EURASIP Journal on Image and Video Processing
a SpringerOpen Journal
IMPACT FACTOR 0.74

This article is part of the series Image and Video Processing for Cultural Heritage.

Research Article Open Access

Procedural Modeling for Digital Cultural Heritage

Simon Haegler[1]*, **Pascal Müller**[2] and **Luc Van Gool**[1,3]

* Corresponding author: Simon Haegler shaegler@vision.ee.ethz.ch

1 Computer Vision Lab, Department of Information Technology and Electrical Engineering (D-ITET), ETH Zurich, CH-8092 Zurich, Switzerland

2 Procedural Inc., Technoparkstrasse 1, spin-off of the Computer Vision Lab, ETH Zurich, 8005 Zurich, Switzerland

3 ESAT - PSI, Centre for the Processing of Speech & Images, KU Leuven, B-3001 Heverlee, Belgium

For all author emails, please log on.

EURASIP Journal on Image and Video Processing 2009, **2009**:852392 doi:10.1155/2009/852392

The electronic version of this article is the complete one and can be found online at: http://jivp.eurasipjournals.com/content/2009/1/852392

Received: 22 February 2009
Revisions received: 5 August 2009
Accepted: 16 November 2009
Published: 21 December 2009

The rapid development of computer graphics and imaging provides the modern archeologist with several tools to realistically model and visualize archeological sites in 3D. This, however, creates a tension between veridical and realistic modeling. Visually compelling models may lead people to falsely believe that there exists very precise knowledge about the past appearance of a site. In order to make the underlying uncertainty visible, it has been proposed to encode this uncertainty with different levels of transparency in the rendering, or of decoloration of the textures. We argue that procedural modeling technology based on shape grammars provides an interesting alternative to such measures, as they tend to spoil the experience for the observer. Both its efficiency and compactness make procedural modeling a tool to produce multiple models, which together sample the space of possibilities. Variations between the different models express levels of uncertainty implicitly, while letting each individual model keeping its realistic appearance. The underlying, structural description makes the uncertainty explicit. Additionally, procedural modeling also yields the flexibility to incorporate changes as knowledge of an archeological site gets refined. Annotations explaining modeling decisions can be included. We demonstrate our procedural modeling implementation with several recent examples.

1. Introduction

In the same way as the general public is used to more and more realistic visualizations of virtual worlds, the demand for such 3D models of archeological sites is growing. These models are not only used for edutainment and site marketing, they also provide a basis for dissemination and scientific discussion about reconstruction hypotheses.

Nonetheless, realistically looking models are also a cause of concern to cultural heritage experts. A caveat often raised is that such models may convey a false impression of certainty about the past, whereas more often than not, one can only guess how large parts of buildings or sites may have looked like. This raises the question of how to deal with such uncertainty, that is, how to resolve the tension between veridical and realistic modeling. On the one hand, one would want to create an enticing experience for the public. On the other hand, scientific rigor commands that one must very clearly communicate such uncertainty to the public. In this paper, we argue that a way out of this conundrum is to create not one model, but multiple models. Variations among multiple, fully realistic models are a natural way of expressing uncertainty while keeping observers intrigued. Obviously, such an approach is only feasible if the creation of these models is efficient.

There are a lot of technologies and commercial tools available today to create realistic 3D models of cultural heritage sites, too many to even begin giving an overview. Unfortunately, they share the high level of manual work involved. Building even a single model therefore is too expensive for the vast majority of sites. Procedural modeling would seem to offer a good alternative, however. This paper discusses how this approach to modeling can resolve some of the main issues (e.g., handling uncertainty) with the modeling of architectural structures in particular. It helps to address the inflexible nature of static geometrical site models and avoids repetitive and labor-intensive manual modeling. It also enables site models to be updated more often as knowledge increases and to express the remaining uncertainty effectively.

The structure of this paper is as follows. Section 2 discusses the veridicality-realism tension in more detail. It also describes some previous methods to express uncertainty in 3D models. Next, we give a short introduction to procedural modeling in Section 3. That section also shows some previous work based on the CityEngine, a tool for the procedural modeling of buildings and sites. Section 4 discusses and demonstrates the advantages of procedural modeling for expressing uncertainty while still delivering enticing realism. Finally, Section 5 discusses the pro's and con's of procedural modeling in the context of expressing uncertainty in cultural heritage and sketches some future research.

