The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2006)
M. Ioannides, D. Arnold, F. Niccolucci, K. Mania (Editors)

Procedural 3D Reconstruction of Puuc Buildings in Xkipché

Pascal Müller [1] Tijl Vereenooghe [2] Peter Wonka [3] Iken Paap [4] Luc Van Gool [1,2,5]

[1]Computer Vision Lab, ETH Zürich, Switzerland
[2]Center for Archaeological Sciences / ESAT, K.U. Leuven, Belgium
[3]PRISM - Partnership for Research in Spatial Modeling, Arizona State University, USA
[4]Institut für Griechische und Lateinische Philologie, Romanistik und Altamerikanistik, University of Bonn, Germany
[5]PSI / ESAT, K.U. Leuven, Belgium

Abstract

This paper examines how architectural shape grammars can be used to procedurally generate 3D reconstructions of an archaeological site. The Puuc-style buildings found in Xkipché, Mexico, were used as a test-case. We first introduce the ancient Mayan site of Xkipché and give an overview of the building types as distinguished by the archaeologists, based on excavations and surveys of the building remains at the surface. Secondly, we outline the elements of the building design that are characteristic of the Puuc architecture. For the creation of the actual building geometries, we further determine the shape grammar rules for the different architectural parts. The modeling system can then be used to reconstruct the whole site based on various GIS (Geographical Information Systems) data given as input, such as building footprints, architectural information, and elevation. The results demonstrate that our modeling system is, in contrast to traditional 3D modeling, able to efficiently construct a large number of high quality geometric models at low cost.

Categories and Subject Descriptors (according to ACM CCS): F.4.2 [Mathematical Logic and Formal Languages]: Grammars and Other Rewriting Systems I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling I.6.3 [Simulation and Modeling]: Applications J.6 [Computer-Aided Engineering]: Computer-Aided Design (CAD)

1. Introduction

This paper addresses the 3D reconstruction of archaeological sites using procedural modeling. This approach is particularly suited for archaeological purposes, as a historically accurate reconstruction often depends on fragmentary remains and formal architectural "rules" as derived from similar buildings at other sites. In this paper we take Puuc-style architecture at Xkipché in Mexico as a test case. Puuc is a style of Pre-Columbian Maya architecture and tends to be rule-based. Xkipché has been the subject of archaeological research over a period of more than 15 years. This resulted in the detailed mapping of the building remains above ground and those recovered through excavations in several areas of the city.

Being able to reconstruct large cityscapes is an important issue for archaeology. These reconstructions should not be limited to a few major monuments, but also include other buildings, such as domestic quarters. In this context, traditional 3D modeling tools often require too much manual work and their application is therefore overly expensive for archaeological projects. In contrast, our approach proves to be efficient and fairly simple. Furthermore, our procedural modeling approach allows for the testing of several hypotheses by adjusting some of the parameters. This results in a powerful platform for archaeological discussion and exploration. Our contribution is the detailed formal description and grammatical encoding of Puuc stone buildings in Xkipché. The implementation is based on our *CGA Shape* grammar, recently introduced in [MWH*06].

The paper is structured as follows: In section 2 we describe the archaeological background and related work in procedural modeling. Section 3 explains the general formal design of Puuc buildings in Xkipché. In section 4, we give a short introduction to CGA Shape and present the rules to generate the 3D models. The results we achieved are shown in section 5 and discussed in section 6. Our conclusions can be found in section 7.

2. Background and Related Work

Xkipché is located in the Puuc region, a vaguely defined geographical area which includes the southernmost parts of the state of Yucatan as well as an adjoining section of the northern part of the state of Campeche in Mexico. In this region the Puuc architecture attained its ultimate refinement. The ruins of Xkipché are situated about 9.5 kilometers southwest of Uxmal and the city was inhabited from the 6th until the 10th century AD. The area beyond Uxmal was archaeologically not well-known until the late 1980s. Since 1991 researchers of the University Bonn (Institut für Altamerikanistik und Ethnologie) have been carrying out archaeological excavations and a detailed study of the exposed building remains [Pre99]. They manually created a 3D reconstruction of the so-called palace building using CAD software. But these models were not very detailed and none of the surrounding buildings have been created.

