TEMPORAL EVOLUTION OF ANCIENT BUILDINGS

Josep Lluis Fita López

DOCTORAL THESIS

TEMPORAL EVOLUTION OF ANCIENT BUILDINGS

Josep Lluis Fita López

2019

DOCTORAL THESIS

TEMPORAL EVOLUTION OF ANCIENT BUILDINGS

Josep Lluis Fita López

2019

DOCTORAL PROGRAM IN TECHNOLOGY

Supervisors:
Dr. Gonzalo Besuievsky and Dr. Gustavo A. Patow

This thesis is presented to obtain the degree of PhD at the University of Girona

Acknowledgments

I would like to thank my supervisors Gonzalo Besuievsky and Gustavo A. Patow for their wisdom, patience and advice that have influenced, encouraged and helped me to the completion of this thesis.

I would like to thank my friends and especially my family: my parents, my sister, my niece and nephew for supporting me during these years of my life.

List of Publications

Here we list the journals and proceedings of national and international conferences where this work has been published:

- FITA J., BESUIEVSKY G., PATOW G.: A *Perspective on Procedural Modelling based on Structural Analysis*. Virtual Archeology Review 8, 16 (2017), 44-50.

- FITA J., BESUIEVSKY G., PATOW G.: *Tools for Structural Analysis and Optimization of Procedural Masonry Buildings*. In Spanish Computer Graphics Conference (CEIG)(2016), 109-113.

- FITA J., BESUIEVSKY G., PATOW G.: *An Interactive Tool for Modelling Ancient Masonry Buildings*. In Spanish Computer Graphics Conference (CEIG)(2017), 73-76.

- FITA J., BESUIEVSKY G., PATOW G.: *Earthquake Simulation for Ancient Building Destruction*. In Eurographics Workshops on Graphics and Cultural Heritage(2018), 141-145.

- FITA J., BESUIEVSKY G., PATOW G.: *Earthquake Simulation Over on Ancient Masonry Buildings*. Submitted (2019).

- FITA J., BESUIEVSKY G., PATOW G.: *A Virtual Reality (VR) Front-End for Earthquake Simulation*. In Spanish Computer Graphics Conference (CEIG)(2019).

- FITA J., BESUIEVSKY G., PATOW G.: *Seismic Simulation on Virtual Reality*. In Eurographics Workshops on Graphics and Cultural Heritage (2019).

List of Acronyms

CAD Computer-Aided-Design

GIS Geographical Information Systems

NURBS Non Uniform Rational B-Splines

PM Procedural Modelling

SM Solid Modelling

VR Virtual Reality

XR X-Reality

List of Figures

List of Tables

Contents

Abstract

Nowadays, the improvement of computer skills and technology in Computer Graphics have benefited fields such as Cultural Heritage, where the main efforts have focused on the digital preservation of historic buildings or urban structures.

In the state of the art, we have detected a lack of techniques based on procedural models of historical buildings combined with structural analysis, which would allow us to understand how a building was designed or why it was conceived in a certain way. In this thesis we have developed a technique to procedurally model ancient stone buildings combined with structural simulation and we have demonstrated its viability based on non-specialized tools designed for cultural heritage users.

On the other hand, some historical events involving natural phenomena, such as volcanoes or earthquakes, determined the evolution of a civilization or a city in which a large part of its urban infrastructure was affected. In the current state of the art on cultural heritage, we have seen that the presence of studies on the reproduction of this type of events to help users in their research is testimonial. For this purpose, in this thesis we present a low-cost tool designed for users not expert in geology, such as cultural heritage professionals, that allows the reproduction of an earthquake on old stone buildings.

Although Virtual Reality has become suitable for the diffusion of historical past events or the recreation of ancient buildings in Cultural Heritage, we have to say that the cost of the most immersive virtual reality devices is still expensive. Thus, in this thesis we have also designed a virtual reality pipeline compatible with low-cost smart-phones where the combination of structural and seismic simulation over ancient buildings is available for any kind of users.

