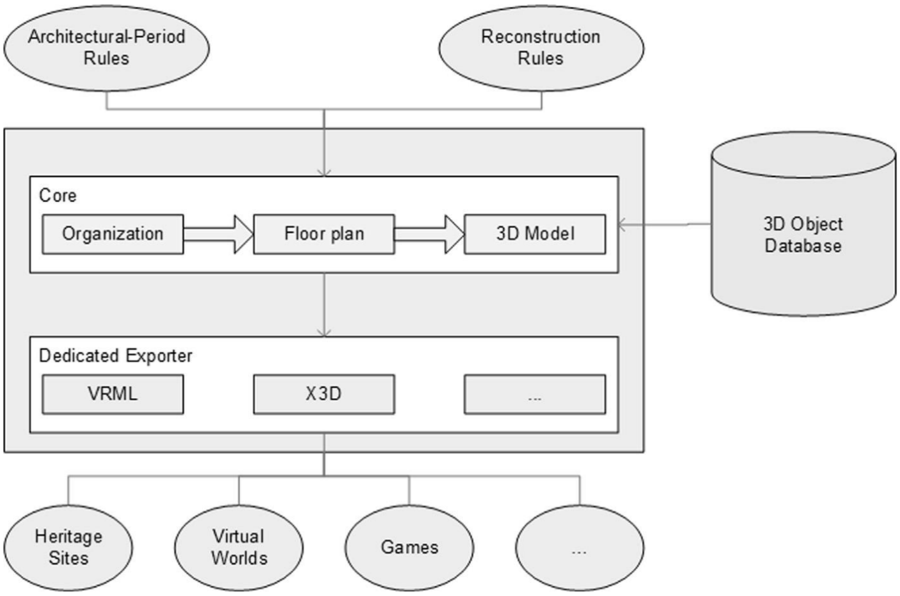

Fig. 1. Method architecture.

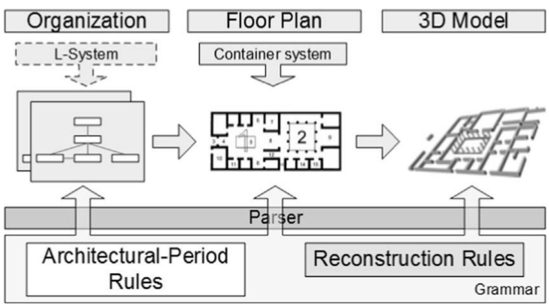

Fig. 2. Method core.

houses are described, showing its application with the case study of Roman heritage houses, in particular Conimbriga's houses.

The method is guided by "Architectural-Period Rules", when the goal is the generation of new houses, or "Reconstruction Rules", when the goal is the reconstruction of existing houses. The "Core" is where most of the processing is done including the parsing of the rules and geometry creation. Alongside the "Core" is a dedicated exporter module, able of producing different types of formats suitable to different areas. A "3D Object Database" is also present to allow the inclusion of existing geometry which may contribute to give to the final models an enhanced realism. This way, Fig. 1 symbolizes the generalisation of the present method, for the reconstruction/generation of Architectural-Period houses, with all the major modules which are described in the next sections.

2.1. Core

The major part of the processing is done inside the core, including the processing of the input rules (either architectural-period or reconstruction), which are encoded in a formal grammar. For this reason and since the rules have to be parsed and are used in the several steps of the core, they are represented again in Fig. 2 and therefore the resultant grammar is the first component to be described in the next section.

The whole method is then guided by a grammar to automatically create the structures. Hence, the term "automatically" has a two-folded implication: on one side it means that the structures may be generated relying on an L-system[1] (and using architectural-period rules); on the other side the list of structures may be specified by the user (reconstruction rules). Thereby, these two features couple with the two different purposes of the method. The L-system is used for the generation of new structures, but if the purpose is the reconstruction of heritage structures that no longer exist, the user may define suitable rules obtained through archaeological data to reconstruct these structures.

In the first step of the method (Organisation), a list of rooms is generated (or other physical structures) by an L-system, or reconstructed through user rules, which are then grouped in a multi-layer connection graph, defining the connections between them.

In the second step (Floor Plan), the information from the Organisation step is used to create the bi-dimensional shapes (representing structures) and distributed in the physical space leading to a floor plan. A container system was also developed to aid in the placement of shapes.

Finally, in the third step (3D model), the floor plan created in the second step is the foundation for the three-dimensional geometry creation.

2.2. Grammar

Most authors presenting approaches within the field of urban procedural modelling, commonly use grammars (e.g. Coelho et al. (2007), Müller et al. (2006b), Parish and Müller (2001), Wonka et al. (2003)), although it is a fact that grammar approaches require many rules to generate the models and are only able to model a small class of objects, as stated by Merrell (2007). If at first this seems a clear disadvantage, compared to other approaches (e.g. image-based), the flexibility provided by grammars makes them a powerful tool for the generation of urban models, showing in the present research more advantages than disadvantages. Furthermore, it is also possible to avoid the text-based complexity of grammars by designing efficient interactive visual editing systems for shape grammars, such as the one presented by Lipp et al. (2008).

[1] L-systems or Lindenmayer systems (named after their creator) consist of formal grammars and were introduced in 1968 by the Hungarian theoretical biologist and botanist from the University of Utrecht, Aristid Lindenmayer (Lindenmayer, 1968).

For these reasons, a generation grammar seems a conceivable solution to achieve the desired aspirations and a simple context-free grammar was defined, using the classic formalisation of generative grammars, as proposed by Chomsky (1956) with production rules written using the Extended Backus-Naur Form (EBNF) notation (ISO/IEC14977, 1996). This way, a top-down parser – recursive descent parser – was also implemented, which is made of several components as the lexical analyser (or lexer) and the syntactic analyser (see Fig. 3). The lexer checks input strings (lexical analysis) formed by regular expressions (Type 3) and converts sequences of characters to tokens (tokenization). The parser checks for the correct syntax (syntactic analysis) to determine the grammatical structure according to the formal grammar.