Puuc is a subtype of Mayan architecture which is characterized by its veneer-over-concrete construction technique resulting in geometric and repetitive façade structures. Well preserved examples of Puuc-style buildings can be found at Uxmal or Kiuic (see figure 1). In contrast to classical Greek and Roman architecture, the formalization of Puuc or Mayan architecture has received little attention from scholars studying the architectural remains. As a result, the current research could not build upon a formal definition of architectural rules and guidelines (with the exception of some short paragraphs about architecture in [And75, Pol80]). However, comparing Mayan and classical architecture, Andrews explicitly states in [And75] that "The smallest temple, including the platform which supports it, can be observed to be composed of a number of discrete parts, or elements, whose ordering appears to be dependent on a set of rules as explicit as those governing the Roman and Greek orders." This quote supports the feasibility of reconstructing Mayan cityscapes by means of procedural modeling.

Figure 1: *Photograph of a well preserved Puuc-style building in Kiuic, Mexico.*

Procedural modeling techniques for urban environments make use of L-systems [PM01, PL91], shape grammars [Sti75, Sti80, WWSR03, MWH*06], and stack-based languages [Hav05]. In many cases, the procedural modeling techniques are adapted to specific problems, such as modeling of Siza's mass housing [Dua02], medieval houses [BBJ*01], castles [GBHF05], and Chinese temples [LWH*05]. Our work is based on the *CityEngine* software suite [PM01, MWH*06], a comprehensive modeling system that allows to design mass models, create architectural details, and place vegetation. In [MVUG05], the grammar-based reconstruction approach using the CityEngine has been successfully introduced to the archaeological community. Therein, the general CityEngine pipeline for a stochastical creation of buildings has been described, but no detail has been given concerning the architectural rules (for an accurate reconstruction of selected buildings).

3. Puuc Architecture in Xkipché

3.1. Building Types

In Xkipché the archaeological research has identified eight main concentrations of buildings (see groups A-H in figure 2), of which group B probably represents the ceremonial center, whereas groups A, C and D have a residential character. The largest and most important building of the site is located in the center of group A; it is referred to as palace building.

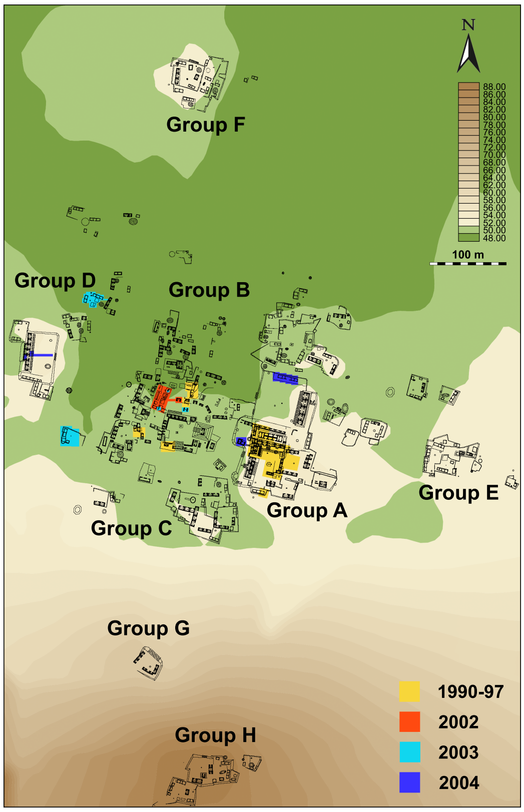

Figure 2: *Overview map of Xkipché with building footprints and elevation. Additionally, the different excavation campaigns over the years are illustrated.*

The archaeologists recognized 18 different building types (illustrated in figure 3). Several of these types, however, only represent one or a part of a building. One of the advantages of Xkipché as a test-case is the level of preservation of the buildings: a lot of the ancient stone buildings are still (partially) standing. Moreover, the shapes of the stones from the respective elements are often very specific. In most cases the archaeologists can easily attribute the architectural fragments, retrieved near collapsed buildings, to their original place in the façade. However, details of the decoration - like e.g. the number and arrangement of colonettes - remain uncertain in many cases. Although the number of stones found can contribute to a more precise reconstruction of the original buildings, one can not neglect the fact that decorated stones of older buildings have frequently been taken away and were reused by the inhabitants themselves.