2. The Problem of Reconstruction Uncertainty

Virtual 3D models of monuments and sites that have largely disappeared can serve many purposes. They allow experts to visualize and investigate features of a site, which may otherwise be difficult to appreciate. Many researchers do indeed see them as more than modern illustrations: they are dynamic visual research models. Moreover, they contribute enormously to the dissemination of archeological research and its effective presentation to the public. Frischer et al. [1] give an overview of the history of Cultural Virtual Reality (CVR) and discuss the need for aesthetic, scientific and technical standards in this field. The level of detail and photo-realism at which one ought to try and produce such reconstructions is disputed. Some warn that the more compelling a reconstruction is, the more the general public may take the correctness of every detail for granted, even if part of the reconstruction is based on not much more than a dedicated guess or one among several hypotheses.

We recognize the relevance of these caveats. It should for instance remain possible to visualize the levels of uncertainty for all parts of a model. Rationales behind particular completions and choices should be documented, preferably also as annotations to the model, allowing scholars to have direct and easy access to them. This is what the *The London Charter* (LC in short) [2] demands in its principles (e.g., in principles 4 and 6). The LC lists a number of fundamental principles for 3D visualization methods in cultural heritage. The introductory article [3] about the charter gives an overview of critical literature about photo-realism and transparency in cultural virtual heritage. These issues have also been raised by Forte [4], a pioneer in 3D modeling of cultural heritage: "Noticeable gaps are represented by the fact that the models are not "transparent" in respect to the initial information (what were the initial data?) and by the use of peremptory single reconstruction without offering alternatives". On the other hand, 3D computer models should not be judged too harshly either. Quoting [5]: "VR is the modern version of the artist that gave a "possible" reconstruction using water-colors". One only has to think of the Halicarnassos mausoleum to have a vivid example of how different such hand-drawn reconstructions could be [6]. Omitting uncertain parts would basically only leave one with thin air.

Moreover, we see at least as big a danger in oversimplified models, and this is what the alternative for realism has often turned out to be in practice [7]. These can be misleading in at least two ways. Copy-and-paste strategies have been popular in the production of such models. Given the uncertainty about the precise shape of a component—for example, a pillar—why not represent it by an example placeholder shape and simply repeat it. Such repetitions may create unrealistic, exact regularity in shapes or patterns. This may convey a false impression of technological sophistication. Also, such models tend to be produced by starting from a library of predefined, pure geometric primitives. Perfectly planar walls, precisely cylindrical arches and pillars, repetitions of identical tiles or decoration, and so forth. tend to be a far cry from actual variations found in handcrafted elements. Such simplifications do not necessarily let observers infer uncertainty.

Furthermore, omissions—another type of simplification—could have the effect in that they often fail to do justice to the true level of decoration of a structure or to the intentions of its creators. One can leave out colors on Greek buildings, for instance, thereby perpetuating one of the most persistent misconceptions about their original appearance. Even if there may be uncertainty about which color ought to go where, making occasional mistakes in the coloration may well be the lesser evil. Similarly, even if one is not absolutely certain about the ornamentation found in certain parts of a building, it may be better to make a dedicated guess at its original state than to simply leave it out.

Hence, leaving out any structures one is not absolutely sure about, combining basic geometric primitives, or adopting copy-and-paste methods, all entail definite dangers just the same. Moreover, such models will almost certainly fail to generate interest from the public. Especially younger viewers need visually compelling images to retain their interest, given their constant exposure to impressive graphical imagery in games and movies [8].

Alternatively, what we propose in this paper is an efficient way of modeling even possibly large sites. The increase in efficiency compared to traditional modeling is such that one can leave behind the idea of building a single model with indications of uncertainty levels. Rather than using coloration, levels of transparency or non-photorealistic rendering (NPR) [9, 10] to indicate that one is not quite that certain about a part of a model, one can produce several, realistic models. Each of these models can be truly enjoyed by the public (coloration or transparency invariably ruin the experience). Thinking about uncertainty automatically leads to the idea of probability distributions. One way of representing those is to sample them, with a sufficiently large number of samples. Rather than producing the model, one could therefore in analogy create a representative multitude of models. Each model is a sample for the underlying, more abstract and semantic description, which also contains explicit indications of uncertainty and expert annotations. It stands to reason to let these models—which are all instantiations from the family of shapes described by the procedural description—share aspects that are certain, while they differ in those parts where some guess-work is needed. Presenting such models in sequence gives the public a good impression of how it may have been, while still clearly bringing home the message of not being certain across the board. The *procedural modeling* strategy makes it possible to create multiple such models from single structures to entire sites (e.g., of Pompeii or imperial Rome, as we will show in the next sections).

3. Procedural Modeling

We start this section by showing the principal differences between manual and procedural 3D modeling. An overview of the relevant work in the field of procedural modeling is given and we introduce our approach for the procedural generation of multiple hypotheses of archeological sites: *shape grammar based procedural modeling*.