Fig. 3 presents in more detail the several steps involved in this parsing mechanism. Thus, the parser takes as input a source string with all the rules and then in the first stage – lexical analysis – this source string is split into meaningful symbols (tokens) defined by a grammar of regular expressions. Then, in the next stage – syntactic analysis – the tokens are checked to see if they form an allowable expression in reference to the defined grammar. Type validity is also performed in this stage. The output of this stage is a tree with all the validated tokens. The final stage – interpreter – simply consists of traversing the tree and taking the appropriate actions for each validated expression.

At its present state, the grammar already has implemented several *commands*, allowing different types of interactions over the generation. These *commands* are grouped into different categories, such as:

Definition of shapes, e.g.
PartByName (*Cubiculum*; *Cubiculum*1) – defines a shape names *Cubiculum*1, belonging to the *Cubiculum* group.
Definition of shape attributes, e.g.
SetProp (*Atrium*; *TypeDim*) – defines a new property, for the shape *Atrium*, named *TypeDim* (type of dimensioning).
Connection Rules, e.g.
AllowedConnection (*Vestibulum*; *Fauces*) – allows the spatial placement of the *Vestibulum* and *Fauces* together.
Placement rules, e.g.
PlacePartByName (*Impluvium*; *Atrium*; *centre*) – places the shape *Impluvium* in the *Atrium* centre.

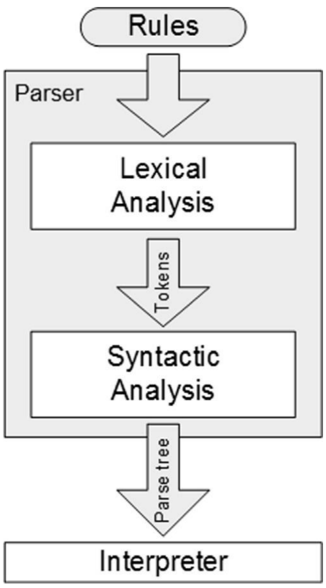

Fig. 3. Parser.

The above examples present just a fraction of the implemented *commands* for each category, since there are many more and the grammar is still being developed to allow new features. Furthermore, there are also several other features of the grammar (including simple arithmetic operations, conditional rules such as "*If*" rules and nested rules), which effectively give an adequate level of control over the whole process of geometric structures creation.

3. Organisation

The grammar rules, aforementioned in the previous section, are used to guide the whole generation process. Thus, in the first step of the method is generated a list of rooms, either by an L-system or through user rules, which is then grouped in a multi-layer connection graph defining the connections between them. This information is used to define the organisation of a house which will serve as the source for the floor plan generation.

3.1. L-system

One main distinction between the L-system used in the current method and the previous approaches (e.g. Parish and Müller (2001)), is that this method is used to define the interior parts of a house instead of defining operations between shapes or defining pathways and populating them with different objects (e.g. buildings, trees, persons). The current use of the L-system may be exemplified with the practical example of Roman houses, where to generate a list of rooms, an L-system may be used in which an alphabet is formed by the symbols corresponding to the different parts of the house. Assuming that the axiom (ω), corresponds to the symbol *RomanStructure* from the *V* alphabet and the productions ($p0$, $p1$,…, pn) correspond to the possible replacements between the symbols, then an example of a Roman house may be represented as:

V={*RomanStructure, RomanHouse*, *AtriumArea*, *Atrium*, *Ala*, *Cubiculum*,…, *PeristyliumArea*, *Peristylium*,…}
ω=*RomanStructure*
*p*0: *RomanStructure*→*RomanHouse*
*p*1: *RomanHouse*→*AtriumArea*, *PeristyliumArea*
*p*2: *AtriumArea*→*Atrium*, *Ala*, *Cubiculum*,…
*p*3: *PeristyliumArea*→*Peristylium*, *Cucina*, *Bathroom*,…

Terminal symbols correspond to Roman rooms and non-terminal symbols may either be a simple abstraction to organise the several rooms or represent containers serving the purpose of placing the rooms during the floor plan step (described in chapter 4).

The rewriting system may then be understood as a decomposition system, where every symbol is iteratively replaced until only terminal symbols, which correspond to different parts of the

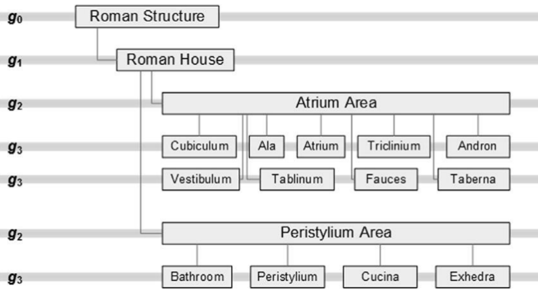

Fig. 4. L-system generations.

house, are reached. This is demonstrated in Fig. 4 for the presented L-system.

In the first generation (*g*1), the symbol *RomanStructure* is replaced by the symbol *RomanHouse*. Then, in the second generation (*g*2), the *RomanHouse* is subdivided in its major areas: the area around the *atrium* and the area around the *peristylum*. Finally, in the third generation (*g*3), the two major areas are subdivided into the corresponding parts of the house.

3.2. Multi-layer connection graph

In this step, the list of rooms produced by the L-system is added to a graph representing the connections between the rooms. This graph serves the purpose of leading to the next step, where each room of the house has to be physically placed in the floor plan next to any other room where it links to. Moreover, it also serves the purpose of determining where passages, doors or other types of objects have to be placed.

One thing to notice is that the connection graph algorithm also allows the use of containers, making this a powerful tool. This means that, when containers are used, there are really two types of entities in the graphs: containers and rooms. Rooms represent terminal entities (i.e. they may not be decomposed into other parts of a house, although they may have other features placed inside them) but a container may have other containers on it, which means that the connection graph algorithm may lead to a generation of several graphs. This is represented in Fig. 5 where the container entities represented in *Graph* 0 lead to the creation of other graphs (*Graph* 0.0 *and Graph* 0.1) and then again for each container entity.

When containers are present, the algorithm starts at the higher level of containers and creates a graph which represents the connections between them. To do this, the algorithm iteratively removes from the list of rooms generated by the L-system (or by the user rules), each one of the higher level containers and adds them to the graph until there is no more containers to add.