Resumen

Hoy en día, la mejora de las habilidades y la tecnología en computación en Gráficos por Computador ha beneficiado a campos como el Patrimonio Cultural, donde los esfuerzos principales se han centrado en la preservación digital de edificios históricos o estructuras urbanas.

En el estado del arte, hemos detectado una falta de técnicas basadas en modelos procedurales de edificios históricos combinadas con análisis estructurales, que nos permitan comprender cómo se diseñó un edificio o por qué se lo concibió de cierta manera. En esta tesis hemos desarrollado una técnica para modelar de manera procedural edificios antiguos de piedra, combinándola con simulación estructural, y hemos demostrado su viabilidad basada en herramientas no especializadas diseñadas para usuarios de patrimonio cultural.

Por otro lado, algunos eventos históricos relacionados con fenómenos naturales, como volcanes o terremotos, determinaron la evolución de una civilización o una ciudad en la que una gran parte de su infraestructura urbana se vio afectada. En el estado actual del arte sobre el patrimonio cultural hemos visto que la presencia de estudios sobre la reproducción de este tipo de eventos para ayudar a los usuarios en su trabajo de investigación, es testimonial. Para este propósito, en esta tesis presentamos una herramienta de bajo costo diseñada para usuarios no expertos en geología, como los profesionales del patrimonio cultural, que permite la reproducción de un terremoto en edificios antiguos de piedra.

Aunque la Realidad Virtual se ha vuelto adecuada para la difusión de eventos históricos pasados o la recreación de edificios antiguos en el Patrimonio Cultural, tenemos que remarcar que el costo de los dispositivos de realidad virtual más inmersivos es todavía costoso. Por lo tanto, en esta tesis hemos diseñado un sistema de realidad virtual adecuado y compatible con teléfonos inteligentes de bajo costo donde la combinación de simulación sísmica y estructural sobre edificios antiguos está disponible para cualquier tipo de usuario.

Resum

Avui dia, la millora de les habilitats i la tecnologia en computació en Gràfics per Computador ha beneficiat camps com el Patrimoni Cultural, on els esforços principals s'han centrat en la preservació digital d'edificis històrics o estructures urbanes.

En l'estat de l'art, hem detectat una manca de tècniques basades en models procedurals d'edificis històrics combinades amb anàlisis estructurals, que ens permetin comprendre com es va dissenyar un edifici o per què es el va concebre de certa manera. En aquesta tesi hem desenvolupat una tècnica per modelar de manera procedural edificis antics de pedra, combinant-la amb simulació estructural, i hem demostrat la seva viabilitat basada en eines no especialitzades dissenyades per a usuaris de patrimoni cultural.

D'altra banda, alguns esdeveniments històrics relacionats amb fenòmens naturals, com volcans o terratrèmols, van determinar l'evolució d'una civilització o una ciutat en la qual una gran part de la seva infraestructura urbana es va veure afectada. En l'estat actual de l'art sobre el patrimoni cultural, hem vist que la presència d'estudis sobre la reproducció d'aquest tipus d'esdeveniments per ajudar els usuaris en el treball de recerca, és testimonial. Per a aquest propòsit, en aquesta tesi presentem una eina de baix cost dissenyada per a usuaris no experts en geologia, com els professionals del patrimoni cultural, que permet la reproducció d'un terratrèmol en edificis antics de pedra.

Encara que la realitat virtual s'ha tornat adequada per a la difusió d'esdeveniments històrics passats o la recreació d'edificis antics en el Patrimoni Cultural, hem de remarcar que el cost dels dispositius de realitat virtual més immersius és encara costós. Per tant, en aquesta tesi hem dissenyat un sistema de realitat virtual adequat i compatible amb telèfons intel·ligents de baix cost on la combinació de simulació sísmica i estructural sobre edificis antics està disponible per a qualsevol usuari.

Chapter 1

Introduction

1.1 Thesis Overview

Over the last years, modelling buildings or urban environments has become a popular subject in Computer Graphics. The development of new building-modelling techniques has benefited other areas of science, such as urban planning, film, video-games and, more recently, Cultural Heritage.