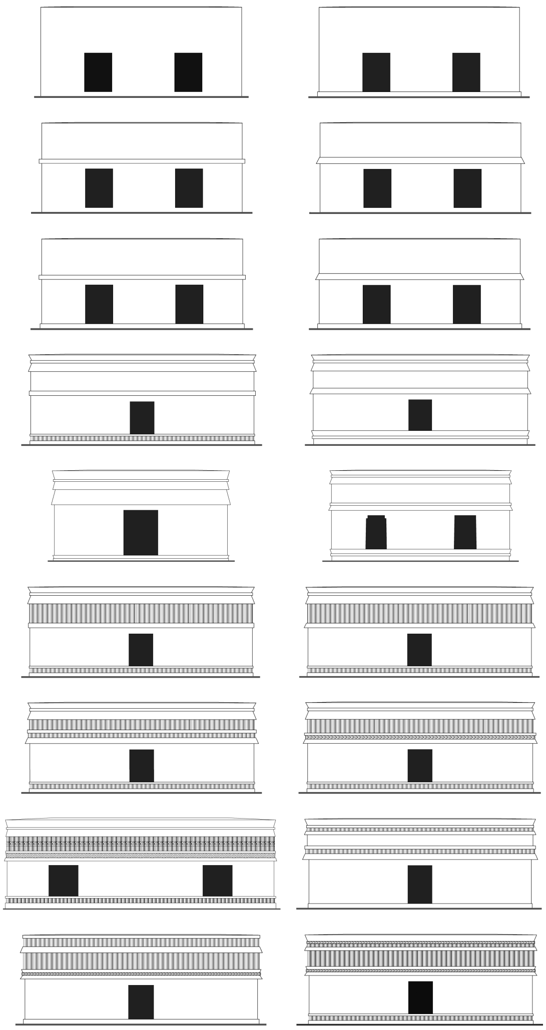

Figure 3: *The 18 building types found in Xkipché. These appearances have been archaeologically reconstructed based on preserved building parts and the remains on the surface.*

3.2. Building Design

Puuc houses were often built on a platform sub-structure, a cut and stucco stone exterior filled with densely packed gravel. The buildings have rubble-filled concrete walls faced by a thin veneer of dressed stone. The dominant characteristics of Puuc-style architecture in Xkipché are a plain lower wall (with openings) above a rather elaborate base molding, and on the upper part of the façade a large medial molding, a frieze (with or without decoration) and a usually high cornice molding [And75, Pol80]. This general building design is illustrated in figure 4 (top).

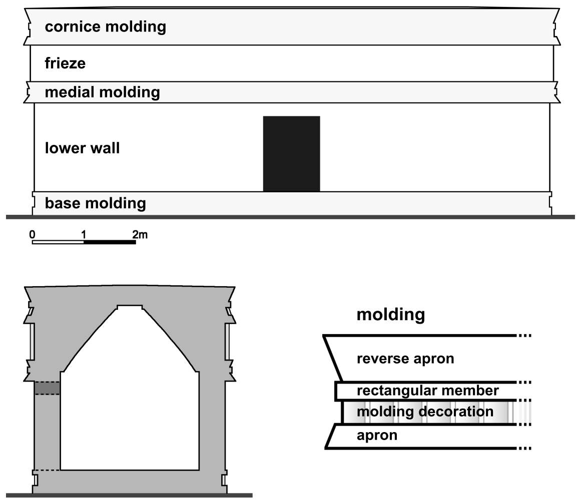

Figure 4: *Top: formal design of a generic stone house in Xkipché. Bottom left: the same building in profile. Bottom right: close up of a four-member molding.*

Apart from the doors, the only wall openings were small rectangular ventilators, often just below the medial molding. Wood was only exceptionally employed as door lintels or unstructurally in the corbelled masonry vaulting system. The average width of the door openings is about 100-120 cm. The door jambs had a strong tendency to incline inwards slightly [Pol80]. Puuc doors were sometimes framed by columns with simple, rectangular capitals [Car86], complemented by small corbels at the top of each jamb.

The frieze decoration (if any) consists usually of colonettes (serrated cylinders) or mosaic elements (lattice work, T-shapes, stylized serpent heads, stepped frets...) of limestone masonry, creating geometric repetition and symmetry. Other typical elements of Puuc façades were long-nosed masks, often supposed to represent the rain god Chac. This more elaborate decoration was frequently found over doorways and at the corner of buildings.

In [Pol80], Pollock gives an overview of moldings, which can have several appearances. Three-member moldings seem to be most common in Puuc architecture, usually consisting of (1) an apron member, a middle rectangular member and an upper reverse apron or (2) an apron member, a decorated member and a rectangular member on top. In Xkipché, some of the cornice moldings consist of four members (illustrated in figure 4, bottom right). Other moldings only have one or two members.

