CEIG - Spanish Computer Graphics Conference (2017)
F. J. Melero and N. Pelechano (Editors)

An Interactive Tool for Modeling Ancient Masonry Buildings

Josep Lluis Fita, Gonzalo Besuievsky and Gustavo Patow

ViRVIG, University of Girona, Spain

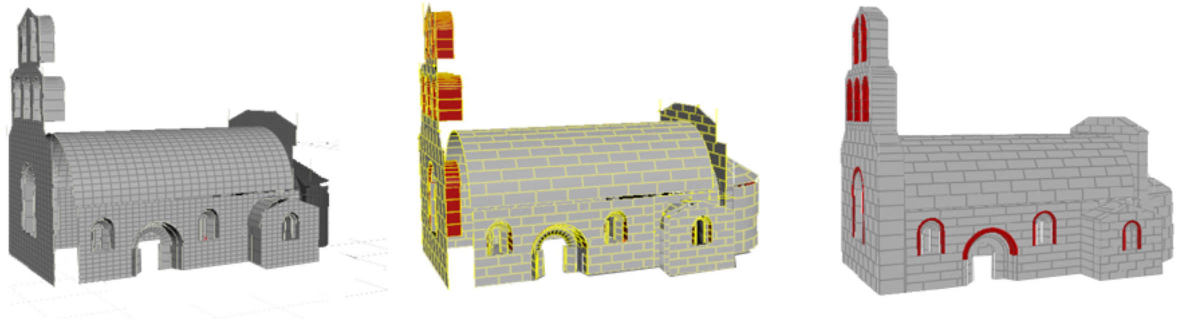

Figure 1: *An input geometry of an ancient building like a church (left), is transformed by our interactive tool in a mesh of stone bricks (middle). Then, we added, through procedural modeling, the volume in the result geometry and the physical features for structural simulation (right). Colors on the final model were added for aesthetic purposes, following actual historic patterns.*

Abstract

We present a new pipeline of an interactive tool that combines procedural modeling of ancient masonry buildings with structural simulation. The tool has been designed for taking an input geometry of an ancient building and re-meshing it into a suitable mesh with a low quad density. Then, it creates the brick outlines on the mesh and adds the brick volumes for structural simulation. The tool was designed and built on a set of off-the-shelf tools. We tested and demonstrated its viability by modeling of a Romanesque church based on a real one from the 11th century, such as the church of Santa Maria de Agullana.

Categories and Subject Descriptors (according to ACM CCS): Procedural Modeling [Computer Graphics]: Virtual Historical Buildings—Masonry Structures

1. Introduction

Nowadays, the technological improvements of Information and Communication Technologies (ICT) have carried out a revolution in areas like video-games, the film industry, urban planning and, more recently, cultural heritage. In particular, for the later group, modeling techniques based on reconstruction of ancient buildings have attracted considerable interest in the Computer Graphics community over the last years, where a large bulk of research has been devoted to the development of tools to facilitate the task to stakeholders, like curators, architects and authorities in charge of preservation and maintenance of ancient buildings, but also to historians and art historians for the many possibilities this technology opens.

However, most of these techniques are based on the visual creation of the buildings as simple 3D objects where a texture is applied with the aim of giving a realistic view. Among the literature, only a few works have focused on masonry buildings and combined the visual results with structural analysis for the simulation of the environments. However, the tools developed so far require a high level of expertise that the ordinary user (e.g., a museum curator or an art historian) usually does not have.

For this reason, we focused our work on modeling masonry buildings, especially Romanesque and Medieval churches, presenting an interactive modeling tool for ordinary users. Our technique can easily transform an input shape provided by the user with any modeling tool (e.g., from Autodesk's Autocad to more sophisticated procedural tools) into a masonry structure ready for structural simulation or analysis. Our main contribution is a tool that

DOI: 10.2312/ceig.20171212

re-meshes a given geometry of a building into a masonry structure feasible for simulation through off-the-shelf tools.

2. Previous Work

One of the first works to deal with the procedural creation of buildings was the one presented by Muller et al. [MWH*06], based on the concept of shape grammars. These grammars worked by iteratively replacing labeled geometry parts by new parts, which were obtained by the execution of a *rule* specifying the transformation operations to apply. This was extended by many authors [KK12, MWA*12, Pat12] trying to simplify modeling tasks, but in almost all cases only considering the outer building shape and not its structure. On the Cultural Heritage side, [JTC09] recovered a 3D model from a single image and [SJ13] introduced a technique based on the generation of buildings procedurally from archaeological data GIS. Then, they export the model to Unity with the aim of generating an interactive application. [CSS*13] introduced a methodology based on procedural modeling of buildings through Roman techniques of construction and the combination of archaeological data. However, none of these works focus on masonry buildings and their structural stability.