Among the techniques used by the researchers in Cultural Heritage for the recreation of ancient buildings, we can find solid modelling techniques, which mainly use polygonal primitives and are based on 3D modelling or Computer-Aided-Design (CAD) programs, see Figure 1.1. However, the generation of virtual environments such as ancient buildings still requires a considerable effort and some researchers, with the aim of facilitating the task of historians, curators and stakeholders, have focused on the development of new techniques, especially those based on procedural modelling. These new techniques were focused on algorithmically generated 3D models through a set of rules that allow the accurate recreation of a building shape, see Figure 1.2.

However, the majority of the developed techniques intended for building recreation have the same problem; the lack of any structural analysis because they were designed for visual recreation or digital preservation. Thus, the buildings are shown as simple 3D models with textures for giving a realistic view to the final user. Structural analysis would allow to understand certain building features such as how a given building architecture works or how a building was conceived by their designers, among other features. On the Cultural Heritage field, there are some research efforts, especially on masonry structures, where visual results and structural analysis are combined. However, these works are based on customized approaches only affordable for those users with a certain degree of expertise. Thus, non-expert users may have serious difficulties when dealing with these techniques.

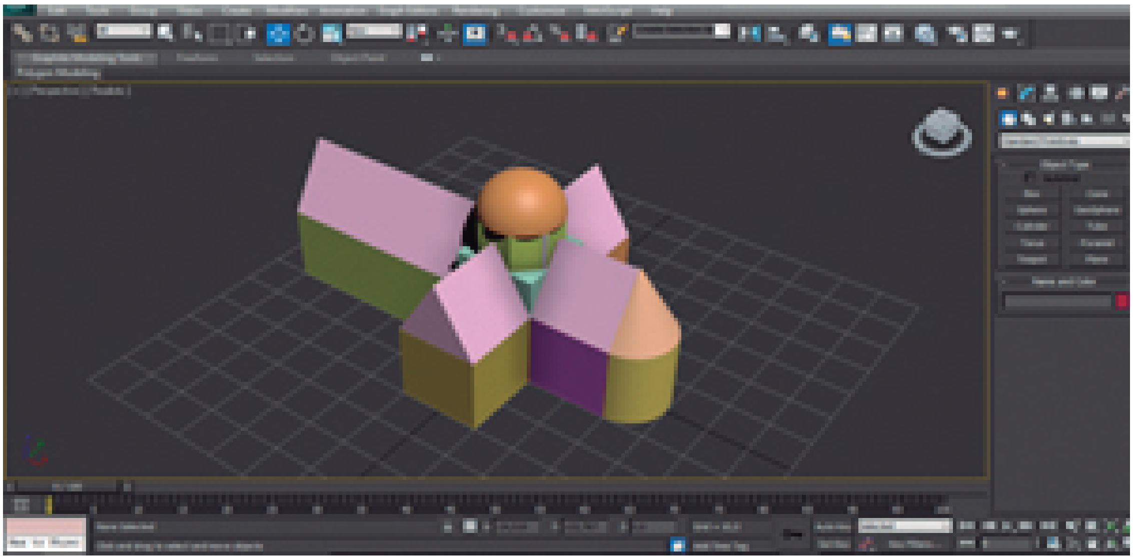


Figure 1.1: A church modelled through solid modelling.

Structural simulation is a powerful tool for analysing the structural degradation of a building or understanding how it was conceived by the architects. An interesting case study is the combination of the subject of study with the simulation of natural phenomena, like an earthquake, whose results can help historians understand past events where a seismic event took place and determined the economic development of the affected area or city. Additionally, on the cultural heritage side, this combination is, until now, nothing more than testimonial.

On the other hand, since the beginning of Virtual Reality (VR), the constant improvements on virtual reality hardware devices and software have allowed the rise of this technology among the general public and the development of applications such as games, serious games and other kind of products for satisfying a growing demand. Thus, Virtual Reality is a suitable technology for the virtual recreation of past events or ancient buildings that has become a core application in Cultural Heritage. Unfortunately, these VR applications have been designed for running on expensive hardware only affordable for people that have the economic resources, or institutions such as universities, foundations or museums. Thus, this is a problem for users who want to deal with Cultural Heritage software and whose low incomes allow them to only afford low-cost devices, which usually are incompatible for rendering applications of high geometric complexity.