4. Grammar-based Reconstruction

4.1. CGA Shape

CGA Shape is a grammar suitable for architectural design. In this subsection we will give a short introduction to CGA Shape necessary to encode the designs in the previous section. A more comprehensive description of CGA Shape is given in [MWH*06].

Shape: The grammar works with a configuration of shapes: a shape consists of a symbol (string), geometry (geometric attributes) and numeric attributes. Shapes are identified by their symbols which is either a terminal symbol $\in \Sigma$, or a non-terminal symbol $\in V$. The corresponding shapes are called terminal shapes and non-terminal shapes. The most important geometric attributes are the position P, three orthogonal vectors X, Y, and Z, describing a coordinate system, and a size vector S. These attributes define an oriented bounding box in space called *scope* (see figure 5).

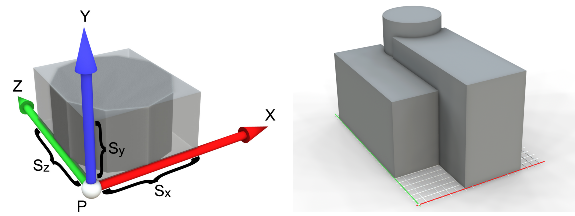

Figure 5: *Left: The scope of a shape. The point P, together with the three axis X, Y, and Z and a size S define a box in space that contains the shape. Right: A simple building mass model composed of three shape primitives.*

Production process: A configuration is a finite set of basic shapes. The production process can start with an arbitrary configuration of shapes A, called the axiom, and proceeds as follows: (1) Select an active shape with symbol B in the set (2) choose a production rule with B on the left hand side to compute a successor for B, a new set of shapes $BNEW$ (3) mark the shape B as inactive and add the shapes $BNEW$ to the configuration and continue with step (1). When the configuration contains no more non-terminals, the production process terminates.

Notation: The CGA Shape production rules are defined in the following form:

id: predecessor : condition $\rightsquigarrow$ successor

where *id* is a unique identifier for the rule, *predecessor* $\in V$ is a symbol identifying a shape that is to be replaced with *successor*, and *condition* is a guard (logical expression) that has to evaluate to true in order for the rule to be applied. For example, the rule

1: fac(h) : h > 9 $\rightsquigarrow$ floor(h/3) floor(h/3) floor(h/3)

replaces the shape fac with three shapes $floor$, if the parameter h is greater than 9. To specify the successor shapes we use different forms of rules explained in the remainder of this section.

Shape operations: Similar to L-systems we use general rules to modify shapes: $T(t_x,t_y,t_z)$ is a translation vector that is added to the scope position P, $Rx(angle)$, $Ry(angle)$, and $Rz(angle)$ rotate the respective axis of the coordinate system, and $S(s_x,s_y,s_z)$ sets the size of the scope. We use [and] to push and pop the current scope on a stack. Any non-terminal symbol $\in V$ in the rule will be created with the current scope. Similarly, the command $I(objId)$ adds an instance of a geometric primitive with identifier $objId$. Typical objects include a cube, a quad, and a cylinder, but any three-dimensional model can be used. The example below illustrates the design of the mass model depicted in figure 5 right:

1: A $\rightsquigarrow$ [T(0,0,6) S(8,10,18) I("cube")]
T(6,0,0) S(7,13,18) I("cube")
T(0,0,16) S(8,15,8) I("cylinder")

Basic split rule: The basic split rule splits the current scope along one axis. For example, consider the rule to split the façade of figure 6 left into four floors and one ledge:

1: fac $\rightsquigarrow$
Subdiv("Y",3.5,0.3,3,3,3){ floor | ledge | floor | floor | floor }

where the first parameter describes the split axis ("X", "Y", or "Z") and the remaining parameters describe the split sizes. Between the delimiter { and } a list of shapes is given, separated by |.

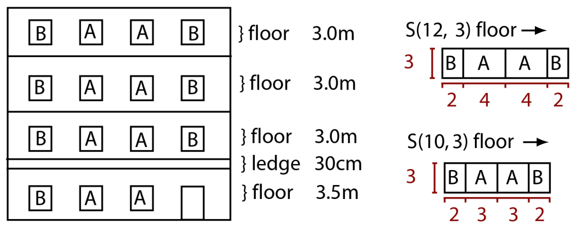

Figure 6: *Left: A basic façade design. Right: A simple split that could be used for the top three floors.*

Scaling of rules: From the previous example we can see the first challenge. The split is dimensioned to work well with a scope of size $y = 12.8$, but for other scopes the rule has to be scaled. From our experience not all architectural parts scale equally well, and it is important to have the possibility to distinguish between absolute values (values that do not scale) and relative values (values that do scale). Values are considered absolute by default and we will use the letter *r* to denote relative values, e.g.

