

Automatic Reconstruction of Virtual Heritage Sites

N. Rodrigues¹, L. Magalhães², J. Moura^{2,3} and A. Chalmers⁴

¹Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal

²Universidade de Trás-os-Montes e Alto Douro, Portugal

³Knowledge Engineering and Decision Support Research Center, Porto, Portugal

⁴The Digital Laboratory, WMG, University of Warwick, UK

Abstract

The virtual reconstruction of heritage sites has been the focus of many projects. These typically involve significant use of manual reconstruction techniques, and thus a great deal of human effort to create the virtual structures. Also, often, there is not sufficient physical evidence to recreate these structures precisely as they may have been in the past.

To address these issues a domain specific modelling method for the automatic generation of virtual heritage structures is presented in this paper. The method is guided by heritage knowledge about the construction rules of heritage structures, encoded in a formal grammar, and may be used to create new structures automatically. The case study entails the automatic reconstruction of the archaeological site of Conimbriga, in Portugal, which contains the ruins of an ancient city of the Roman Empire. The results show the generation of a virtual reconstruction of a particular house, the House of the Skeletons, which had an important relevance to the city because of its architecture.

Categories and Subject Descriptors (according to ACM CCS): F.4.2 [Mathematical Logic and Formal Languages]: Grammars and Other Rewriting Systems – Grammar types, Parallel rewriting systems, Parsing; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Virtual Reality; I.3.8 [Applications].

1. Introduction

In the recent past, computer algorithms have been used to generate virtual environments, where one of the most common applications is the generation of human-made structures. Buildings and indeed entire cities have been reconstructed and new worlds generated. The main focus of these efforts is to place most of the hard work of creating virtual environments on computer algorithms rather than on human resources. Thus the latter may use the spared time to enhance the final models or thinking about new ways to increase their realism.

In this paper, a method is proposed for the automatic generation of heritage sites with particular focus on Roman civilization structures, where several rules were written based on the knowledge left by Vitruvius (Marcus Vitruvius Pollio – Roman architect and engineer who lived in the 1st century BC, author of “De architectura”), mostly through the reading of the Portuguese adaptation from M. Justino Maciel “Tratado de Arquitectura” [Mac06]. A study of Roman Architecture, along with the director of the Monographic Museum of Conimbriga – Dr. Virgílio

Hipólito Correia, with the goal of determining several options regarding the reconstruction of the House of the Skeletons, is also responsible for most of the rules written to produce the results presented.

The use of real rules to support the generation of virtual environments has been recently explored by the authors of this paper, either on the generation of modern structures [RDG*08] or on the generation of ancient civilizations (e.g. Roman civilization) [RMMC07]. For the generation of modern houses the authors presented a method that took into consideration strict legal rules, provided by RGEU (Regulamento Geral das Edificações Urbanas, General Regulations for Urban Buildings), which is one of the main documents whereby all Portuguese architectural projects have to comply.

For the generation of ancient civilizations, we have presented an initial approach to extend the method applied on modern houses, to Roman structures, where the generation is guided by Vitruvius’ rules.

Further more in this paper new developments and techniques to our previous work are introduced to establish a new method for the generation of heritage sites, such as:

- A grammar capable of encoding Vitruvius' rules.
- A parser which interprets the rules of the proposed grammar proceeding accordingly.
- A multi-layer connection graph representing the connections and relations between the different parts of Roman structures.
- A container system to improve the spatial distribution of physical elements over the scene.

The method convenes on two different purposes. The first one is the generation of new structures where the structures are generated through the use of an L-System. The second one is the reconstruction of heritage structures no longer existing for which some kind of knowledge is available (e.g. floor plans, photographs). This last one plays the main role in this paper where the example of the reconstruction of the House of the Skeletons is presented.

1.1 Related work

In the field of Procedural Modelling, the approaches are based on algorithms to automatically generate virtual structures. These methods may be used for the creation of new worlds. Additionally, they may also be used for the reconstruction of non-existing worlds, for which some kind of knowledge is available (e.g. floor plans, photographs), to support the reconstruction of realistic environments.

One initial approach to the construction and analysis of architectural design is based on the shape grammars, presented in [Sti75], which have become the foundation for many applications comprising different architectural styles (e.g. [KE81, Kni81, Fle87, Dua02]). Indeed, most of the procedural techniques for the generation of virtual environments use grammars to describe the structures being generated, either these are based on shape grammars or some more specific grammars (e.g. L-Systems [PM01], split grammars [MWH*06]).