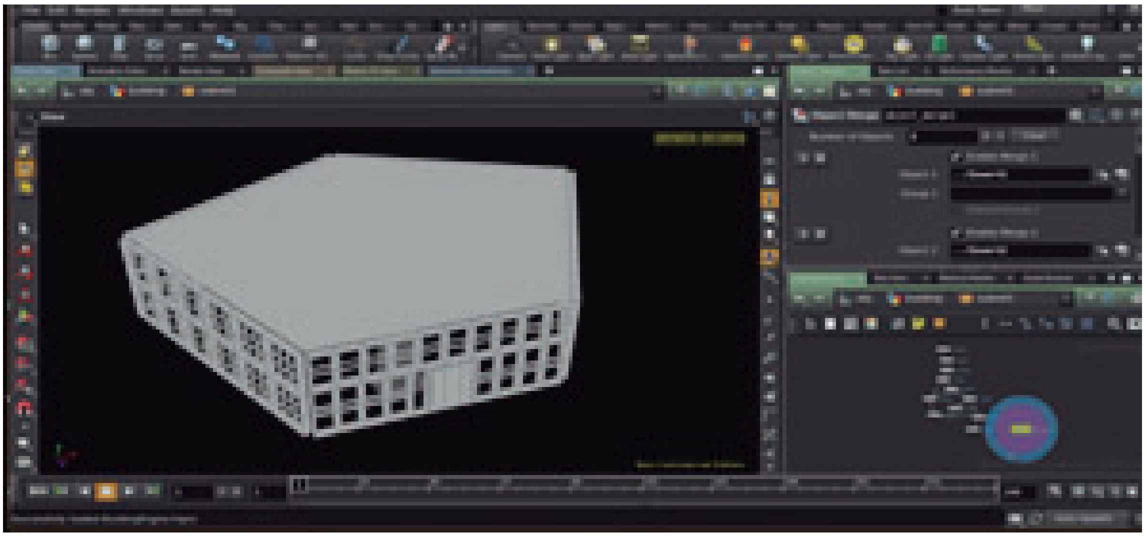


Figure 1.2: A building modelled procedurally.

1.2 Thesis main goals

There are a lot of cultural heritage tools and methodologies for the digital preservation of historical buildings for helping the daily task of users such as historians, art historians, curators and other professional profiles related with this field of science. Most of these methodologies and tools are based on expensive and customized software only affordable for specialized users with a certain level of expertise, or large institutions such as museums or universities.

The goals of this thesis are, first, the development of suitable methodologies and tools for helping the cultural heritage professionals in their daily task about digital preservation of ancient masonry buildings, through non-specialized software affordable for all users at a low-cost. In addiction, through this thesis we want to show the viability of the tools and methodologies designed for this purpose, through off-the-shelf tools and free software, whose management does not need any level of expertise and are affordable for all kind of cultural heritage users and institutions. Thus, the main lines of research in this thesis have focused on:

- *A tool for modelling reliable historical buildings from scratch.*
- *A tool based on the recreation of earthquakes over historical buildings.*
- *A Virtual Reality application affordable for the general public for earthquake recreation over historical buildings.*

1.3 Thesis contributions

The general description about the virtual recreation of buildings in Cultural Heritage and Computer Graphics introduced in the first section of this chapter has mentioned some issues and gaps in the Cultural Heritage field, such as the lack of structural analysis in the use of building modelling techniques, specific customized software for structural simulations, the lack of simulation of natural phenomena in combination with building modelling, and the absence of VR methodologies for structural and earthquake simulation at low-cost for the diffusion of past events.