3.1. Reading the Architect's Mind

With traditional modeling, the artist works almost exclusively at the geometrical and visual level of the model: vertices, lines, surfaces, materials and colors are modified part-by-part (and building-by-building) to optimally match the drawings, blueprints and possibly scans provided by archeologists and surveyors. The artist uses techniques like conventional mesh modeling, sculpting and image-based modeling to create the geometry from the template data. In other words, the modeler reconstructs the visual appearance of the site step by step. The structural logic of the model does not necessarily match the architectural design of the site, as only the visual end effects should be similar. It follows that there are multiple situations during the modeling process where uncertainties are explicitly (or worse, implicitly) glossed over.

In contrast, the procedural modeling approach reduces the geometrical aspect during modeling in favor of the *structural and semantic* information (including uncertainty). In a procedural site description the spatial and semantic relationships between objects (also called *shapes* in grammar-based procedural modeling) are more important than the absolute geometrical coordinates and appearance attributes. If the architectural relationships between elements of the site are correctly captured into procedural descriptions (also called *rule sets* in grammar-based modeling), valid spatial coordinates and visual attributes within the bounds of uncertainty follow automatically. Ultimately and ideally, procedural descriptions are a translation of the architect's mind into a machine readable format. Examples for these claims are given in Section 3.3.

In this shift of focus from the geometrical to the semantic/structural level we see a number of advantages. First, and most importantly, the text-based procedural models inherently provide a container for the explicit formulation of uncertainty, for annotations and for the tracking of changes (cf. LC, principle 4). Second, the models are more coherent and the risk of errors is reduced. For example, model parts which have architecturally the same (hypothetical) meaning will also receive the same geometrical and visual attributes, that is, dependency relationships as defined in the LC (principle 4.10) are explicitly stated. Third, the model is greatly reduced in storage size as duplicate elements can be shared. Uncertainty bounds still generate variations, but only at the time of instantiation (3D model creation). Fourth, the model structure and uncertainties remain human readable in absence of a visualization tool (cf. LC principle 5). Fifth, if architectural rules are present in the literature (e.g., Palladio [11] and Vitruvius [12]), they usually can be easily translated into modeling rules.

For archeological reconstructions, the actual sources of data are the same for traditional and procedural modeling [13]. A nonexhaustive list might include (1) archeological drawings and sketches, (2) text-based literature, (3) photographs, (4) raster data (e.g., digital elevation model), (5) vector data (e.g., geodesic measurements), (6) surveys (e.g., by laser scanning, photogrammetry, or computer vision methods).

3.2. Related Work in Procedural Modeling

Research in procedural modeling has been going on for several decades. *Architectural* procedural modeling can draw on a number of *production systems* such as Semi-Thue processes [14], Chomsky grammars [15], graph grammars [16], shape grammars [17, 18], attributed grammars [19], L-systems [20], or set grammars [21]. With hard-coded modeling rules written in a programming language like C++ on one side of the spectrum and more constrained grammar-based systems on the other, all these methods present the user with different expressiveness and efficiency.

Shape grammars as introduced by Stiny were successfully used for the construction and analysis of architectural designs [22–26]. Classical shape grammars have not been designed to operate automatically, they are missing a control mechanism that selects rules and transformations in each production step. We and others therefore have tried to make the shape grammar concept more applicable to architectural modeling in practice [27, 28]. A result is available through the "CityEngine" software package [29]. Recently there has been work to provide the users of shape and design grammars with visual editing tools and solve some of the complexity issues arising from local changes [30].

A complementary technology to shape grammars is the concept of the *Generative Modeling Language*, or GML for short, introduced by Berndt et al. [31] where an architectural or archeological asset is described by a set of low-level operations on (control) mesh components. GML shares with shape grammars the possibility to alter shapes through the simple interpretation of parameter settings. A similar method to our grammar-based approach is presented in [32], but their approach still involves a lot of manual work to re-create a whole urban area.

3.3. Shape-Grammar-Based Modeling with the CityEngine

The CityEngine (Figure 1) is a tool that allows for efficient modeling of 3D scenes at large scale and in arbitrary detail while retaining the flexibility to adapt the model at hand for future changes as knowledge about a site gets refined (for example, as an excavation campaign progresses). Compared to traditional 3D modeling tools, the price to pay for this greater flexibility is a certain amount of programming knowledge needed by the user in order to create the rules. We also recognize the fact, that a grammar based modeling approach may not be suited for very irregular architectural structures.

Figure 1. The CityEngine workbench as used in the RomeReborn project: the windows on the left and upper right show the building mass models acquired by scanning the *Plastico di Gismondi*. The lower window on the right shows the detailed building facades created by applying the Computer Generated Architecture (CGA) shape grammar rule files.