Finally, when a graph is completed, every node of the graph is traversed and for each node corresponding to a container a new graph is created and the whole process is repeated. This is done until there are only room entities in the lower level graphs.

4. Floor plan

In this chapter, a floor plan approach is presented, motivated by the analysis of Roman house Architecture. The results from the "organisation" step are then used to guide the generation of the floor plan which introduces a container system, and several other sub-steps, such as the generation of permutations, the dimensioning and the placement of rooms. The general idea is to dimension and physically place each one of the rooms, generated inside the matching containers determined in the "organisation" step.

The first step in the generation of the floor plan is to generate all permutations, within each container, to determine all possible placement order combinations for the rooms inside them. Then, if the idea is to match some user criteria, the most favourable permutation (the one that matches more user choices) is chosen. Otherwise, if the idea is to present alternative distributions, this may be used to automatically present dissimilar scenarios. Subsequently, all the rooms (as well as other features, e.g. ornaments) are dimensioned to their proper dimensions and placed in the bi-dimensional physical space according to the multi-layer connection graph, resulting in a floor plan. Most of these aspects are further described within the next sections, starting by the container system, essential for the correct comprehension of all the steps involved in the generation of the floor plan.

4.1. The container system

Considering that in the floor plan every feature of a house is represented by a bi-dimensional shape, the shapes may be classified into: terminal shapes, which represent final geometric features (e.g. *cubiculum*), or non-terminal shapes (e.g. container). Within this context, the "terminal" term means that there are no other rooms of a house contained in a shape. Though, a terminal shape may contain some other ornaments or even containers that may be used to place some features (e.g. textures). The container is therefore a shape, which may or may not have a visible representation, and serves the purpose of organising (i.e. placing, sizing, aligning) all the other shapes which are placed inside it. A container may also contain other container objects.

This container approach is somewhat similar to form containers, which are available for programmers in software development tools to aid the design of graphical interfaces. However, it has some more characteristics, including the possibility to align a shape centred in a container, distribute shapes over a container and the possibility to automatically size containers depending on the size of the clients (shapes) placed on it.

It is also possible to distribute objects over shapes. This feature may be exemplified with Roman Architecture, where columns have often to be placed along certain areas. Such is the example of placing columns along a *peristylum*, by distributing them among the edges.

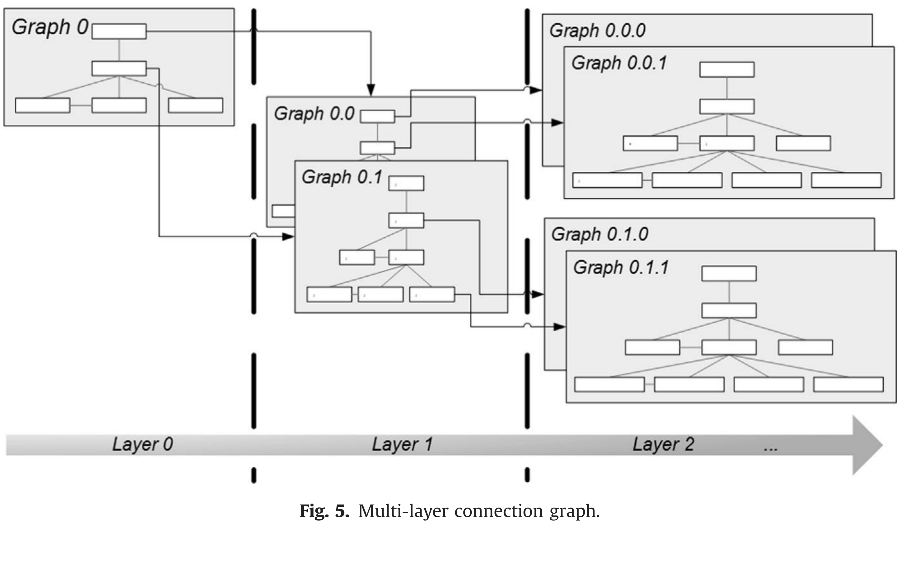

Fig. 5. Multi-layer connection graph.

It is this combination of different possibilities of dealing with shapes that makes this container system a powerful tool, which may be used to simplify the placement of the different parts of a house. Although the greater advantages of the system are mostly achieved when the houses have a highly aligned distribution of rooms, as in a classic Roman heritage house (see Fig. 6), in houses with more rough distributions it may also be a helpful tool.

The use of containers is demonstrated by applying several containers to align the different features of a Roman house with a classic disposition represented in Fig. 6.

The proper container hierarchy for the geometry represented in Fig. 6 may then be represented with the tree presented in Fig. 7.

For the example of Fig. 6 were used nine containers. The first container (at layer 0) correspond to the whole outer geometry of the house and encloses two containers (at layer 1), to allow the alignment of the *atrium* area (on the left side of Fig. 6) and the *peristylium* area (on the right side of Fig. 6). Finally, these two containers enclose other containers (at layer 2) which enclose the several rooms of the house.

The container system and the multi-layer connection graph are interrelated, as the multi-layer connection graph describes how the different parts of the house should be connected and the container system is used to make the spatial distribution of the different parts of the house. Actually, the multi-layer connection system defines how the containers or any other shapes are organised, i.e. nested and connected. The container system uses the information from the multi-layer connection graph to create the shapes. Therefore, it tries to distribute the different parts of the house inside the containers, placing together parts that have connections in the multi-layer connection graph.

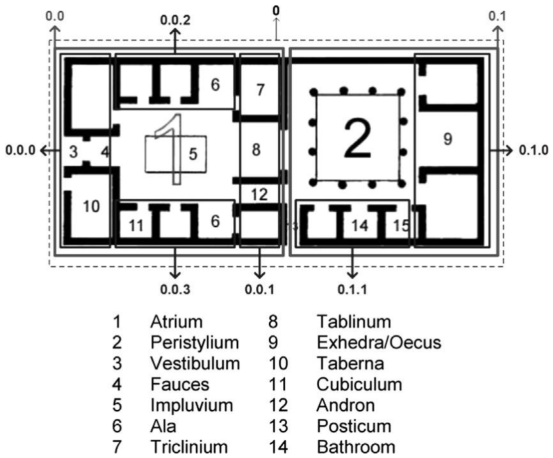

Fig. 6. Use of containers in a classical Roman house. Adapted from Empire (2007).