According to the goals of this thesis, the motivation is to give a satisfactory answer that addresses the described problems and the gaps detected in the literature. Thus, with the use of off-the-shelf tools and free software, we have made the following contributions:

- FITA J., BESUIEVSKY G., PATOW G.: A *Perspective on Procedural Modelling based on Structural Analysis*. Virtual Archeology Review 8, 16 (2017), 44-50:

  This paper presents a survey of the state-of-the-art in modelling and structural analysis techniques used in the Cultural Heritage field. Also, a set of tests were carried out over a procedurally modelled structure for understanding how a masonry building works and what the possibility of structural analysis are. The survey and the test results were also presented in the Archaeologica 2.0 conference, celebrated in Valencia, Spain.

- FITA J., BESUIEVSKY G., PATOW G.: *Tools for Structural Analysis and Optimization of Procedural Masonry Buildings*. In Spanish Computer Graphics Conference (CEIG)(2016), 109-113:

  This paper presents off-the-shelf tools that allow the optimization of a given masonry building, roof and walls, and understanding how it works under a structural simulation. The test results were presented in the Spanish Computer Graphics conference that took place in Salamanca, Spain.

- FITA J., BESUIEVSKY G., PATOW G.: *An Interactive Tool for Modelling Ancient Masonry Buildings*. In Spanish Computer Graphics Conference (CEIG)(2017), 73-76:

  This paper presents a tool that creates a masonry building from a given traditional model, supporting structural analysis. The test results were presented in the Spanish Computer Graphics conference, celebrated in Seville, Spain.

- FITA J., BESUIEVSKY G., PATOW G.: *Earthquake Simulation for Ancient Building Destruction*. In Eurographics Workshop on Graphics and Cultural

Heritage(2018), 141-145:

This paper presents a tool for the recreation of seismic movements based on surface waves. This tool was designed for non-experienced users in seismic simulation. The test results were presented in the Visual Heritage conference that took place in Vienna, Austria.

- FITA J., BESUIEVSKY G., PATOW G.: *Earthquake Simulation Over on Ancient Masonry Buildings*. Submitted (2019):

  This paper is an extension of the paper *Earthquake Simulation for Ancient Building Destruction*. In this paper we compare our test results with an standalone earthquake simulator based on a free software. Also, we recreated and showed the test results of our earthquake simulator based on a real case whose effects affected an Italian church. This paper is under revision at an international journal.

- FITA J., BESUIEVSKY G., PATOW G.: *A Virtual Reality Front-End for Earthquake Simulation*. In Spanish Computer Graphics Conference (CEIG)(2019):

  This paper presents an immersive tool for the recreation of seismic events with low-cost devices. The test results were presented in the Spanish Computer Graphics conference, celebrated in San Sebastian, Spain.

- FITA J., BESUIEVSKY G., PATOW G.: *Seismic Simulation on Virtual Reality*. In Eurographics Workshop on Graphics and Cultural Heritage(2019):

  This paper presents a new test that has been carried out to reinforce the results discussed in the *A Virtual Reality Front-End for Earthquake Simulation* paper. The test results were presented in the Visual Heritage conference that took place in Sarajevo, Bosnia and Herzegovina.

1.4 Organization of the dissertation

The contributions have been distributed in chapters and grouped by the same thematic subject for a better description and comprehension of the contents of this thesis, where modelling and structural analysis solutions are combined for obtaining suitable tools for Cultural Heritage users. We summarize here the different chapters:

- *Chapter 2:* In this chapter, we review the current state-of-the-art about the methodologies and tools available for procedural modelling in Cultural Heritage. Also, we review structural simulation approaches related to procedural modelling of buildings. Additionally, we review how an earthquake occurs and the methodologies available in civil engineering, mainly because its presence in Cultural Heritage is practically testimonial. Finally, we review the current literature about VR methodologies related to Cultural Heritage.

- *Chapter 3:* This chapter presents a methodology designed for modelling historical masonry buildings in any architectural style, it supports structural analysis and is available for cultural heritage users such as historians, curators and stakeholders who want to use it for their research. We tested its viability through a Romanesque church model. Additionally, we present a tool based on brute-force optimization over a masonry building with walls and the unreinforced masonry roof provided by Panozzo et al. [1]. We carried out a set of tests whose results allow to understand how the structural analysis works.