The shape grammar as implemented in the CityEngine is a programming language dedicated to generate architectural 3D content. As introduced in [27], we will use the term *CGA shape* from now on (CGA stands for Computer Generated *A*rchitecture). The CityEngine is meant as a tool for complete sites, and not only individual buildings. It also supports the creation of street networks and the addition of vegetation. Uncertainty about vegetation and streets can also be captured with the CGA rules. The CityEngine can take GIS data as input, such as building footprints or height maps. Figure 2 summarizes all this.

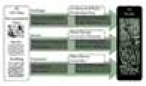

Figure 2. The CityEngine pipeline integrates CGA shape grammar rules for building reconstruction with GIS data such as building footprints and height maps. Note that the rules can also be used to generate hypotheses for street models and vegetation distribution.

The idea behind grammar-based modeling is to define rules that iteratively refine a design by adding more and more detail. These rules operate on shapes which consist of a geometry in a locally oriented bounding-box. Listing 1 shows a generic grammar rule. A crude predecessor shape is refined by replacing it by a series of finer successor shapes. The replacement can be conditioned on properties of the model created so far. A choice between alternative replacements can also be determined by chance, with pre-specified relative probabilities (also encoded in a condition). The replacement rule does not only have to specify the finer successor shapes to be added, but also how they have to be positioned with respect to each other, like a regular, horizontal repetition, for example.

Listing 1:A generic *CGA shape* grammar rule. Note that a rule must at least have one successor list (with at least one successor) and that the condition is optional if $N = 1$ (for syntactic details consult [29])

```
predecessor -->

  condition1: successor0 … successorM

  …

  conditionN: …
```

The column to the right in Figure 3 shows the generation sequence of a complete (although simplistic) building example. The corresponding rules are shown in Listing 2. These rules make use of a generic subdivision scheme (mass → facade → floor → window) which works for many modern styles. A similar rule set was used to reconstruct the *Candler* building and its beaux-arts patterns shown on the left side of Figure 3.

Listing 2: This complete rule set is able to generate the simple building shown in Figure 3 (right column). These rules work on mass models of arbitrary dimensions. Note that `comp` and `split` are *shape operations*, a special type of successor.

```
// STEP 1

Building -->

    comp (f) {front : Front | side: Side | top: Roof}

Front -->

    split (y){4 : GroundFloor | {~3.5: Floor}* }

Side -->

    split (y) {4 : Floor | {~3.5: Floor}}

Floor -->

    split (x) {1 : wall | {~3: tile} | 1 : wall}

GroundFloor -->

    split (x) {1 : Wall | {~3: Tile} *

          | ~3: Entrance | 1 : Wall}

// STEP 2

Tile -->

    split (x) {~1: Wall | 2 : Subtile | ~1: Wall}

Subtile -->

    split (y) {1 : Wall | 1.5 : Window | ~1: Wall}

Entrance -->
```

```
    split (x) {~1: Wall | 2 : split (y) {2.5 : Door
          | ~2: Wall} | ~1: Wall}
Window -->
    i (classic_window)
Door -->
    i (front_door)
```

Figure 3. **On the right, from top to bottom, we show a simple CGA shape grammar derivation sequence.** First, the mass model (aka. initial shape) is shown. Next, we show the intermediate result after a first subdivision (step in Listing 2) and the final model after step . This subdivision sequence (mass facade floor window) is quite generic and was used to reconstruct the *Candler* building (Atlanta, USA) which is shown on the left.

The terminal rules (where shapes are included that themselves do not result from compositions of other shapes) add predefined shapes, from a library of shapes typical for the architectural style of the building which is being modeled (like a Greek capital or a Gothic window). Interestingly enough, GML (Generative Modeling Language) [31] is an excellent example of how also such terminal components can be modeled efficiently using, again, procedural strategies.

Listing 3 shows an example where uncertainty is modeled using probabilistic and conditional rules. The example utilizes two different kinds of probabilistic attributes: the attributes denoted with `attr` are evaluated (i.e., random values are freezed) *prior* to the generation of the model and are typically used to control global design parameters, that is, these attributes stay constant during the whole generation process to ensure a consistent design even in the presence of uncertainty. The other kind of attributes in this example are used in the successor conditions (see `SubR` in the listing). They randomly select a successor *on each call of the rule*. In addition, comments can be inserted inside the grammar rules to directly document the (archeological) sources (cf. principle 4 of the LC [2]).