1: floor ⤳ Subdiv("X",2,1r,1r,2){ B | A | A | B }

where relative values r_i are substituted as $r_i * (Scope.sx - \sum abs_i)/\sum r_i$ ($Scope.sx$ represents the size of the x-length of the current scope). Figure 6 right illustrates the application of the rule above on two different sized floors (with x-length 12 and 10).

Repeat split: To allow for larger scale changes in the split rules, we often want to tile a specified element. For example

1: floor ⤳ Repeat("X",2){ B }

tiles the floor into as many elements of type *B* along the x-axis of the scope as there is space. The number of repetitions is computed as $repetitions = \lceil Scope.sx/2 \rceil$ and we adjust the actual size of the element accordingly.

Component split: Up until this point all shapes (scopes) have been three-dimensional. The following command allows to split into shapes of lesser dimensions:

1: a ⤳ Comp(type, param){ A | B | ... | Z }

where *type* identifies the type of the component split with associated parameters *param* (if any). For example we write $Comp(\text{"}faces\text{"})\{A\}$ to create a shape with symbol *A* for each face of the original three-dimensional shape. Similarly we use $Comp(\text{"}edges\text{"})\{B\}$ and $Comp(\text{"}vertices\text{"})\{C\}$ to split into edges and vertices respectively. To access only selected components we use commands such as $Comp(\text{"}edge\text{"},3)\{A\}$ to create a shape *A* aligned with the third edge of the model or $Comp(\text{"}sidefaces\text{"})\{B\}$ to access all the side faces of e.g. a cube or polygonal cylinder.

4.2. Reconstruction Rules

In the following we explain the rules that we created to model Puuc buildings as they have been introduced in section 3. The rules are slightly simplified for space reasons, however, they are in principal sufficient to capture most of the variety of the Puuc architecture. The rules use *control parameters* that can be read from the GIS database. As geometric parameter we use the building footprint. The other control parameters are scalar values typeset in *italics*. The first rule takes the GIS footprint and extrudes it to a volumetric shape. Rule #2 creates façade shapes for each building face. As seen in figure 4 a façade can be broken down into several elements (rule #3). Note that we shorten height, width, depth or angle to *h*, *w*, *d* or *a* to save space in the rule description.

1: footprint ⤳ S(1r,*building_h*,1r) facades
2: facades ⤳ Comp("sidefaces"){ facade }
3: facade ⤳ Subdiv("Y",*base_molding*,1r, *medial_molding_h*, *frieze_h*,*cornice_molding_h*)
{ base_molding | lower_facade | medial_molding | frieze | cornice_molding }

This example shows a building with one door. Buildings with multiple doors can be generated by variations of rule #4. The buildings have only one façade with a door. All the other façades consist of a plain wall. The rule parameter *projection* in rule #6 is used to vary the thickness of the wall (by extruding the wall with distance *projection*, used for example in rule #16). The doorframe consists of three elements: the lintel on top and two jamb elements. Note that the lintel is arranged directly below the medial molding and that the jambs can be a little bit sloped (defined by angle given with the control parameter *jamb_a*).

4: lower_facade : Shape.visible("front") ⤳
Subdiv("X",1r,*door_w*+2**door_frame_w*,1r)
{ wall(0) | door | wall(0) }
5: lower_facade ⤳ wall(0)
6: wall(projection) ⤳ T(0,0,-projection)
S(1r,1r,*wall_d*+projection) I("wall.obj")
7: door ⤳
Subdiv("Y",*door_h*,1r){ Subdiv("X",1r,*door_w*,1r)
{ jamb(-*jamb_a*) | null | jamb(*jamb_a*) } | lintel }
8: lintel ⤳ S(1r,1r,*wall_d*) I("lintel.obj")
9: jamb(angle) ⤳ R(0,0,angle) S(1r,1r,*wall_d*) I("jamb.obj")

The moldings come in many varieties, but are all composed of the same elements. Figure 7 shows how the maximal four members of a molding are put together. The most important geometrical parameters are the heights of the elements and the projection parameters (to control how far elements are extruded from the plane that contains the wall).