- *Chapter 4:* This chapter introduces an affordable tool designed for the Cultural Heritage field that allows non-expert users in seismology, such as historians and curators, to recreate earthquake simulations. This chapter presents the theoretical basis of the seismic movements. The test simulations have been carried out over ancient masonry structures from the Medieval period such as stone walls and the Romanesque church previously mentioned.

- *Chapter 5:* In this chapter we introduce an affordable approach for those users who deal with our previous solution, for an immersive experience through a masonry structure. The main feature of our VR software is the combination of structural and seismic simulation that is capable of running on low-cost devices and headsets compatible with Google Daydream or Google Cardboard.

- *Chapter 6:* This chapter presents the conclusions of the techniques and tools described on the previous chapters and the future work for improving our techniques based on off-the-shelf tools affordable at low-costs for all kind of Cultural Heritage users.

Chapter 2

Previous Work

2.1 Introduction

Cultural heritage artefacts such as buildings or monuments are constantly exposed to their natural degradation or to their destruction by natural phenomena such as volcanoes or earthquakes. The ancient *Minoan civilization* [2] is an example whose evolution was affected by the eruption of the *Thera volcano*. The Roman city of *Pompeii* [3] is another example where a natural event such as an earthquake and the subsequent *Vesuvius volcano* eruption changed the urban environment and his surroundings forever, see Figure 2.1. Also, the human actions over cultural heritage artefacts are important factors for their destruction. A clear example is the case of the Buddhas of Bamiyan [4], in Afghanistan, that were damaged and totally destroyed by the *Taliban* actions. For these reasons, the digitalization of cultural heritage buildings or monuments and the development of techniques to help this process have become an important research target for Cultural Heritage preservation and diffusion, for Cultural Heritage tourism and for preservation for future generations among professionals. However, virtual recreation of historical environments is not trivial if we want to model an ancient building of a period with historical accuracy.

On one hand, virtual recreation implies knowing topics such as the class and the economy of the society that built the building, the kind and finality of the building, the architectural style of the period, its evolution during the years of construction and the process followed by the builders. These subjects require a multidisciplinary team to answer them all. However, in the literature we can find some authors that have addressed all these topics in their works, particulary about the medieval period, such as Robert A. Scoot [5]; the architecture techniques and the process followed for building churches and cathedrals of this period, as Fitchen [6]; the design evolution of Roman basilicas to Medieval Churches in Western Europe and how the physical forces supported for these