Most techniques for procedural building generation mainly concentrate on the generation of modern structures. In the past few years several methods have been presented (e.g. [PM01, GPSL03, WWSR03, FBS05, Mar05, MWH*06]) which address different aspects for Procedural Urban Modelling. Most of them are discussed by Watson et al. in [WMV*08] in addition to several other aspects, advantages and practical applications of this promising area.

A common feature among these works is that the main effort is on the generation of the structures' exteriors. In most of the approaches the interior of the structures is not generated, thereby they are not traversable. This feature may be adequate to some fields of application although they are not applicable when this feature is necessary. In [Mar05], the author addresses this issue by presenting two different approaches, using graphs which represent the rooms of a house and the connections between them, to generate modern traversable houses. In [HBW06], Hahn et al. also address interior generation, in real time, by randomly dividing rectangular floors, corresponding to buildings' interiors, into rectangular rooms and hallways.

Both these methods ([Mar05] and [HBW06]) which address interior generation still prove to be unsuccessful if the goal is the generation of structures which may represent real buildings. In [HBW06] the floors are randomly divided and though in [Mar05] the author took some architectural aspects under consideration (the author acknowledge Christopher Alexander's "A Pattern Language" [AIS97], which is also recognized by several others authors), the generated models still lack some realism.

In what concerns culture heritage structures, in [MWH*06], some authors of the previously mentioned literature used prior knowledge to propose a shape grammar, named *CGA shape*, which may be used to generate different kinds of three-dimensional models, including heritage environments. In [MVUG05], Roman housing architecture were also addressed and CityEngine (a system introduced by Parish and Müller in 2001) used. Indeed one of the examples presented is the modelling of Pompei (Roman city destroyed in 79 AD) which was done in collaboration with archaeologists. In [MVW*06], Müller et. al used the same techniques to procedurally create 3D reconstructions of stone houses in Xkipché supported by archaeological research.

In this paper the centre of attention, of the presented method, goes to the individual building and to the heritage architectural rules by which they were built to generate complete 3D traversable models.

1.2 Overview

This paper is structured as follows: Sections 2, 3, 4 and 5 describe each step of the proposed method for the generation of heritage structures. Section 6 shows some results achieved with the proposed method, applied to the construction of the House of the Skeletons and in Section 7 some conclusions and plans for future work are presented.

2. Generation of heritage structures

The following method is intended for the generation of heritage structures, although for demonstration purposes it is applied to the particular case of Roman structures. Likewise Roman houses are the presented case study.

The method is composed of three major steps as presented in Figure 1. In the first step (Organization) a list of rooms is generated (and other physical structures), either by an L-System or through user rules which is then grouped in a multi-layer connection graph defining the connections between them.

The second step (Floor Plan) is responsible for the creation of the bi-dimensional structures and distributing them in the physical space creating a floor plan, recurring to a container system.

The third step (3D Model) is responsible for the creation of the three-dimensional geometry which is then mapped to the screen.

The whole method is guided by a grammar to automatically create the structures, either by heritage knowledge

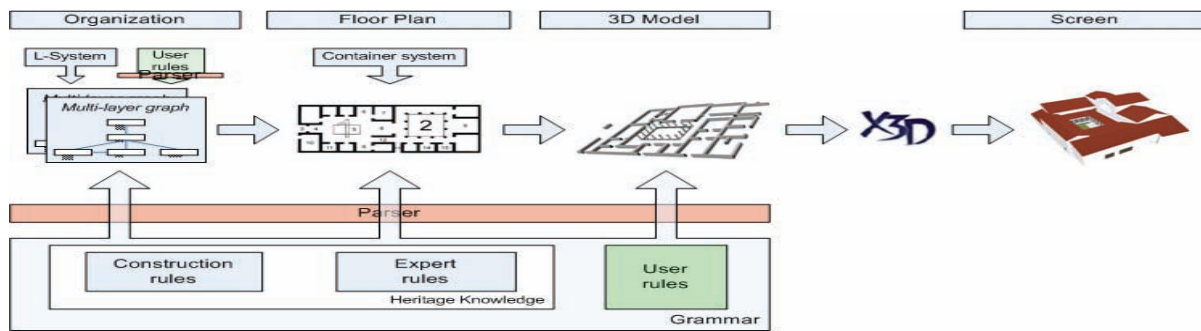


Figure 1: Method for the generation of heritage structures.

composed by a heritage construction set of rules (Vitruvius's in the present case study) together with Expert rules (rules inferred from experts on the particular case study) or by specific rules declared by the user (User rules). 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; on the other side the list of structures may be specified by the user (but created through the rules he specifies, i.e. without any visual interaction). 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 goal is the reconstruction of heritage structures no longer existing, the user may define suitable rules obtained through archaeological data to reconstruct these structures. For these reasons the user rules are represented twice in Figure 1.