Listing 3: Uncertainty about the window dimensions is encoded into shape attributes `winW` and `winH` where the values have certain probabilities assigned. The actual value of the attributes is determined prior to the generation process and stays the same for each call to `Tile` and `Sub`. In contrast, rule `SubR` uses *conditions per successor list* to decide on the window height. In this latter case, each window would potentially get a different height.

```
attr randomly = 0
attr winW = 2 // fixed, no uncertainty
attr winH =
 60%: 1.5 // from "arch. source"
 20%: 1.6 // from "arch. source 2"
 else: 1.7
Tile -->
 case randomly == 0:
    split (x) {~1: Wall | winW : Sub | ~1: Wall}
 else:
    split (x) {~1: Wall | winW : SubR | ~1: Wall}
// sub tile alternative 1
Sub -->
  split (y) {~1: Wall | winH : Window | ~1: Wall}
// sub tile alternative 2
SubR -->
    60% : split (y) {~1: Wall | 1.5 : Win | ~1: Wall}
    20% : split (y) {~1: Wall | 1.6 : Win | ~1: Wall}
   else : split (y) {~1: Wall | 1.7 : Win | ~1: Wall}
```

In the next sections, we show further examples of projects carried out with the CityEngine. We offer them in an order that highlights additional features of our modeling tool one at a time. Section 4 combines these features and showcases the visualization of uncertainty using multiple models.

3.4. The Semper Observatory

The first example is an exact reconstruction of an existing building and does not exhibit any uncertainty. Nevertheless, it is included here to demonstrate the adaptive behavior of the facade rules, which is a precondition for valid models when multiple hypotheses are created.

Figure 4 shows a reconstruction of the observatory, originally designed by Swiss architect Gottfried Semper (1803–1879). The observatory was built from 1861 to 1864 and restored from 1995 to 1997. Semper was very educated in formal architecture and wrote a highly regarded book about style [33]. His buildings are examples of neoclassical architecture and include all elements of this widespread style.

Figure 4. **(a) Semper's Sternwarte (Observatory), photograph on the left, rendering on the right.** (b) The rule set used to reconstruct Semper's Observatory automatically adapts to any configuration of the underlying mass model. This is a very useful feature to quickly visualize multiple reconstruction-hypotheses of a site.

By including *self-similar volume shapes*, *facade ornaments*, *repetitions* and *self-occlusion* on a small scale, the Sternwarte is a very interesting example to model with the CityEngine [27]. Moreover, as the bottom row of Figure 4 shows, the facade design adapts to arbitrary variations of the underlying mass model. It would be easy to generate multiple models, if it would e.g., no longer be known how many storeys the building originally had, for instance.

3.5. Reconstruction of Puuc Buildings

The Puuc example builds upon the features used by Semper's observatory and introduces the so-called *control parameters* which are simply a set of shape attributes used to control the overall result of a grammar rule set (e.g., a building type). These parameters are the key tool to express uncertainty, for example, by storing a range of possible values (see `winH` example in Listing 3).

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 (Figure 5).

Figure 5. **Photograph of a well preserved Puuc-style building in Kiuic, Mexico.**

By creating a single set of rules for all Puuc building types with the help of archeologists of the German Archeological Institute, we were able to generate each of the building types in about 5 to 10 minutes (by simple modification of the control parameters of the grammar rule set) [34]. Additionally, there are rules to specify materials and textures. Please note that the building types have all been created according to specific archeological data and no random variations have been used for this reconstruction either. See Figure 6 for selected buildings, with height and molding combinations. These images have been rendered in OpenGL and are screenshots from the interactive previewing system of the CityEngine. Higher quality renderings can be created with offline tools.

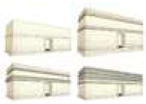

Figure 6. **Four building hypotheses that have been created by using the single rule set described in [34].** Simple modifications of its control parameters lead to the different building appearances.

Additionally, the rules have been extended to generate other more complex buildings. Figure 7 pictures a whole building. This image has been created with Pixar's RenderMan.

Figure 7. **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.

3.6. Reconstruction of Ancient Pompeii

The example of Pompeii is a large scale reconstruction project, beyond the actual excavations, as large sections of the site have not been excavated yet. Mostly only footprints are known, so that buildings had to be extruded with extensive use of probabilistic rules.

Pompeii, an ancient Roman town destroyed in 79 AD, was reconstructed in collaboration with archeologists who provided ground plans (footprints) and drawings/sketches of selected building types. We used this information to abstract 190 design rules to model the complete city including the streets and placement of trees [35]. The resulting model shown in Figure 8 has about 1.4 billion polygons at its highest level of detail. Figure 9 shows a typical (but hypothetical) street-level view of the city (also see Section 4).

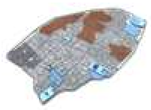

Figure 8.

Figure 9. **This (hypothetical) street-level view of ancient Pompeii was created by translating archeological drawings, figures and ground-plans into CityEngine rules and rendering the resulting geometry with Pixar's RenderMan.**

3.7. Rome Reborn 2.0

Rome Reborn [36, 37] is an international project, inspired and led by Prof. B. Frischer, University of Virginia, that aims to create an interactive 3D digital model illustrating the urban development of ancient Rome (Figure 10). As a start (i.e., Rome Reborn 1.0), a model of 4th century AD Rome was created. At that time, all well-known, major buildings had already been erected.