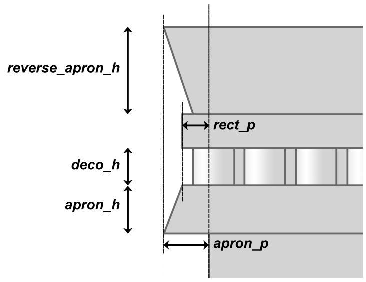

Figure 7: *The five parameters defining the appearance of a molding (plus the molding height). By setting the height of a member to zero the element can be switched off.*

All three different moldings (*base_molding*, *medial_molding* and *cornice_molding* as introduced in rule #3) are generated using the same procedure (initiated via rule #13). The only difference is the assignment of the different control parameters to the corresponding rule parameters:

10: cornice_molding ⤳ molding(*cornice_apron_h*, *cornice_deco_h*,*cornice_reverse_apron_h*, *cornice_apron_p*,*cornice_rect_p*)

This rule induces the creation of the cornice molding. The rules for the medial and base molding are defined accordingly, i.e. with the corresponding control parameters. In the following, we show the rule for a decorated molding. We also developed a similar rule to create moldings without decoration element (triggered via condition and then using other projection proportions).

13: molding(apron_h,deco_h,revap_h,apron_p,rect_p) $\rightsquigarrow$
Subdiv("Y",apron_h,1r,revap_h)
{ apron(rect_p,apron_p) | deco(rect_p*0.8-apron_p*0.2) |
rect(rect_p) | revap(rect_p*0.8-apron_p*0.2,apron_p) }

The four molding members are encoded as follows. All members have in common that they consist of a projected wall (described in rule #6): both aprons consist of a projected wall and a sloped element in front of the wall, the decoration member consists of colonettes in front of the projected wall, and the rectangular member consists of a projected wall only. Note that we call the projection parameters p, $p1$ and $p2$.

14: apron(p1,p2) $\rightsquigarrow$ wall(p1)
T(0,0,-p2) S(1r,1r,p2-p1) I("apron.obj")
15: deco(p) $\rightsquigarrow$ wall(p-$molding\_colonette\_diameter$/2)
T(0,0,-p) colonettes($molding\_colonette\_diameter$)
16: rect(p) $\rightsquigarrow$ wall(p)
17: revap(p1,p2) $\rightsquigarrow$ wall(p1)
T(0,0,-p2) S(1r,1r,p2-p1) I("reverse_apron.obj")

The colonettes are created using cylinders. The rule just creates one empty element at the end so that the cylinders at the corner are not created twice. The modeling of building corners is a common challenge to most procedural building models.

18: colonettes(d) $\rightsquigarrow$ Subdiv("X",1r,d*1.2)
{ Repeat("X",d*1.2){ colonette(d) } | ε }
19: colonette(d) $\rightsquigarrow$ S(d,1r,d) I("cylinder.obj")

In the following we describe two of several different frieze types: one without and one with decoration (colonettes). Other frieze rules include the positioning of the masks (edge mask and front mask) which are separately reconstructed ornaments of high geometrical complexity.

20: frieze : $frieze\_decoration$ == "none" $\rightsquigarrow$ wall($frieze\_p$)
21: frieze : $frieze\_decoration$ == "colonettes" $\rightsquigarrow$
wall($frieze\_p$) colonettes($frieze\_colonette\_diameter$)

5. Results

By using the rule set presented above, we are able to generate each of the building types listed in figure 4 in about 5 to 10 minutes (by simple modification of the control parameters). Additionally, we use rules for specifying materials and textures that are not shown in the text. Please note that we created all types according to archaeological data and we do not use random variations for the reconstruction. See figure 8 for selected buildings, with interesting height and molding combinations. These images are rendered in OpenGL and are screenshots from the interactive previewing system of the CityEngine. High quality renderings can be created with offline rendering.

Additionally, we extended the rules to generate other more complex buildings (few of them in Xkipché). Figure 9 shows a closeup of ornamented colonettes including mosaics and figure 10 pictures a whole building. These images have been created with Pixar's RenderMan. Ambient occlusion has been used to simulate the exterior lighting.