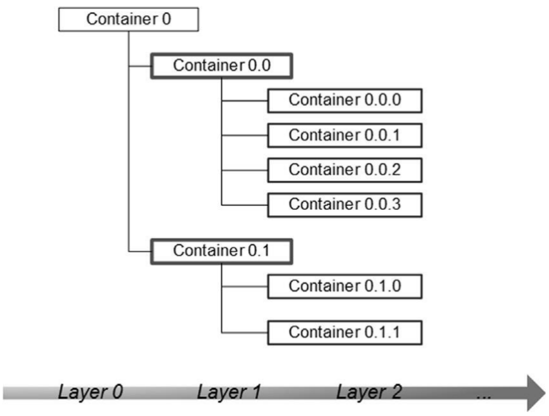

Fig. 7. Container hierarchy

Finally, the use of containers is also extended to the final 3D geometry since they may be used in objects' faces. This is particular useful to distribute textures over a face representing different characteristics of a Roman house (e.g. floor, walls, ceilings). Since containers may contain other containers, this appears to be a powerful mechanism to apply detailed textures producing impressive results. The use of containers to apply textures is shown in Section 4.2.1.

4.2. Dimensioning

Before the actual placement of the different parts of the house occurs, it is imperative to set them to a proper size. For this reason, the floor plan generation uses the information from the grammar rules to define the dimensions for each part of the house. When the user does not specify room dimensions, it is possible to automatically set them, either by using some default values or following architectural rules. Indeed, it is possible to write rules to act as a trigger for other rules. However, if the user decides to define the dimensions these will override any type of pre-defined rules. To exemplify this, once again Roman Architecture is used throughout the rest of this section.

For example, one starting point to generate a Roman house may be defining the dimensions of the atrium. According to Vitruvius' literature, this room may be used to regulate some of the dimensions of the remaining rooms (e.g. *alae*, *tablinum*). Actually, the length of the atrium is sufficient, since its width and height may be determined through its length. This is demonstrated for the dimensioning of the *atrium* of a Roman house. According to Vitruvius' “De architectura”, there are three ways of dimensioning typical Roman *atriums*. Supposing that it is intended to use the first type of dimensioning, it is stated that its length is divided into five equal parts and then three parts are given to its width. This could be encoded by the following three rules:

SetProp (*Atrium*; *TypeDim*)
Atrium.TypeDim = 1
If (*Atrium.TypeDim* = 1; *Atrium.Width* = (*Atrium.Length*/5)3; 0)

In the first rule, an attribute named “TypeDim” is created for the shape *Atrium*, to identify the type of the dimensioning of the *atrium*. The second rule serves the purpose of setting the previously created attribute to the value “1”. The second rule will also act like a trigger to the third rule, since the condition (*Atrium.TypeDim* = 1) of the rule “If” will return true, therefore setting the *atrium* width to three thirds of its length.

4.2.1. Placement

The general idea of the algorithm is to use a multi-layer connection graph to guide the placement of each part of the house. This is done for each layer of the connection graph, and the geometric consistency observed in several heritage houses, led to the development of the container system, presented in Section 4.2.1, to easily arrange the different rooms in the physical space. The placement process was one of the arduous implementation steps of the floor plan creation process, due to several aspects involved. For example, the placement algorithm has to meet all the connections defined by the multi-layer connection graph. Moreover, the dimensions of each room have to be respected and, on some occasions, each part must be orderly placed. Additionally, often the process has also to deal with incomplete information (such as the dimensions of rooms). After all the rooms have been dimensioned in the first step of the algorithm, the whole connection graph is traversed and all the shapes (containers or rooms) are created, dimensioned and placed. This process is done in a post-order traversal sequence, since for client containers (the containers which dimensions are determined by the clients placed on it) it is only possible to determine their size after the size of its clients (other containers or rooms nested in) has been established, which is done in a recursive manner.

As the graph is traversed, the placement process has to deal with the correct positioning of each room inside the corresponding container and use several metrics to avoid empty spaces which lead to undesired floor plan incoherencies. This situation may happen since it is possible for a user to define the specific dimensions for each room, for these reason sometimes it is necessary to distribute a container empty space for each room place on it or to create more rooms to fill that space. This last solution seems to be a more appealing alternative in some cases, implying further research, reason why the author includes a way of creating new parts in the present algorithm.

Considering heritage structures such as Roman houses, one may notice that they follow some architectural coherence, observable in Fig. 6. This means that in the present example two containers may be used to group the *atrium* and *peristylium* areas. However, it would not make much sense filling the *atrium* containers with parts normally located on the *peristylium* area and vice-versa. For this reason, the general idea of the proposed approach is to create parts of the house which are valid within the container where they will be inserted, rather than simply creating random parts to fill up the empty space or even use some sort of subdivision process.

The present approach, together with the possibility of the L-system to provide different generation geometries, allows the generation of a large number of different models, which allows meeting some of the initial assumptions of the present work: allows dissimilarity among generated models and completes information when missing. Fig. 8 shows (on the top left corner) the floor plan of the Conimbriga's "Casa do Medianum Absidado" and five corresponding floor plan generations. Just some random generations were chosen, since the approach allows setting the desired number of models.

Finally, the last part of the placement process resides in the placement of shapes, which are intended to be centred or distributed over the edges of others shapes. This is only done at the final stage of the algorithm, since all the previous steps modify the position and the size of the several shapes where these last items are to be placed. For this reason, it would not make much sense to place centred and edge parts (shapes distributed over the edges), every time the shape where they are to be placed changes, since it would slow down the placement process.