Figure 10. **Rome Reborn 2.**0: The approx. 7000 domestic buildings between the major monuments have been reconstructed using grammar-based procedural modeling (as implemented by the CityEngine). The upper figure shows the area around the Circus Maximus and the lower figure shows a typical, although hypothetical, street view. Both images have been rendered in mental images' mentalray (Circus Maximus model courtesy of Bernard Frischer, IATH).

The detailed 3D models of Rome's monuments—such as the famous Colosseum and Circus Maximus—have been created manually in several man-years of work by experts in archeology and computer graphics from all over the world. To reconstruct the surrounding urban environment and "fill-in" buildings in similar detail using similar methods would have been a daunting task. Thus, grammar-based procedural modeling was used for the Rome Reborn 2.0 version. In this way, about 7000 domestic buildings (insulae, the apartment buildings of the time where the lower and middle classes would live) have been reconstructed. As a matter of fact, also about 60% of ancient Rome's temples were modeled with the CityEngine.

Rome Reborn 2.0 combines all the features used in the previous examples and adds one peculiarity to the input data: in contrast to the Pompeii model, the Rome Reborn 2.0 model did not start with building footprints but with crude polyhedral volumes for each building. These were the simplistic representations of the insulae from the Rome Reborn 1.0 version. They were obtained by laser-scanning the famous "Plastico di Gismondi", a huge plaster exhibited and preserved in the Museum of Roman Civilization in Rome [38]. This put the adaptive behavior of the facade rules (Section 3.4) to the test, as an enormous variety of facade topology had to be handled. Another novelty is that the Rome Reborn 2.0 model mixes manually modeled monuments of the first stage of the project (Rome Reborn 1.0) with these procedural insulae and temple models. Near certain monuments, the amount of detail in the procedural models actually had to be limited in order to keep the visual balance with those manually modeled monuments in check. Figure 10 shows a snapshot of a small part of the model, rendered with the "mental image RealityServer".

4. Examples of Reconstruction Uncertainty

As already discussed in Sections 1 and 2, we propose to create multiple models of a building or site, that differ in the aspects which are not certain.

In the already mentioned case of Pompeii, for most buildings only the footprint is known with reasonable certainty. Thus, quite different buildings can be erected for each footprint. It has to be added that the rule set used was location dependent, as extra information such as the affluence, building period, and population density of a city district was fed into the system. Figure 11 shows pairs of views of two different models generated from the same underlying CityEngine description. As can be seen, virtually everything—except the overall style—is different: roof types, facade and floor layouts, window and door dimensions, and so forth. These views have not been rendered with a high-end renderer, but nevertheless demonstrate the idea of showing multiple, realistic models to give a feel of uncertainty versus constancy of style to a naive observer.

Figure 11. **For each view two hypotheses have been created in order to compare roof types, floor layouts and door/window dimensions.** Note that in this example, the footprints of the buildings and their functions (e.g., atrium) are fixed, that is, there is a high certainty of correctness for the footprints.

In a second example, we would like to show how the uncertainty can be controlled in a more fine-grained manner, for example, for specific parts of a building. Figure 12 shows multiple hypotheses for a typical facade from the "Rome Reborn 2.0" project. We artificially reduced the amount of uncertainty on the lower floor by fixing the layout to the sequence door-window-door. Also, we constrained the doors to the left and right. The left door is always large and the right door is always small. The small window between the doors has a probability of 50%. The upper floor uses the unmodified rules from "Rome Reborn 2.0". Although this example does not correspond to a genuine archeological case, it demonstrates how a part of a building could remain fixed between models if there would exist more precise knowledge about it, whereas other parts can remain as unconstrained as before.

Figure 12. **This example uses a subset of the rules created for the "Rome Reborn 2.**0" project. We constrained the rule attributes such that the lower floor always has the same layout (door-window-door) and always contains a larger door at the left and a smaller door at the right. The window between the doors has a probability of 50% (the first five facades are with, the last 5 without the window). The upper floor uses the original range of layouts designed for "Rome Reborn".

5. Discussion and Conclusion

Producing multiple, realistic models as a way of sampling the underlying, partially uncertain description of buildings and sites and thereby lift the tension between veridical and realistic model visualization, hinges on the ability to efficiently create 3D models. We have proposed procedural modeling as a strategy to make this possible. Hopefully, the examples have convinced the reader that procedural modeling does indeed carry a great promise.