Whiting and co-authors [WOD09] introduced a method to create parameterized ancient buildings from a set of rules. The method automatically searches the parametric space for a stable configuration according to physical constraints with the resolution of inverse static problems through quadratic equations of equilibrium. Later on, Fita et al. [FBP16b] developed a similar technique, but based on simple off-the-shelf tools. [WSW*12] integrated the study of a building soundness through the integration of architecture design and structural analysis. The aim of their method is to create a new stable structure through the analysis of the stability gradient, according to constraints previously introduced by the user, such as horizontal direction, vertical direction and block thickness. Panozzo and co-workers [PBSH13], introduced an algorithm that automatically generates a 3D masonry structure from an input shape (e.g., a NURBS surface) in such a way that it is a self-supporting structure without mortar, and its shape is as close as possible to the one given as input, but constructible with masonry blocks. More recently, Deuss and colleagues [DPW*14], extended this technique to processes all kind of generated masonry models, by detecting subsets of the original input masonry structure that can be built independently of any other part of the same roof, providing detailed instructions to build these through blocks, holding them with hooks and chains. For an in depth survey of the alternatives for structural analysis in cultural heritage, we suggest the interested reader to refer to [FBP16a].

However, in spite of all these efforts, we think that there is still need for an interactive tool designed for its easy use combining procedural modeling and structural simulation, filling in the gap that still exists with this kind of tools for non-expert users.

3. Interactive tool

This section presents an interactive tool adapted and designed for transforming an input mesh representing the basic shape of an ancient building (or one of its components), such as a Romanesque church provided by a designer (e.g., a cultural heritage expert), into a masonry structure ready for simulation. We show the pipeline at figure 2. In the following sections we will present the process step by step, starting from the original shape, transforming it into a set of masonry bricks, and ending in the simulation itself.

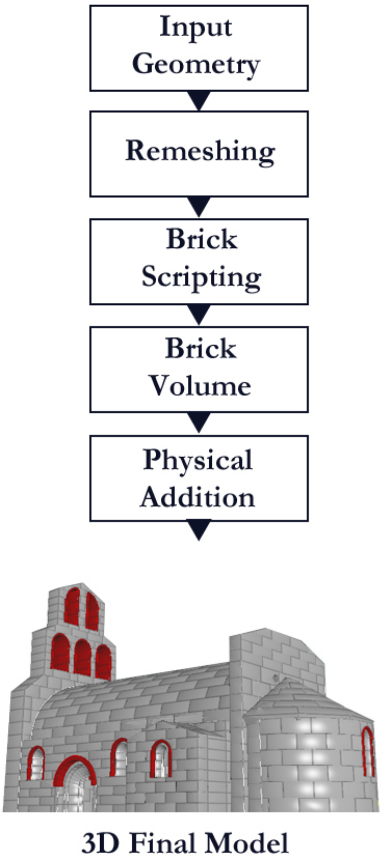

Figure 2: *Scheme of our pipeline.*

3.1. Input geometry

Here we assume that users start with a modeling tool, which can be either a traditional tool like Autodesk's Maya, Autodesk's 3DSMax or Blender, or a procedural tool, and design the building basic structure. This model can be made of any kind of surface, from NURBS to simple polygonal representations. Once users have finished this design stage, they can use the tool for easily transforming the created building into a masonry structure made of bricks and ready for structural studies.

3.2. Remeshing

When users make the design of a building, usually they focus on its shape and do not think about the final masonry buildings. This is the main objective of our tool. The first step towards our model transformation is to convert the input shape into a suitable mesh. If the input shape is made of splines such as a NURBS surface, a standard quad conversion tool is used. In our implementation, we used Houdini's *convert* tool for this task, resulting in a high resolution, quad-tesellated surface. Once the input surface has been converted to a regular quad mesh, we need to remesh it again in order to generate quads that are, more or less, the size of the final bricks in our masonry model.

For the remeshing process, the first input datum is the desired length of the bricks, which in our implementation is specified by the user. The desired brick height is obtained automatically as half of brick width, with the aim of preserving a proper shape ratio for the bricks. Then, our algorithm starts an iterative process, collapsing vertical or horizontal edges to generate larger quads, until the average quad size equals the desired value from the information provided. If the current value is smaller than the desired one, our tool dissolves the corresponding mesh edges to generate a larger quad. This process is first done for each horizontal row of bricks, ensuring that the correct dimensions and positions are obtained for the bricks. Our algorithm repeats the process until it has remeshed the entire mesh. We can see a resulting remeshed model at Figure 3.