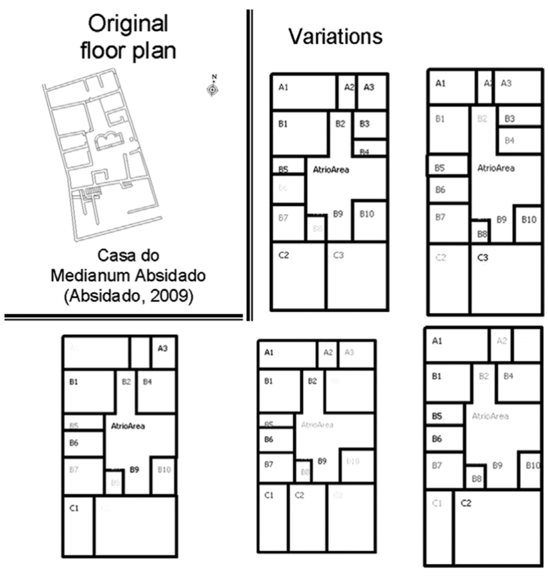

Fig. 8. Generated Conimbriga's models.

Centring a shape inside another, or distributing shapes along the edges of another shape, is simply a case of basic mathematics to determine the correct shapes' positions. The distribution of shapes is very valuable not only for the distribution over terminal shapes (e.g. a room of a house), but also in non-terminal shapes (e.g. containers) which may have no visual representation. With this feature, a great deal of physical structures may be distributed in a seamless way making the presence of the containers imperceptible. This approach was used in the placement of Roman columns among the *atriums* of several Roman Conimbriga's house models, such as the *tuscan* columns of the House of the Skeletons' (Roman heritage house, Conimbriga, Portugal) *atrium* which may be seen in Figs. 13 and 14 in chapter 7.

5. 3D model

The floor plan is the starting point for the creation of the 3D geometry, which is achieved in three steps. In the first step, walls are raised from the floor plan along with the creation of floors, ceilings, floors and materials such as mosaics. Then, in the second step, doors and windows are created (among other features) and in the last step, roofs are also created and added to the final model.

5.1. Walls, ceilings and floors

The creation of walls is supported by the dimensioning rules described in Section 4.2. Indeed, in some circumstances, this is led by Vitruvius' literature, which allows for the determination of the atrium height from its length. In other circumstances, default values may be used, based on information provided by experts (e.g. analysing existing characteristics of the house or similar houses). Then, for each room of the house (where applicable), a ceiling is created. Similarly, floors are created for each room, using the container system presented in Section 4.2.1.

5.2. Doors, windows and other features

Most known methods and tools for the automatic generation of models representing man-made structures usually rely on procedural techniques, which often produce models through the means of simple shape composition (or decomposition – depending on the technique). Without disregarding the efficiency of these techniques, one fact is that sometimes it is not easy (or it is even impossible) to faithfully reproduce complex objects using these techniques. In areas such as archaeology, one main imperative piece to recreate the past is the realism, which means that it is crucial to create features with a higher level of detail and sometimes it is advisable to use manually modelled objects, rather than to try to use some algorithm which would resemble the real structures that are pretended to be represented (either in reconstruction or generation). Moreover, often there are evidences which indicate that most probably, within a certain architectural style, similar structures use identical features.

This is what led the authors to consider an object database which may be used to represent some more specific house features. Once again, Roman architecture is an example of these specific features where in the example of the House of the Skeletons, several objects (e.g. *triclinium* window, *impluvium*, *tuscan* columns), represented in Fig. 9 were modelled in an external application (3D Studio Max), and placed accordingly to grammar rules.

Integrating these objects into the final geometry is simply a matter of placing it in the desired place (e.g. *impluvium*, columns) or by subtracting the corresponding object geometry to the wall where they are to be placed (e.g. windows).

This way, the present approach may be considered a hybrid one, since it combines procedural techniques along with manually modelled objects, thus allowing a more realistic appearance of the final models and gathering the best of the worlds: the efficiency of procedural techniques and the realism of manually modelling.

5.3. Roofs

Although it is possible to generate roofs using Straight Skeleton approaches (e.g. Felkel and Obdržálek (1998))) most of the times heritage structures have specific types of roofs. For example, a study of Roman architecture, along with the director of the Monographic Museum of Conimbriga – Doutor Virgílio Hipólito Correia, with the purpose of determining several options regarding the reconstruction of the House of the Skeletons, led to the development of two different types of roofs: roofs composed by two faces (*peristylium*) and roofs composed by only one face (*atrium*). Also, according to Roman architectural rules, an inclination angle of 25° was considered in both these types of roofs.

6. Dedicated exporter

The output of the method is produced by a dedicated exporter module capable of producing several formats according to the purpose of the models. Furthermore, the idea of this exporter module, although at its present state it only supports the X3D formats, is more than an all-purpose exporter that simply maps the geometry primitives to a specific format.

Indeed, the goal is to include a sub-module for each format which has to be capable of exploiting the optimisations of that specific format. For example, if the desired format is X3D, the final model ought to include these format optimisation capabilities, e.g. cloning, inlining, billboards, etc. This modularization allows for the inclusion of newer formats such as WebGL, a Javascript API based on web standards, that accesses GPU capabilities in the computer where the browser is running. Indeed some authors are already presenting 3D heritage models using WebGL (Ferreira et al., 2014).

This way, the exporter model is a specialised tool, which produces clever models where performance may be a critical issue, making those suitable to different fields of application (e.g. virtual worlds, video games and heritage sites).

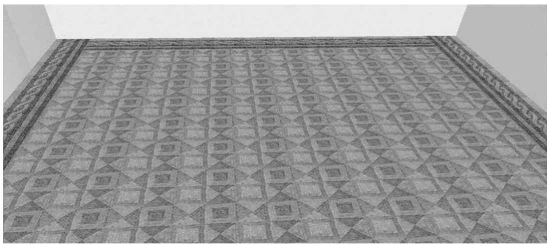

Fig. 11. House of the Skeletons' floor.

Fig. 12. Tuscan columns.