2.1 Roman architecture

One of the most prominent features of Roman houses resides in the vast diversity of floor plans, which has changed over the years due to geographic adaptation and to the status and position of the owner. Therefore, there is no rigid organization determining the type and number of rooms of a house, as well as the physical organization, that is disassociated from these aspects. Though there are frequent typical organizations (Figure 5, in Section 4.2, represents the simplified typical organization of a classic Roman house).

One feature of the classic Roman house, i.e. a house featuring the most common rooms and organization, is that it is divided in two parts: a part grouped around the *atrium* and a part grouped around the *peristylum*. This organization is the source for writing a set of rules encoded in the grammar developed in Section 2.2.

2.2 Generation grammar

Although the grammar allows for the definition and transformation of shapes, it has more general purpose than a proper shape grammar, which is why the authors decided to refer to it as Generation Grammar.

Beside the definition of shapes, the grammar encodes several other features that augment the capabilities of the proposed grammar, including:

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.

Beside these specific examples there are also several other features of the grammar (including simple arithmetic operations, conditional rules such as “If” rules, and the possibility to nest rules inside other rules) which effectively give an adequate level of control over the whole geometric structures creation process.

One thing to notice is that most rules allow reference either to a set of similar shapes (e.g. a rule that refers to a set of cubiculum will affect all the cubiculum shapes) or to a specific shape through a unique identifier which allows to apply the rule only to one shape (e.g. a rule that refers to a cubiculum ID will only affect that specific cubiculum). This is particularly useful to define general properties (properties of a type of shape) and specific properties (properties of a specific instance of a shape).

Notation

All the rules are available to the user in a formula like notation. This notation, represented in (1), is similar to the common spreadsheet formulas in today's software, making the coding of the rules very easy since wizards for each rule may also be available.

$$\text{Rule}_n(\text{Parameters}) \quad (1)$$

“Parameters” correspond to the accepted parameters for each rule which may be one or combinations of the following: simple arithmetic operations, constants, shapes, shape attributes, nested rules.

For example to define a 4 m long and 3 m wide cubiculum the following rules could be used:

```
defineShape(cubiculum)
setProperty(length; cubiculum; 4)
setProperty(width; cubiculum; 3)
```

The process may also be further simplified with a visual editor making it easier to define shapes and encode rules. Indeed this is the logical process to pursue to simplify the user's work [LWW08].

2.3 Parser

Although the presented grammar seems to be an important tool to define the rules which will guide the whole system, it would not be of much use without an adequate parser (represented in Figure 2).

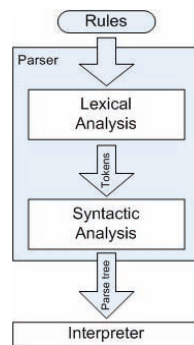


Figure 2: Parser.

The parser takes as input a text file with all the rules and then in the first stage, lexical analysis, it 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.

3. Organization

In the first step of the method a list of rooms is generated, 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 organization of a house which will guide the generation of the floor plan.

One main distinction between the L-System used in the present method and previous approaches (e.g. [PM01]) is that this one is used to define the interior features of a house (i.e. rooms, ornaments, etc.) instead of defining operations between shapes or defining pathways and populate them with different objects (e.g. buildings, trees, persons).