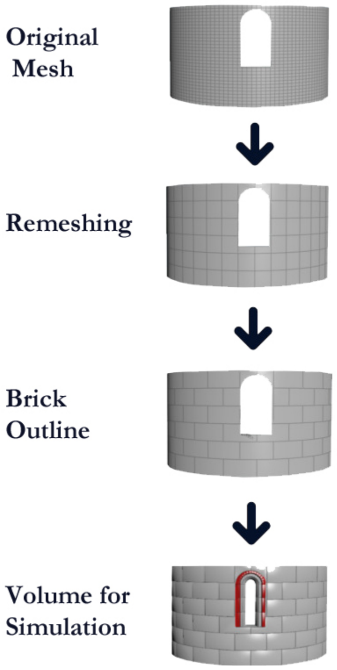

Figure 3: *Starting with an original mesh shape. The process Remeshes and makes the brick outline over the given mesh. Finally, it adds the volume for structural simulation.*

3.3. Brick shape

Once the remeshing step has been completed, our interactive tool is capable of detecting the features of the resulting mesh (e.g. points, edges and primitives) and through these features it selects the proper edges with the aim of making the brick mesh.

To achieve this result, our interactive tool uses a set of python scripts. These scripts start by loading all the geometry features of the mesh. Then, the procedure iterates through all the edges in the model, generating a list of edges, containing the edges that have only two neighboring primitives. This is done for avoiding selecting edges that could produce problems in future stages of the processing (e.g., edges near a corner of the mesh or edges belonging to an architectural structure such as windows or doors), in order to correctly preserve the original shape. We filter the resulting edge list by selecting the vertical ones. Depending whether a brick row is odd or even, we select its edges interleaved to generate a brick-like pattern, see Figure 3. This step is repeated for all the brick rows in the structure. With the list of selected edges, we apply a dissolve operation to generate the brick outlines on the mesh.

3.4. Brick volume and physics

When the scripts have finished their process, our tool applies an operation for transforming the resulting mesh into a set of bricks. With this action, we ensure that the bricks are working independently, i.e., they do not share any vertex, edge or face, when we start the structural simulation. Moreover, the width assigned to each brick is given by the user, and it is achieved with standard extrusion tools. for practical visualization, a suitable color for the structure can be also assigned at this stage for each individual brick, or families of bricks. Finally, the tool adds the physical properties corresponding to an ancient building, such as density of the stone and other materials, friction, and environmental temperature, among others. When our tool has finished these steps, the 3D model of the building is ready for structural simulation.

4. Results

We have made a 3D prototype of an old 11th century Romanesque church, *Santa Maria de Agullana*. Through it, we have designed a set of tests for verifying the feasibility of our interactive tool. The software platform we used for the implementation is *Houdini* from SideFX [Sid12], together with the open source *Bullet* solver (available as Houdini plug-in). The physical features for our test are similar to the ones reported in [FBP16a]. In this case, the *master* walls have a width of $2m$ and the remaining walls have a width of $1m$, the vault roof has a width of $0.7m$ and we set a value for the rotational stiffness of $9000Nm/rad$, the roof of the apse has the same physical values as the vault roof. As a result, we can see on figure 4 that our generated model satisfies the soundness criterion resulting from the structural simulation. After the simulation, in the figure we can see that our prototype holds together, even though it has suffered some structural movements that can be appreciated mainly in the curvature of the roof.

5. Conclusion

We decided to develop a pipeline for combining the procedural modeling of ancient masonry buildings and the structural simulation techniques for filling the gap in the state-of-the-art literature. We have based our development on off-the-shelf tools for building and testing a model based on a Romanesque church from the 11th century, following the specifications of the pipeline and the features