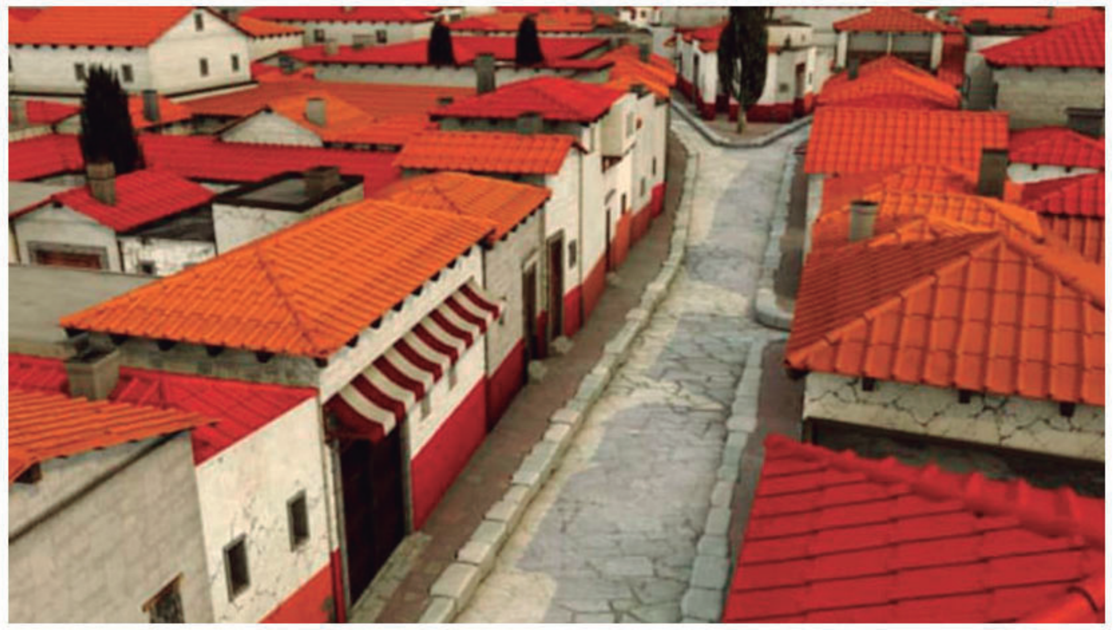


Figure 2.1: A reconstruction of the ancient Pompeii (from City Engine [13]).

historical buildings, as Henry-Claude et al. [7]; or the art and symbolism that exists behind the Gothic Cathedrals architecture, as Ralls [8].

On the other hand, virtual recreation for Cultural Heritage implies solving some previous dilemmas before starting the work of modelling the historical buildings themselves. Thus, an analysis of the techniques and technologies available on the current state-of-the-art in Computer Graphics or Cultural Heritage is needed. In case that these techniques are not suitable, we must design and develop new techniques such that, in any future use, they will make the task of modelling historical buildings, simulation or diffusion of past events in the cultural heritage field at low-cost easier. Additionally, these new techniques must be tested for their viability in the cultural heritage community.

The state-of-the-art presented in this chapter shows the multidisciplinary techniques available and developed in Computer Graphics and related fields. We introduce the main features of solid modelling techniques in Cultural Heritage and an example of how users took advantage for rebuilding an ancient structure. Also, we present the main features of procedural modelling, the relevant works that have contributed to the development of this technique and the main contributions of procedural modelling and structural simulation in Cultural Heritage.

Also, we review the works on seismic simulation over buildings and the building recreations in Virtual Reality that give a satisfactory answer to the demands of cultural heritage users.

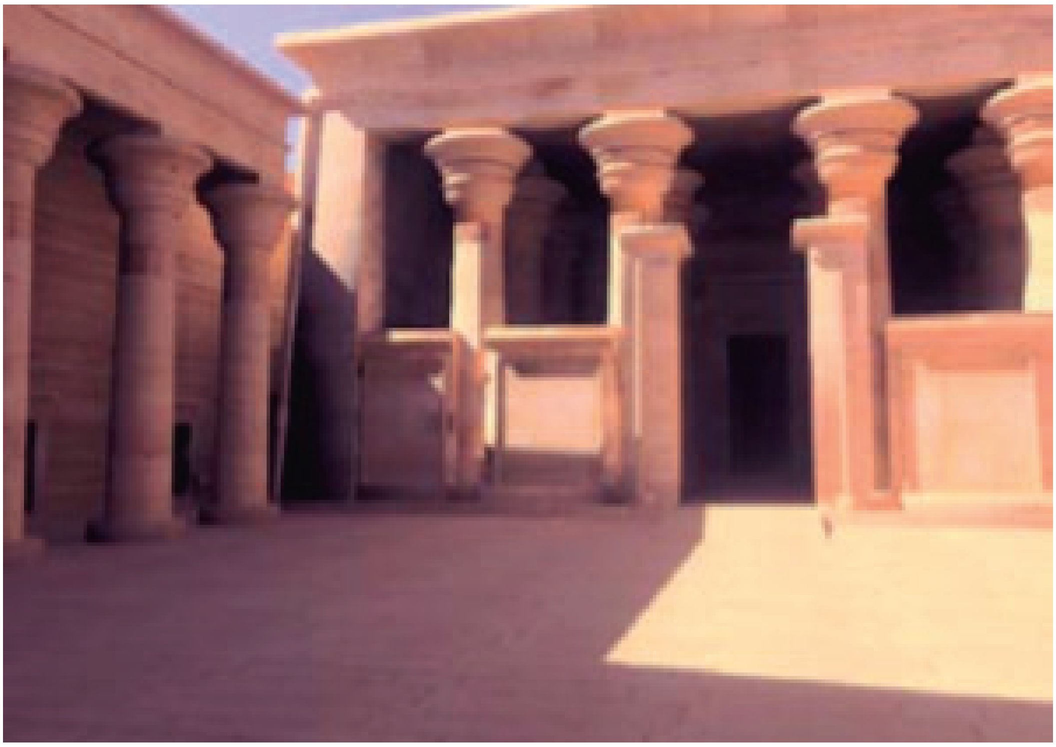


Figure 2.2: The visual representation of the Kalabsha temple generated by solid modelling (from Sundstedt [9]).