3.1 L-System

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 Roman house. Assuming that the axiom (ω) corresponds to the symbol *RomanStructure* from the V alphabet and the productions (p_0, p_1, \dots, p_n) correspond to the possible replacements between the sym-

bols, then an example of a Roman house may be represented as:

$$V = \{RomanStructure, RomanHouse, AtriumArea, Atrium, Ala, Cubiculum, \dots, PeristylumArea, Peristylum, \dots\}$$

$$\omega = RomanStructure$$

$$p_0: RomanStructure \rightarrow RomanHouse$$

$$p_1: RomanHouse \rightarrow AtriumArea, PeristylumArea$$

$$p_2: AtriumArea \rightarrow Atrium, Ala, Cubiculum, \dots$$

$$p_3: PeristylumArea \rightarrow Peristylum, Cucina, Bathroom, \dots$$

$$p_4: Cubiculum \rightarrow Cubiculum, Cubiculum$$

This process may be understood as a decomposition system, where every symbol is iteratively replaced until only terminal symbols, which correspond to different parts of the house, are reached. This is demonstrated in Figure 3, for the presented L-System.

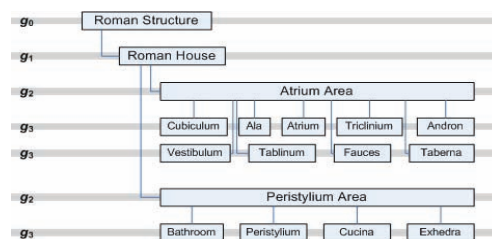


Figure 3: L-System generations.

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.

This process may be understood as a decomposition system, where every symbol is iteratively replaced until only terminal symbols, which correspond to different parts of the house, are reached.

3.2 Multi-layer connection graph

In this step, the list of rooms is added to a graph representing the connections between the rooms. This graph serves the purpose of leading to the next step where the position of each room of the house has to be placed next to any other room where it links to. Moreover it also serves the purpose of determining where passages or doors have to be placed.

One thing to notice, beside the definition of the structure of the specific parts, is that the connection graph algorithm also allows for the definition of containers, making this a powerful tool. This means that, when containers are used, there are really two types of graphs created: container graphs, and part graphs. Since a container may also have other containers on it, the connection graph algorithm may lead to a generation of several graphs. This is represented in Figure 4.

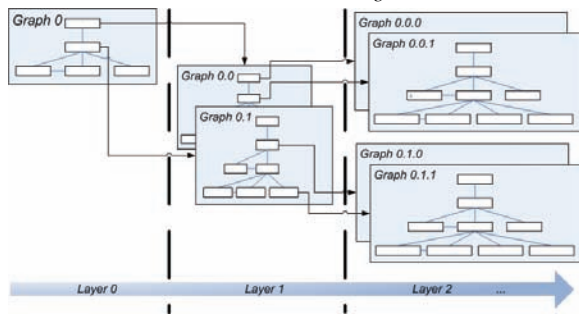


Figure 4: Multi-layer connection graph.

When containers are present the algorithm simply starts at the higher level of containers and creates a graph which represents the connections between them. To do this the algorithm simply iteratively removes from the list of rooms generated by the L-System (or by the user rules), each 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 house parts in the graph.

4. Floor plan

The results from the “organization” step are used to guide the next step of the method: the generation of the floor plan. This is further divided into two different sub-steps: the dimensioning and the spatial distribution.

The first sub-step, dimensioning, is responsible for setting all the rooms of the house with their proper dimensions. The second, spatial distribution, is responsible for placing all the elements in the bi-dimensional physical space according to the multi-layer connection graph from the first sub-step, resulting in a floor plan. To aid this step a container system is also introduced. Although it is not strictly necessary in this step it is also used to map textures (see Section 6.3).

4.1 Dimensioning

The floor plan generation uses the information from the grammar rules to define the dimensions of each part of the house. When the user does not specify these features, default values are used and the whole generation follows Vitruvius’ rules.

However, if the user decides to define the dimensions these will override the predetermined rules from the grammar. This is what happens in the presented case study where most of the dimensions of the different parts of the House of the Skeletons were known.

Most of the rules serve as a trigger for other rules. For example, one starting point to generate a Roman house may be defining the dimensions of the *atrium*. This room regu-

lates some of the dimensions of the remaining rooms (e.g. *ala*, *tablinum*). Actually the length of the *atrium* is sufficient since its width and height are determined through its length (according to Vitruvius’ rules).

4.2 Spatial distribution

As in the dimensioning of rooms, the user may also override the predetermined rules from the grammar. In previous work [RDG*08] the room placement uses a greedy algorithm along with a grid to make the number of possible positions for each room finite. This technique proved to be successful in the generation of modern houses constrained to a number of legal rules.