But also in 3D modeling, there is nothing like a free meal. Rather than directly seeing the outcome of one's modeling efforts, procedural modeling requires that first the rule-based description be created. This is currently done by writing the grammar rules in a text editor and it requires some training. On the other hand, as the community of procedural modelers grows, one can imagine that the rule sets for many styles will be available before the project starts, or that the required style can be derived from a similar style, with minor modifications. Also, work is currently being done to provide the artists with visual editing tools for creating the rule sets.

With the examples of Pompeii and Rome Reborn 2.0 we show multiple samples of parts of models with high uncertainty. Parts about which the experts feel certain remain unchanged. Moreover, the proposed multi-model strategy can also naturally express the relative certainty of different hypotheses, by adhering to each with their prescribed frequency. If one believes there is about 90% chance it was one way, and about 10% that a second hypothesis was the right one, 9 out of 10 models can follow the first hypothesis, and only 1 out of 10 the second. This kind of expert pondering would even be difficult to show if one would simply juxtapose two models according to the two hypotheses. Exposing the audience to a multitude of models would seem the natural way to go.

This all said, quite a bit of work on the CityEngine remains to be done in order to make it fully compatible with the uncertainty dream. If for a building in Pompeii it is known that there were two windows of certain dimensions on the ground floor, say, and this ground floor would leave space for a lot more of such windows, then fixing these structures would not necessarily coincide with fixing a certain parameter of one rule. Structures one feels certain about may cut right down the middle of structures normally generated by a rule or a set of rules. One possibility to fix such structures would be to implement interactive editing tools which duplicate the rules and attributes belonging to this part automatically. One copy would be used to freeze the part one is certain about, the other copy would be used to generate variations for the other, uncertain parts as before. This has not been done yet.

Also, in this paper we have focused on the model of a building or site at a fixed moment in time. In reality, many monuments and sites have changed during the ages [10]. As a matter of fact, the grand vision behind the Rome Reborn project is not so much to produce a model of the city for the 4th Century AD, but to produce similar models for other time periods, in order to show the city's evolution. That will entail the creation of time series of models, where again certain aspects remain fixed and others vary. Such modeling may then require several time series of models, if we want to stick to our strategy. But as the London Charter [2] suggests, 3D models can indeed be an ideal way of illustrating changes over time. Even if this is an additional challenge, procedural modeling can again help to deliver.

Acknowledgments

The authors gratefully acknowledge EC projects 3D-COFORM and V-CITY for their support. They want to thank their colleagues, who have contributed in essential ways to different example cases discussed in this paper. Their thanks also go to the reviewers, whose comments helped to improve the quality of this paper a lot.

References

1. B Frischer, F Niccolucci, N Ryan, J Barcelo, From CVR to CVRO: the past, present, and future of cultural virtual reality. in *Proceedings of the EuroConference on Virtual Archaeology between Scientific Research and Territorial Marketing (VAST '00), November 2000, Arezzo, Italy*, , ed. by Niccolucci N
2. R Beacham, F Niccolucci, H Denard, The London Charter (Version 2), . 1, February 2009, http://www.londoncharter.org/ webcite
3. R Beacham, F Niccolucci, H Denard, An introduction to the London charter. Papers from the Joint Event CIPA/VAST/EG/EuroMed (2006)
4. M Forte, About virtual archaeology: disorders, cognitive interactions and virtuality. Virtual Reality in Archaeology, 247–259 (2000)
5. J Barcelo, Visualising what might be: an introduction to virtual reality techniques in archaeology. Virtual Reality in Archaeology, 9–36 (2000)
6. A Müller, R Ammon, *The Seven Wonders of the World* (McGraw-Hill, New York, NY, USA, 1966)

7. L Van Gool, M Waelkens, P Müller, T Vereenooghe, M Vergauwen, Total recall: a plea for realism in models of the past

8. S Roskams, *Excavation* (Cambridge University Press, Cambridge, UK, 2001)

9. M Roussou, G Drettakis, Photorealism and non-photorealism in virtual heritage representation. in *Proceedings of the 1st Eurographics Workshop on Graphics and Cultural Heritage (VAST '03), November 2003*, , ed. by Chalmers A, Arnold D, Niccolucci F, pp. 47–56

10. T Zuk, S Carpendale, WD Glanzman, Visualizing temporal uncertainty in 3D virtual reconstructions. Proceedings of the 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST '05), 2005

11. A Palladio, *The Four Books on Architecture* (The MIT Press, Cambridge, Mass, USA, 2002)

12. Vitruvius, in *The Ten Books on Architecture (Bks*, ed. by . I–X) (Dover, New York, NY, USA, 1960)

13. The Sagalassos research project (July 2009, http://www), . sagalassos.be/ webcite

14. M Davis, R Sigal, EJ Weyuker, MD Davis, *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science* (Academic Press, New York, NY, USA, 1994)