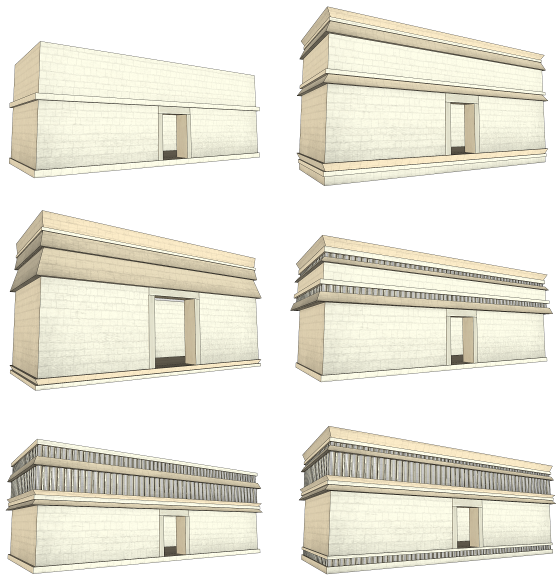

Figure 8: *This image shows various buildings that have been created in minutes by using the rule set described in the paper. Simple modifications of its control parameters lead to the different building appearances.*

Figure 9: *The rules for colonettes and frieze have been extended to be able to reconstruct also more complex building appearances.*

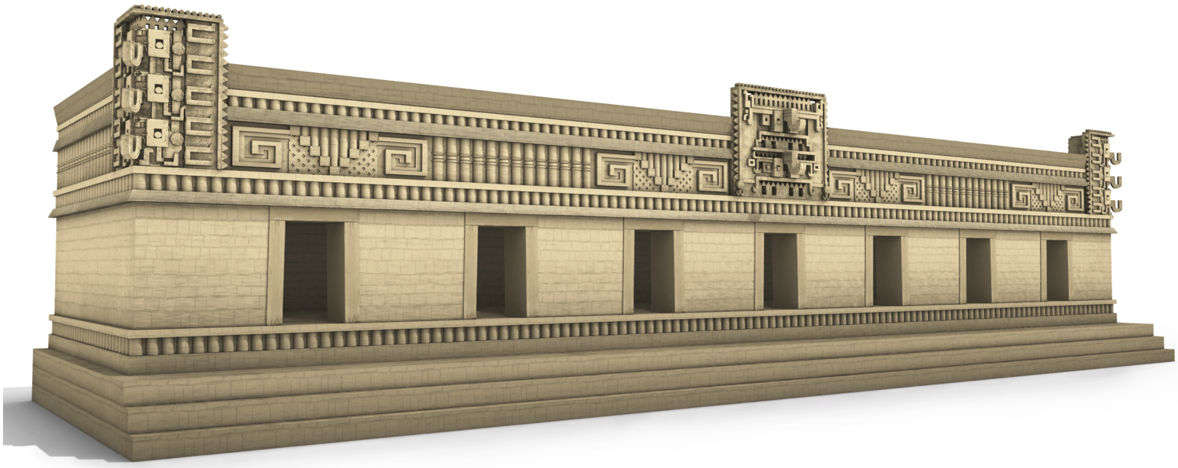

Figure 10: *Detailed reconstruction of one of the few highly ornamented buildings in Xkipché. The whole building has been generated procedurally by using the CityEngine, except the complex mask ornaments which have been created with traditional mesh modeling software. The image has been rendered in Pixar's RenderMan.*

6. Discussion

In this section, we want to identify contributions and open problems that are of interest for future research.

Procedural modeling and archaeology: Archaeology is an interesting application area for procedural modeling because information is only available in fragments. Therefore, the virtual reconstruction can not only be based on scanning, but needs to rely on human synthesis of data from multiple and heterogeneous sources. We believe that procedural modeling rules are an interesting and useful form of knowledge representation for such a synthesis. First, these rules can be used to create reconstructions that form the basis of archaeological discussion and presentation. Second, the rules themselves are formal and comprehensive. This notation has some advantages over a mere description in words, illustrated examples and annotated plans. The grammar framework ensures that essential parameters are not left unspecified and all information for a reconstruction is available.

Reconstruction detail: While we were able to create fairly detailed reconstructions of Xkipché buildings, modeling is generally an open ended problem and there are many opportunities for extensions: (1) It would be possible to integrate more GIS data, such as the exact position of the doors - now we only estimate door positions through the grammar. (2) Moldings could be made more accurate by using more parameters. (3) Additional (molding) decoration styles could be implemented - we discussed only the colonettes in detail. (4) Originally, the building surfaces were covered with plaster and painted with mineral and organic pigments [Car86], but since their exact appearance is still under archaeological debate, we did not include colorful textures such as paintings in the current model.