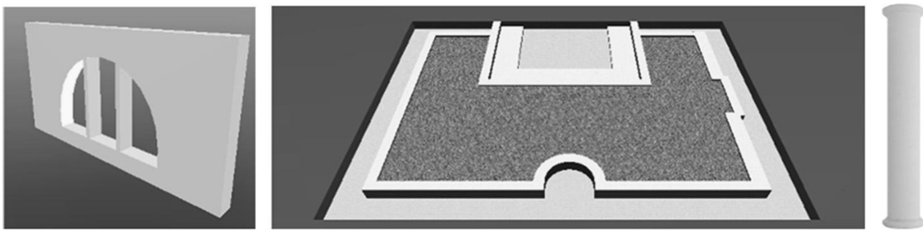

Fig. 9. External objects (Silva et al., 2004).

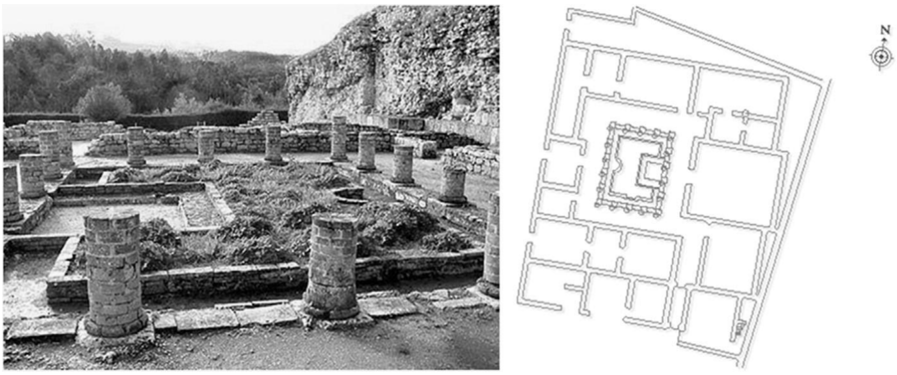

Fig. 10. Current aspect of central peristylium of the House of the Skeletons (left). Floor plan (right).

With the dedicated exporter all the internal operation of the general method is not affected by the output and goals of the models. Then, supporting new formats is only a question of adding new sub-modules and integrating them with the dedicated exporter, which is only responsible for translating internal geometry to final graphic primitives.

7. Results

This chapter presents some of the major results matching the two different purposes of the method: the reconstruction of heritage structures and the generation of new ones.

7.1. Reconstruction

The first results presented match one of the purposes of the method: the reconstruction of heritage structures, that no longer exist, for which some kind of knowledge is available (e.g. floor plans, photographs). The case study cope with the reconstruction of the House of the Skeletons (Conimbriga, Portugal) where though some ruins are still present, most of the house is destroyed.

In its reduced dimensions (945 m^2) the House of the Skeletons (the name came from historical factors) can be taken as a paradigm of Conimbriga's quality residences: quality of the architectural plan, economy of means, emphasis in the ornamental placement of the mosaics, intelligent use of the autonomous part of the construction. The analysis of the house comes across, though, with some inherent difficulties because the façade was destroyed by the construction of the late-imperial wall (Silva et al., 2004) which may be seen on the right side of the photograph of Fig. 10 (left).

The floor plan in Fig. 10 (right), designed from the ruins of Fig. 10 (left), was the only existing physical information of the House of the Skeletons. This fact led to the elaboration of a complete study of the Roman architecture, in collaboration with the director of the Monographic Museum of Conimbriga – Doutor Virgílio Hipólito Correia (Correia, 1993), with the purpose of inferring several options regarding the reconstruction of the House of the Skeletons. This study was partially based on Vitruvius' knowledge to determine several rules, referring to some features of the house, such as: walls' height, roof top disposition and the *peristylium* columns' height and position.

The study of Roman Architecture led to the development of two different types of roofs: roofs composed by two faces (*peristylium*) and roofs composed by only one face (*atrium*). According to Roman architecture rules, it was considered an inclination angle of 25°. In addition, the wood structure that supports the roof was also modelled and represented accordingly. As regards to the doors, only exterior doors were represented, since it is assumed that when the house was occupied it was not common to have interior doors. Instead, at the very least, they were used curtains in interior connections between rooms.

The more specific features of the house (e.g. columns, *triclinium* window) where modelled in an external application (3D Studio Max), added to the 3D object database and placed according to the grammar rules. This way the realism may be enhanced since some of the features may be modelled from existing evidence.

Several textures were also applied into the final model in order to enhance the realism. Here the most significant are highlighted. To represent the floors, the container system presented in Section 4.2.1, was used to place the mosaics on the floor of the house. These textures were reproduced from pictures taken from real mosaics, which are conserved in the Monographic Museum of Conimbriga. Fig. 11 shows the result applied to the floor surface of the House of the Skeletons.

Another important feature worthy of notice is the inclusion of the *tuscan* columns (shown in Fig. 12), manually modelled by Gonçalves (2002), but placed automatically by the container system.

In Fig. 13 the results for the reconstruction of the House of the Skeletons are presented.

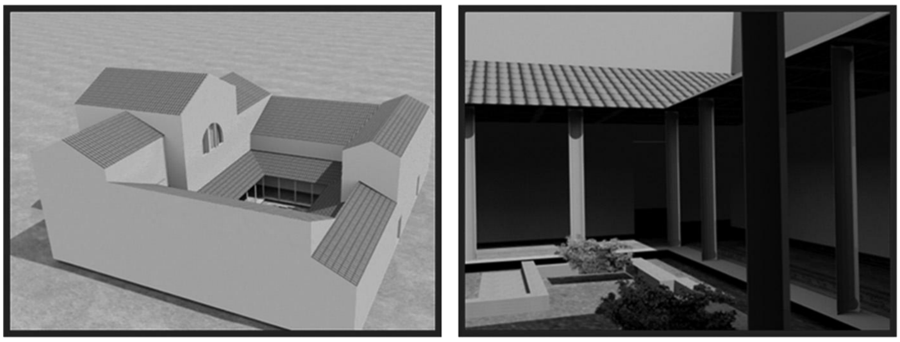

Fig. 13. Rendered model of the House of the Skeletons.