15. M Sipser, *Introduction to the Theory of Computation* (Course Technology, Boston, Mass, USA, 1996)

16. H Ehrig, G Engels, H-J Kreowski, G Rozenberg, *Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages and Tools* (World Scientific, Hackensack, NJ, USA, 1999)

17. G Stiny, *Pictorial and Formal Aspects of Shape and Shape Grammars* (Birkhauser, Basel, Switzerland, 1975)

18. G Stiny, Introduction to shape and shape grammars. Environment and Planning B **7**(3), 343–361 (1980). Publisher Full Text

19. DE Knuth, Semantics of context-free languages. Mathematical Systems Theory **2**(2), 127–145 (1968). Publisher Full Text

20. P Prusinkiewicz, A Lindenmayer, *The Algorithmic Beauty of Plants* (Springer, Berlin, Germany, 1991)

21. P Wonka, M Wimmer, F Sillion, W Ribarsky, Instant architecture. ACM Transactions on Graphics **22**(3), 669–677 (2003). Publisher Full Text

22. F Downing, U Flemming, The bungalows of buffalo. Environment and Planning B **8**(3), 269–293 (1981). Publisher Full Text

23. J Duarte, in *Malagueira grammar—towards a tool for customizing alvaro siza's mass houses at malagueira, Ph*, ed. by . D. thesis (MIT School of Architecture and Planning, Cambridge, Mass, USA, 2002) PubMed Abstract | Publisher Full Text

24. U Flemming, More than the sum of its parts: the grammar of queen anne houses. Environment and Planning B **14**, 323–350 (1987). Publisher Full Text

25. H Koning, J Eizenberg, The language of the prairie: Frank Lloyd Wright's prairie houses. Environment and Planning B **8**(3), 295–323 (1981). Publisher Full Text

26. G Stiny, WJ Mitchell, The palladian grammar. Environment and Planning B **5**(1), 5–18 (1978). Publisher Full Text

27. P Müller, P Wonka, S Haegler, A Ulmer, L Van Gool, Procedural modeling of buildings. ACM Transactions on Graphics **25**(3), 614–623 (2006). Publisher Full Text

28. B Watson, P Müller, O Veryovka, A Fuller, P Wonka, C Sexton, Procedural urban modeling in practice. IEEE Computer Graphics and Applications **28**(3), 18–26 (2008). PubMed Abstract

29. Procedural Inc./CityEngine (2009, http://www), . procedural.com/ webcite

30. M Lipp, P Wonka, M Wimmer, Interactive visual editing of grammars for procedural architecture. ACM Transactions on Graphics **27**(3, article 102), 1–10 (2008)

31. R Berndt, DW Fellner, S Havemann, Generative 3D models: a key to more information within less bandwidth at higher quality. Proceedings of the 10th International Conference on 3D Web Technology (Web3D '05), March 2005, Bangor, UK, 111–122

32. PJ Birch, SP Browne, VJ Jennings, AM Day, DB Arnold, Rapid procedural-modelling of architectural structures. Proceedings of the International Symposium on Virtual Reality, Archeology and Cultural Heritage (VAST '01), November 2001, Glyfada, Athens, 187–196

33. G Semper, HF Mallgrave, M Robinson, *Style in the Technical and Tectonic Arts, or, Practical Aesthetics* (J. Paul Getty Trust Publications, Los Angeles, Calif, USA, 2004)

34. P Müller, T Vereenooghe, P Wonka, I Paap, L Van Gool, Procedural 3D reconstruction of Puuc buildings in Xkipche. Proceedings of the 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST '06), 2006, Nicosia, Cyprus, 139–146

35. P Müller, T Vereenooghe, A Ulmer, L Van Gool, Automatic reconstruction of Roman housing architecture. *Proceedings of the International Workshop on Recording, Modeling and Visualization of Cultural Heritage, May 2005, Ascona, Switzerland* (Balkema Publishers (Taylor & Francis group)), pp. 287–297

36. B Frischer, et al. Rome Reborn 2.0 (July 2009, http://www), . romereborn.virginia.edu/rome_2.0.php webcite

37. KA Dylla, P Müller, A Ulmer, S Haegler, BD Frischer, Rome reborn 2.0: a framework for virtual city reconstruction using procedural modeling techniques. Proceedings of the 37th Annual Conference, Envisioning the Past: Virtual Reconstructions of Archaeological Sites (CAA '09), 2009 in press

38. G Guidi, B Frischer, M de Simone, et al. Virtualizing ancient Rome: 3D acquisition and modeling of a large plaster-of-Paris model of imperial Rome. in *Videometrics VIII, January 2005, San Jose, Calif, USA*, vol. 5665, , ed. by Beraldin J-A, El-Hakim SF, Gruen A, Walton JS, pp. 119–133