In this work a similar approach is proposed, in which the general idea of the algorithm is to use the multi-layer connection graph presented in Figure 4 to place each part of the house. This is done for each layer of the connection graph and the geometric consistency of Roman houses led to the development of a container system to easily arrange the different rooms in the physical space.

The container system

The shapes are divided into two types: terminal shapes, which represent the final geometric feature (e.g. *cubiculum*) or non-terminal shapes (e.g. container).

The “terminal” term means that there are no other parts of the house contained in this shape. Though, it may contain some other ornaments or even containers that may be useful to align some features (e.g. textures).

The container is a shape, which may or not have a visible representation, and serves the purpose of aligning all the other shapes which are placed inside it. A container may also contain other container objects. Moreover it is also possible for any shape to have a container property thereby having the functionalities of a container.

The container approach is somewhat similar to graphic containers which are available for programmers to use in forms when they are designing user interfaces. However it has some more characteristics, including the possibility to align a shape centred in the container and the possibility to size the container automatically depending on the size of the clients placed on it. There are two types of containers: absolute containers and client containers. The first one matches the containers which have fixed dimensions (width and length) and the second to the containers whose dimensions are determined by the clients placed on it. Furthermore, there are two ways of placing a client into a container: by placing the type of desired alignment (e.g. left alignment, right alignment) and by specifying an absolute position of placement (x , y coordinates). It is also possible to distribute objects over a container. This is the example of placing the *peristylum* columns along an edge of a container (see item n° 2 in Figure 5).

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 Roman house. Although the greater

advantages of the system are mostly achieved when the houses have highly aligned distribution of rooms as in the classic Roman house, in houses with more rough distributions it may also be a helpful tool.

The use of containers is demonstrated in Figure 5 where several containers may be used to align the different features of a Roman house with a classic disposition.

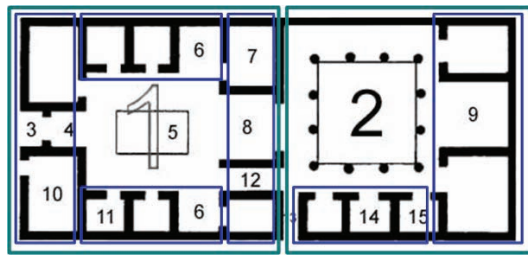


Figure 5: Use of containers in a classical Roman house.

Containers may also be used on any shape face. This is particular useful to distribute textures over a shape 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 6.3.

5. 3D model

3D geometry generation starts with wall creation where, for each room of the house, a wall is raised for each edge represented in the floor plan. At the same time, i.e. when each room is traversed, floors, ceilings and roofs are also created. Notwithstanding that for the creation of roofs, due to their specificities and simplicity, conversely to previous work there was no need to follow a “Straight Skeleton approach” (or similar) as the one proposed by Felkel and Obdržálek in [FO98].

After creating the basic geometry for each room, doors and windows are also integrated in the model by subtracting the corresponding geometry to the wall where they are to be placed. Finally, some other ornaments (e.g. columns) are also added and the final scene is mapped to the screen where X3D is used. Some screenshots of the House of the Skeletons are shown in Section 6.

6. Results

The results presented match one of the purposes of the method: the reconstruction of heritage structures, no longer existing, 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 where, though some ruins are still present, most of the house is destroyed.

In its reduced dimensions (945 m²) the House of the Skeletons (the name came from historical factors) can be taken as a paradigm of the residences of quality of Conimbriga: 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 [SG04] which may be seen on the right side of the photograph (left) of Figure 6.



Figure 6: Current aspect of central peristyle of the House of the Skeletons (left). Floor plan (right).

6.1 Modelling

Although, due to its specific features this house can not be automatically generated by the L-System, it still serves the purpose of assess the grammar. As stated in the beginning of this paper, besides the fact that the presented method allows for the generation of new structures, it also supports the automatic generation of structures for which some features are known. In this case study, the floor plan represented in Figure 6 (right), represents the basis for writing the rules which led to the creation of the House of the Skeletons 3D model.

The floor plan, designed from the ruins of Figure 6, was the only existing 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 – Dr. Virgílio Hipólito Correia [Cor93], with the goal 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 as walls height, roof top disposition and the *peristylum* columns height and position.

6.2 Doors, windows and roofs

The study of Roman architecture led to the development of two different types of roofs: roofs composed by two faces (*peristylum*) and roofs composed by only one face (*atrium*). An inclination angle of 25° was considered, according to Roman architecture rules.