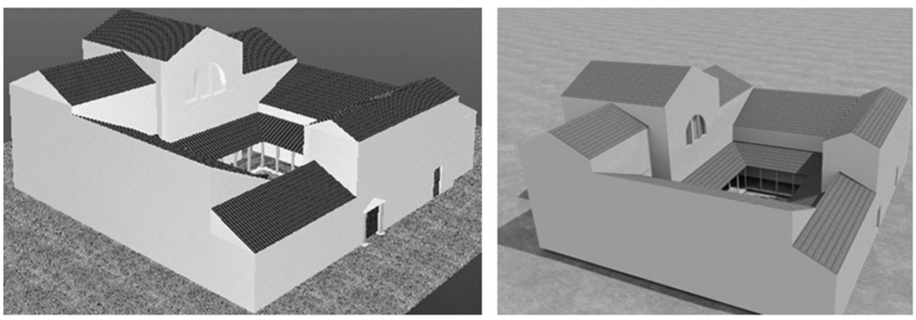

Fig. 14. House of the Skeletons; left: manual model, right: grammar model.

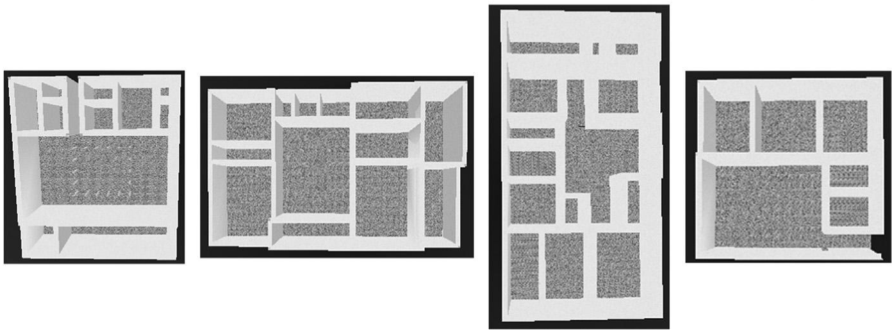

Fig. 15. Generated 3D models of Conimbriga's houses.

Table 1
Generation of Conimbriga's houses.

Number of houses	Time (s)
1	~1
10	~11
100	~92
1000	~936

When compared to a manually designed model of the House of the Skeletons the reconstruction accomplished with the presented technique shows very similar features. Fig. 14 shows the manual and grammar models.

7.2. Generation of Conimbriga's houses

There is not a production ready application yet that produces completely traversable houses with a high level of detail for every architectural style. Still, when this paper was written, there was already the possibility to generate a large number of different models with distinguishable dissimilarity taking as reference known buildings. This is present in the floor plans shown in Fig. 8, of past Section 4.2.1, and it is also possible to generate simplified 3D models featuring the base geometry (walls, floors, ceilings) of the houses and also some ornaments (e.g. columns).

Fig. 15 presents some of these 3D models generated for known Conimbriga's Roman houses. Once again, just some random generations were chosen.

Table 1 presents a performance test concerning the generation of several variations based on a Conimbriga's house (House of the Skeletons), with an average of 21 rooms, distributed by 9 containers. The increase of the number of models generated is rather linear with the time the generation takes. This is due to the algorithm features, where every generation is focused on the particular house, and does not increase the processing complexity. Note that no performance issues were taken into account, concerning that the generation time and the presented time values include the creation of the 2D (jpg) file, representing the floor plans, and the 3D files (X3D), representing the whole three dimensional geometry. The test was conducted on a system equipped with an Intel Core 2 running at 2.4 GHz with 1 GB of RAM.

8. Conclusion and future work

The method presented in this paper represents an approach aiming for the reconstruction and generation of heritage houses, which may be used in areas such as archaeology, but also others such as video games, animation movies or even virtual worlds. For this reason a grammar was proposed to encode different architectural styles. This way a house organisation may be defined using an L-system alongside with a multi-layer connection graph. The whole method was applied both in the reconstruction and generation of houses of a specific architectural style, i.e. Roman Architecture houses, from information gathered from the existing ruins, Vitruvius' rules and the rules given by an expert from the Monographic Museum of Conimbriga, Portugal – Doutor Virgílio Hipólito Correia. The results attained lead to believe that more complete applications may be developed over the proposed method to generate completely traversable heritage houses as well as other structures (e.g. temples, forums, etc.). Indeed, the randomness features, which allows the generation of a large variety of different houses within a certain architectural style, comprise an important role in the future development of tools to generate entire cities.

Certainly, the ability of future software to produce complete environments will be a valuable asset allowing archaeologists, historians or other experts who want to produce their own models. This way, expert tools may be used either to try to recreate the past or to conceive new theories based on fictional generated models.

Acknowledgements

The authors thank the Director of the Monographic Museum of Conimbriga, Doutor Virgílio Hipólito Correia, for the support concerning the information about Roman Architecture essential to reconstruct and generate Conimbriga's houses.

References

Aliaga, D.G., Rosen, P.A., Bekins, D.R., 2007. Style grammars for interactive visualization of architecture. IEEE. Trans. Vis. Comput. Graph. 13 (4), 786–797.
Bessa M., Coelho A.F.V.C.C., Moura J.P., Ferreira F.N., Cruz J.B., Sousa A.A.D., 2005. Modelação expedita de ambientes virtuais urbanos para utilização em dispositivos móveis. In 13° Encontro Português de Computação Gráfica, pp. 85–93.
Bostrom G., Fiocco M., Puig D., Rossini A., Goncalves J.G.M., Sequeira V., 2004. Acquisition, Modelling and Rendering of Very Large Urban Environments. In: 3D Data Processing, Visualization, and Transmission (3DPVT), 2nd International Symposium, pp. 191–198.
Charbonneau N., Boulerice D., Booth D.W., Tidafi T., 2006. Understanding Gothic Rose Windows with Computer-Aided Technologies. In: 24th eCAADe Conference, pp. 770–777.
Coelho, A., Bessa, M., Sousa, A.A.d., Ferreira, F.N., 2007. Expeditious modelling of virtual Urban environments with geospatial L-systems (6 June). Comput. Graph. Forum 26 (4), 769–782.
Chen, G., Esch, G., Wonka, P., Müller, P., Zhang, E., 2008. Interactive procedural street modeling. ACM Trans. Graph. 27 (3), 1–10.
Chomsky, N., 1956. Three models for the description of language. IRE Trans. Inf. Theory 2 (3), 113–124.
Correia V.H., 1993. O Futuro dos Estudos Arqueológicos em Conimbriga. In: Correia, V.H. (Ed.). Perspectivas sobre Conimbriga.
Empire, 2007. The Roman Empire.