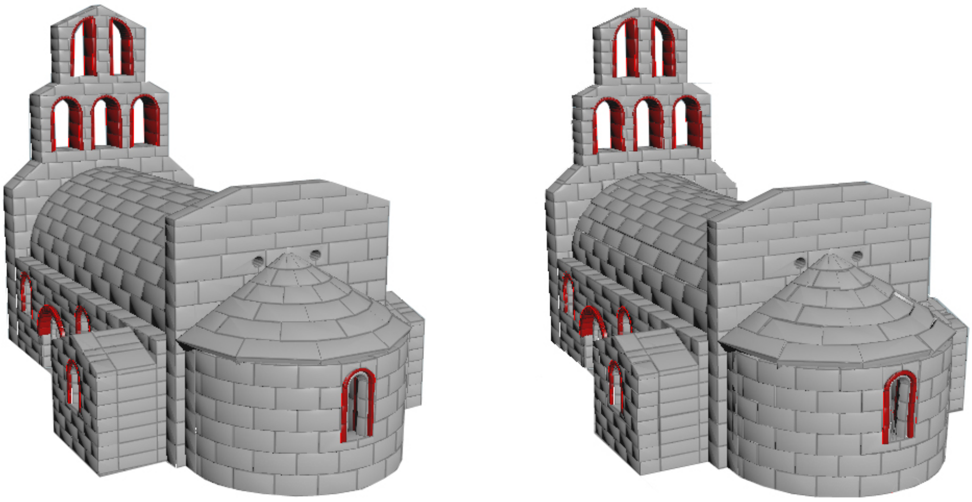

Figure 4: *Our model, (left) before and (right) after the structural simulation test. Observe how the roof after the simulation is not a perfectly cylindrical shape anymore, but it has got a slight saddle shape.*

described in the previous work [FBP16b]. As a result, stakeholders (curators, art historians, etc.) will be able to better understand, how the different elements of a masonry building (e.g. walls, arches, canon vault, etc.) work in combination.

However, the tool we present in this paper is only another step in the research bridging the gap between Computer Graphics, architecture and structural simulation of ancient masonry buildings. Further improvements for our tool include building a procedural library of shapes based on Romanesque, Gothic and other architecture styles with the aim of facilitating the work of users. This way, by simply assembling structural sound components a non-expert user as the ones enumerated above would not only be able to create sound masonry buildings, but also would be able to text new hypothesis in a simple, but efficient way.

Acknowledgments

We would like to thank the anonymous reviewers for their useful comments. This work was partially funded by the TIN2014-52211-C2-2-R project from Ministerio de Economía y Competitividad, Spain.

References

[CSS*13] CAPPELLINI V., SALERI R., STEFANI C., NONY N., DE LUCA L.: a Procedural Solution to Model Roman Masonry Structures. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2 (July 2013), 149–153. 2

[DPW*14] DEUSS M., PANOZZO D., WHITING E., LIU Y., BLOCK P., SORKINE-HORNUNG O., PAULY M.: Assembling self-supporting structures. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 214:1–214:10. 2

[FBP16a] FITA J. L., BESUIEVSKY G., PATOW G.: A Perspective On Procedural Modeling Based On Structural Analysis. *Virtual Archaeology Review Accepted for publication* (2016). 2, 3

[FBP16b] FITA J. L., BESUIEVSKY G., PATOW G.: Tools for Structural Analysis and Optimization of Procedural Masonry Buildings. In *Spanish Computer Graphics Conference (CEIG)* (2016), Garcia-Alonso A., Masia B., (Eds.), The Eurographics Association. 2, 4

[JTC09] JIANG N., TAN P., CHEONG L. F.: Symmetric architecture modeling with a single image. *ACM Trans. Graph. 28*, 5 (2009), 113:1–113:8. 2

[KK12] KRECKLAU L., KOBBELT L.: Interactive modeling by procedural high-level primitives. *Computers & Graphics*, 0 (2012), –. 2

[MWA*12] MUSIALSKI P., WONKA P., ALIAGA D. G., WIMMER M., VAN GOOL L., PURGATHOFER W.: A survey of urban reconstruction. In *EUROGRAPHICS 2012 State of the Art Reports* (May 2012), EG STARs, Eurographics Association, pp. 1–28. 2

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 2

[Pat12] PATOW G.: User-friendly graph editing for procedural modeling of buildings. *IEEE Computer Graphics and Applications 32* (2012), 66–75. 2

[PBSH13] PANOZZO D., BLOCK P., SORKINE-HORNUNG O.: Designing unreinforced masonry models. *ACM Transactions on Graphics - SIGGRAPH 2013 32*, 4 (July 2013), 91:1–91:12. 2

[Sid12] SIDEFX: Houdini 12, 2012. http://www.sidefx.com. 3

[SJ13] SALDANA, JOHANSON: Procedural modeling for rapid-prototyping of multiple building phases. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-5/W1* (Feb. 2013). 2

[WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28* (December 2009), 112:1–112:9. 2

[WSW*12] WHITING E., SHIN H., WANG R., OCHSENDORF J., DURAND F.: Structural optimization of 3d masonry buildings. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 159:1–159:11. 2