In addition the wood structure that supports the roof was also modelled and represented accordingly. The complete roof structure is illustrated in Figure 7.



Figure 7: House of the Skeletons – roof structure.

Regarding the doors, only exterior doors were represented, since it is assumed that at the time the house were in use it was not common to have interior doors. Instead, at the very least, in interior connections between rooms, curtains were used.

The more specific features of the house (e.g. columns, triclinium window) were modelled in an external application (3ds Max), and placed according to the grammar rules. This allows for a library of objects (doors, windows, ornaments) that even though manually created are automatically placed. The advantage is the enhanced realism that may be achieved this way, where some of the features may have been modelled from existing evidence.

6.3 Textures

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 was used to reproduce the mosaics according to some models conserved in the Monographic Museum of Conimbriga. Figure 8 shows the result applied to the floor surface of the House of the Skeletons.

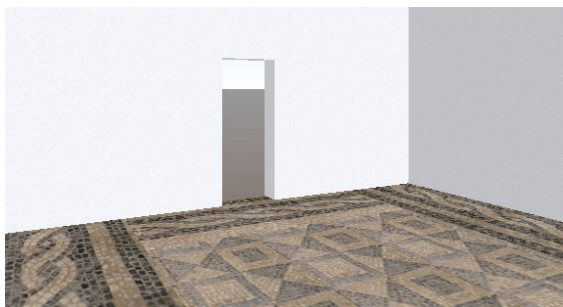


Figure 8: House of the Skeletons – floor.

Another important feature worthy of notice is the representation of the Tuscan columns. To achieve this, a technique presented in [SG04] was used, which consists of

combining a half-pipe and a texture to achieve the result previously presented in Figure 7.

The method also makes use of billboards (X3D grouping node). The coordinate system is modified so that the billboard always turns to point at the viewer independently of its position in the world. The billboards benefit from the symmetry of the columns and the number of columns to present in the 3D model of the House of the Skeletons. The combination of the billboard, along with the textured cylindrical object, results in a solution with more realistic lighting which has performance advantages.

In all the above cases, as well as in the other places where textures were applied, it is possible to specify by means of the grammar which texture to use. So for the specific features of a Roman house, a library can also be used containing several Roman textures.