Efficiency and useability: The main part of the reconstruction work was reading and ordering the archaeological information and references (1 week). The model presented in this paper has then been created in three days: one day of architectural analysis, one day of modeling the elements (mainly the frieze-decoration was taking time) and one day for the actual implementation and encoding of the rules. Any building in Xkipché can now be reconstructed in detail in 2 minutes, if the user specifies the 20 - 30 parameters describing the building. We estimate that a professional CAD modeler will need 2-3h for the same task, but he would also need the models of the individual elements. Another big advantage of our approach is that after the initial model is created, archaeological researchers can create new models without any CAD-knowledge using a high-level user interface to specify parameters.

Future work: We are planning to investigate approaches to reconstruct the traditional houses (made of organic materials) and other aspects of urban environments, such as walkways and the vegetation in general. Therefore, we are working on a tighter GIS integration of the CityEngine by developing a practical GIS format which allows archaeologists to define the needed attributes directly in a common editor like ESRI's ArcGIS system. We expect that such an integration will enormously enhance the usefulness of our approach and make it applicable to all kinds of reconstruction scenarios.

7. Conclusion

We presented a method to procedurally create 3D reconstructions of stone houses in Xkipché. The reconstruction is based on archaeological research and makes use of shape grammars to encode the architectural design of buildings of the Puuc architecture. We demonstrated that this approach is a promising tool for archaeology, as it allows for the precise encoding of archaeological knowledge, simple and fast parameter-based modeling, and accurate 3D reconstructions of architectural content.

Acknowledgments

The authors thank Simon Haegler for helping with the renderings and the anonymous reviewers for their constructive comments on improving this paper. This research is supported in part by EC IST Network of Excellence EPOCH, EC IST Project CyberWalk, CHIRON Marie Curie EST Research Programme, NSF contract IIS-0612269 and NGA grant HM1582-05-1-2004. The excavation at Xkipché has been funded by the German Research Foundation (DFG): Archäologisches Projekt Xkipché (1991-1997) and Epiklassikum in Nord-Yucatan (2002-2006).

References

[And75] ANDREWS G. F.: *Maya Cities. Placemaking and Urbanization*. University of Oklahoma Press, 1975.

[BBJ*01] BIRCH P., BROWNE S., JENNINGS V., DAY A., ARNOLD D.: Rapid Procedural-Modelling of Architectural Structures. In *Virtual Reality, Archaeology and Cultural Heritage (VAST)* (2001), pp. 187–196.

[Car86] CARVER N. F.: *Silent Cities of Mexico and the Maya*. Documan Press Ltd, 1986.

[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.

[GBHF05] GERTH B., BERNDT R., HAVEMANN S., FELLNER D. W.: 3D Modeling for Non-Expert Users with the Castle Construction Kit v0.5. In *Virtual Reality, Archaeology and Cultural Heritage (VAST)* (2005), Mudge M., Ryan N., Scopignooutit R., (Eds.).

[Hav05] HAVEMANN S.: *Generative Mesh Modeling*. PhD thesis, TU Braunschweig, 2005.

[LWH*05] LIU H., WANG Q., HUA W., ZHOU D., BAO H.: Building Chinese Ancient Architectures in Seconds. In *International Conference on Computational Science* (2005), pp. 248–255.

[MVUG05] MÜLLER P., VEREENOOGHE T., ULMER A., GOOL L. V.: Automatic Reconstruction of Roman Housing Architecture. In *International Workshop on Recording, Modeling and Visualization of Cultural Heritage* (2005), Balkema Publishers (Taylor & Francis group), pp. 287–297.

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural Modeling of Buildings. *Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics 25*, 3 (2006), 614–623.

[PL91] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants*. Springer Verlag, 1991.

[PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *Proceedings of ACM SIGGRAPH 2001* (2001), Fiume E., (Ed.), ACM Press, pp. 301–308.

[Pol80] POLLOCK H. E. D.: *The Puuc. An Architectural Survey of the Hill Country of Yucatan and Northern Campeche, Mexico*. Peabody Museum of Archaeology and Ethnology Harvard University, 1980.

[Pre99] PREM H. J.: Geschichte eines Mayapalastes - Ausgrabungen in Xkipché, Yucatán. *Antike Welt 30*, 6 (1999), 545–554.

[Sti75] STINY G.: *Pictorial and Formal Aspects of Shape and Shape Grammars*. Birkhauser Verlag, Basel, 1975.

[Sti80] STINY G.: Introduction to shape and shape grammars. *Environment and Planning B 7* (1980), 343–361.

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *Proceedings of ACM SIGGRAPH 2003 / ACM Transactions on Graphics 22*, 3 (2003), 669–677.