Frischer B., Abernathy D., Guidi G., Myers J., Thibodeau C., Salvemini A., Muller P., Hofstee P., Minor B., 2008. Rome Reborn. In: ACM SIGGRAPH new tech demos (SIGGRAPH'08), Article 34.
Felkel P., Obdržálek Š., 1998. Straight Skeleton Implementation. In: 14th Spring Conference on Computer Graphics, pp. 210–218.
Ferreira C., Rodrigues N., Gonçalves A., Hipólito-Correia V., 2014. Reconstructing Conimbriga – Digital Cantaber. In: IVAPP'2014 – 5th International Conference on Information Visualization Theory and Applications.
Fuchs A., 2006. Outils numériques pour le relevé architectural et la restitution archéologique, vol. PhD.
Guidi, G., Frischer, B., Lucenti, I., Donno, J., Russo, M., 2008. Virtualizing ancient Imperial Rome: from Gismondi's physical model to a new virtual reality application. Int. J. Dig. Cult. Herit. E-Tour., 240–252.
Guidi G., Frischer B., De Simone M., Cioci A., Spinetti A., Carosso L., Micoli L.L., Russo M., Grasso T., 2005a. Virtualizing ancient Rome: 3D acquisition and modeling of a large plaster-of-Paris model of imperial Rome. In: Proceedings of the SPIE Videometrics VIII, vol. 5665, pp. 119–133.
Guidi G., Micoli L., Russo M., Frischer B., Simone M.D., Spinetti A., Carosso L., 2005b. 3D Digitisation of a Large Model of Imperial Rome. In: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling.
Gonçalves A., 2002. Reconstrução de ambientes históricos utilizando VRML: o caso do Fórum Flaviano de Conimbriga, vol. MSc.
Greuter S., Parker J., Stewart N., Leach G., 2003. Real-time procedural generation of 'pseudo infinite' cities. In: Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, pp. 87-ff.
Ingram R., Benford S., Bowers J., 1996. Building Virtual Cities: applying urban planning principles to the design of virtual environments. In ACM Symposium on Virtual Reality Software and Technology (VRST'96), pp. 83–91.
ISO/IEC14977, 1996. ISO/IEC 14977: 1996(E).
Kimberly D., Frischer B., Mueller P., Ulmer A., Haegler S., 2009. Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques. In: CAA 2009, pp. 62–66.
Lindenmayer, A., 1968. Mathematical models for cellular interaction in development, Parts I and II. J. Theor. Biol. 18, 280–315.
Lechner T., Watson B., Wilensky U., 2003. Procedural city modeling. In: 1st Midwestern Graphics Conference, pp. 1–6.
Lipp M., Wonka P., Wimmer M., 2008. Interactive visual editing of grammars for procedural architecture. In: ACM SIGGRAPH.
Luca, L.D., Véron, P., Florenzano, M., 2007. A generic formalism for the semantic modeling and representation of architectural elements. Vis. Comput. 23 (3), 181–205.
Martin J., 2005. The Algorithmic Beauty of Buildings: Methods for Procedural Building Generation, vol. Honors Thesis.
Merrell P., 2007. Example-based model synthesis. In: 2007 Symposium on Interactive 3D Graphics and Games, pp. 105–112.
Müller P., Vereenooghe T., Ulmer A., Gool L.V., 2005. Automatic Reconstruction of Roman Housing Architecture. In: International Workshop on Recording, Modeling and Visualization of Cultural Heritage, pp. 287–298.
Müller P., Vereenooghe T., Wonka P., Paap I., Gool L.V., 2006a. Procedural 3D Reconstruction of Puuc Buildings in Xkipche. In: Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST), pp. 139–146.
Müller, P., Wonka, P., Haegler, S., Ulmer, A., Gool, L.V., 2006b. Procedural modeling of buildings. ACM Trans. Graph. 25 (3), 614–623.
Park S.M., 2005. Tall building form generation by parametric design process, vol. PhD.
Parish Y.I.H., Müller P., 2001. Procedural modeling of cities. In: 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 301–308.
Rodrigues N., Dionísio M., Gonçalves A., Magalhães L., Moura J.P., Chalmers A., 2008a. Incorporating Legal Rules on Procedural House Generation. In: Spring Conference on Computer Graphics, pp. 69–76.
Rodrigues, N., Dionísio, M., Gonçalves, A., Magalhães, L., Moura, J.P., Chalmers, A., 2008b. Rule-based Generation of Houses. Comput. Graph. Geom. 10 (2), 49–65.
Rodrigues N., Magalhães L., Moura J.P., Chalmers A., 2007. Geração Automática de Estruturas Romanas. In: CAA Portugal.
Rodrigues N., Magalhães L., Moura J.P., Chalmers A., 2008c. Automatic Reconstruction of Virtual Heritage Sites. In: Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 08), pp. 39–46.
Rodrigues N., Magalhães L., Moura J.P., Chalmers A., Santos F., Morgado L., 2009. ArchHouseGenerator – A Framework for House Generation. In: Slactions 2009 International Conference, Life, Imagination, and Work Using Metaverse Platforms, pp. 11–20.
Silva F., Rodrigues D., Gonçalves A., 2004. House of the Skeletons – A Virtual Way. In: XXXII CAA – Computer Applications and Quantitative Methods to Archaeology Conference.
Willmott J., Wright L.I., Arnold D.B., Day A.M., 2001. Rendering of large and complex urban environments for real time heritage reconstructions. In: Conference on Virtual Reality, Archaeology, and Cultural Heritage, pp. 111–120.
Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W., 2003. Instant architecture. ACM Trans. Graph. 22 (3), 669–677.