6.4 Encoding Vitruvius' rules

The proposed grammar in Section 2.2 is powerful enough to encode Vitruvius' rules. It even allows the user to create attributes over shapes and then write rules over those attributes. This is demonstrated for the dimensioning of the *atrium* of a Roman house. According to Vitruvius "De architectura", typical Roman *atria* have three types of possible dimensioning. 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)
```

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 of the rule "If" will return true, therefore setting the *atrium* width to three thirds of its length.

7. Conclusion and future work

In this paper a method for the generation of heritage structures is presented, in particular Roman structures. Several rules were written based on the knowledge left by Vitruvius and tested against a practical example of the House of the Skeletons. The results show that the grammar is flexible enough to encode several rules regarding Roman structures.

The whole method partially tested in previous work on the generation of modern structures and now extended to cover a broader range of applications, shows satisfying results in the generation of 3D geometry of Roman structures. The technique presented here considerably improves previous approaches. The multi-layer connection graph has an important role in the organization of Roman houses and the container system also proves to be a useful tool, not

only in the placement of rooms, but also in the mapping of textures.

Some further work is still necessary regarding several aspects of the method, including the development of a visual interface to interact with the grammar and allowing new geometries for the structures (e.g. circular, octagonal).

Finally, as mentioned in the beginning of this paper, this is part of a project which entails the automatic reconstruction of the archaeological site of Conimbriga, in Portugal. There is thus still a great number of structures to be generated (e.g. Flavian Forum) to investigate the method with the complete generation of a Roman city. Once this is completed, it will be submitted to the scrutiny of experts in Roman architecture.

Acknowledgements

The authors thank the Director of the Monographic Museum of Conimbriga, Dr. Virgílio Hipólito Correia, for the support concerning the virtual reconstruction of the House of the Skeletons and the necessary information about Roman Architecture essential to create the grammar rules.

References

- [AIS97] ALEXANDER C., ISHIKAWA S., SILVERSTEIN M.: *A Pattern Language*. Oxford University Press. 1997.
- [Cor93] CORREIA V. H.: *O Futuro dos Estudos Arqueológicos em Conimbriga*. In Correia, V. H. (Ed.) *Perspectivas sobre Conimbriga* (Lisboa, Ed. Âncora), P. 49-80. 1993.
- [Dua02] DUARTE J.: *Malagueira Grammar – towards a tool for customizing Alvaro Siza’s mass houses at Malagueira*. PhD thesis, MIT School of Architecture and Planning. 2002.
- [FBS05] FINKENZELLER D., BENDER J., SCHMITT A.: *Feature based Decomposition of Façades*. In Proc. of Virtual Concept, Biarritz, France. 2005.
- [Fle87] FLEMMING U.: *More than the sum of its parts: the grammar of Queen Anne houses*. Environment and Planning B: Planning and Design 14 (1987): 323-350. 1987.
- [FO98] FELKEL P., OBDRŽÁLEK S.: *Straight Skeleton Implementation*. In Proc. of the 14th Spring Conference on Computer Graphics, Budmerice, Slovakia, 210-218. 1998.
- [GPSL03] GREUTER S., PARKER J., STEWART N., LEACH J.: *Undiscovered Worlds – Towards a Framework for Real-Time Procedural World Generation*. Fifth International Digital Arts and Culture Conference, Melbourne, Australia. 2003.
- [HBW06] HAHN E., BOSE P., WHITEHEAD A.: *Persistent Realtime Building Interior Generation*. In Proc. of ACM Siggraph Symp. on Videogames, ACM Press, pp. 179–186. 2006.
- [KE81] KONING H., EISENBERG J.: *The language of the prairie: Frank Lloyd Wright’s prairie houses*. Environment and Planning B 8 (1981): 295-323. 1981.
- [Kni81] KNIGHT T.: *The forty-one steps*. Environment and Planning B 8 (1981): 97-114. 1981.
- [Mar05] MARTIN J.: *The Algorithmic Beauty of Buildings: Methods for Procedural Building Generation*. Honors Thesis, Trinity University. 2005.
- [MVUG05] MÜLLER P., VEREENOGHE T., ULMER A., GOOL L.: *Automatic Reconstruction of Roman Housing Architecture*. International Workshop on Recording, Modeling and Visualization of Cultural Heritage, Balkema Publishers (Taylor & Francis group), pages 287-297. 2005.
- [MVW*06] MÜLLER P., VEREENOGHE T., WONKA P., PAAP I., GOOL L.: *Procedural 3D Reconstruction of Puuc Buildings in Xkipche*. In Proc. of Eurographics Symp. on Virtual Reality, Archaeology and Cultural Heritage, Eurographics, pp. 139–146. 2006.
- [MWH*06] MÜLLER P., WONKA P., HÄGLER S., ULMER A., GOOL L.: *Procedural modeling of buildings*. ACM Transactions on Graphics 25, 3. 2006.
- [PM01] PARISH Y., MÜLLER P.: *Procedural modeling of cities*. In Proc. of ACM SIGGRAPH, ACM Press / ACM SIGGRAPH, New York, 301-308. 2001.
- [RDG*08] RODRIGUES N., DIONÍSIO M., GONÇALVES A., MAGALHÃES L., MOURA J., CHALMERS A.: *Incorporating Legal Rules on Procedural House Generation*. In Proc. of Spring Conference on Computer Graphics, Bratislava (Slovak Republic). 2008.
- [RMMC07] RODRIGUES N., MAGALHÃES L., MOURA J., CHALMERS A.: *Geração Automática de Estruturas Romanas*. In Proc. of CAAP2007, Leiria (Portugal). 2007.
- [Rom] THE ROMAN EMPIRE
<http://www.roman-empire.net/society/soc-house.html>
- [SG04] SILVA F., GONÇALVES A.: *House of the Skeletons – A Virtual Way*. Proceedings of the XXXII CAA – Computer Applications and Quantitative Methods to Archaeology Conference, Prato, Italy, April 2004.
- [Sti75] STINY G.: *Pictorial and Formal Aspects of Shape and Shape Grammars*. Birkhauser Verlag, Basel. 1975.
- [WMV*08] WATSON B., MÜLLER P., VERYOVKA O., FULLER A., WONKA P., SEXTON C.: *Procedural Urban Modeling in Practice*. IEEE Computer Graphics and Applications, Vol. 28, No. 3, page 18-26. 2008.
- [WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: *Instant architecture*. ACM Transactions on Graphics 22, 3, 669–677. 2003.