2.2 Modelling Techniques

2.2.1 Solid Modelling (SM)

One of the first modelling techniques used for the recreation of ancient buildings in Cultural Heritage is solid geometry, which combines quadric and boolean operations with primitives for modelling a building. A recreation example using solid modelling was developed by the interdisciplinary team lead by Sundstedt and co-workers [9]. In this work, the authors satisfied the archaeologist requests and with high accuracy, creating a digital mock-up of the ancient Egyptian temple of Kalabsha, built of sandstone masonry at the year 30 BC, in their original location and with a realistic illumination that was prevalent in ancient times, as can be seen in Figure 2.2.

However, modelling with this technique presents some disadvantages such as the difficulty of modification or addition of new features to the modelled building and the size of the final file that contains the model. For these and other reasons, some researchers in Computer Graphics have developed new techniques for modelling during the past years, such as procedural modelling.

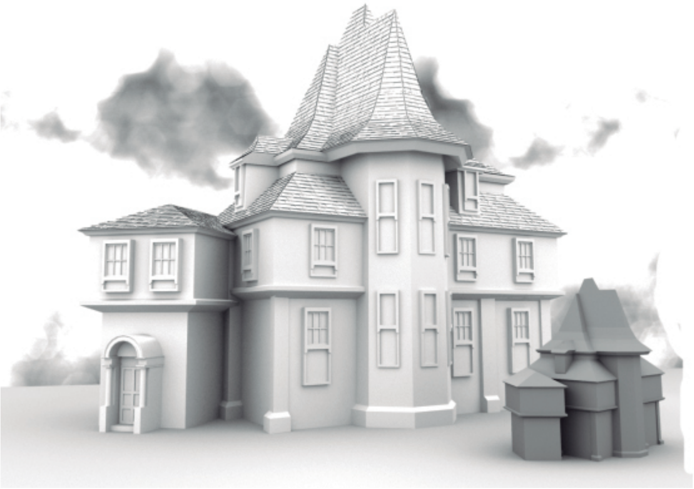


Figure 2.3: The visual representation using an extrusion tool(from Kelly and Wonka [10]).

2.2.2 Procedural Modelling (PM)

Procedural modelling is a technique, usually based on shape grammars, that has a strong advantage in front of other modelling techniques. This advantage is the ease of modification or addition of new features to the building model.

Muller et al. [11] were the first to develop a technique based on shape grammars. This technique starts with a base rule that has a geometric shape of a building associated. Then, the initial shape is iteratively replaced by new shapes for modifying the geometry using rules, see Expression 2.1. This shape grammar allows the users to obtain a great variety of buildings and city models.

$$SeedRule \rightarrow SuccessorRule1 \rightarrow SuccessorRule2 \tag{2.1}$$

Later, this technique based on shape grammars was extended by Kelly and Wonka [10], that introduced the concept of an user interface for carrying out extrusions applied to the geometry of a building, see Figure 2.3. Through this interface, the user is capable of modelling and specify architectural surfaces such as curved roofs, buttresses, vertical walls, chimneys, bay windows and more. Patow [12] presented a visual language for generating a graph of rules for modelling buildings procedurally. Through this visual language, the users can add or edit new graph rules for easily changing the appearance of the building, see Figure 2.4. *Procedural* was a company that developed the *City Engine* software, a visual tool for the building generation. Nowadays, this software is part of ESRI [13], and allows using the shape grammars for modelling all kind of buildings. Krecklau and Kobbelt [14] developed an intuitive procedural modelling tool that provides an interface designed for non-expert users that allows the creation and modification of a

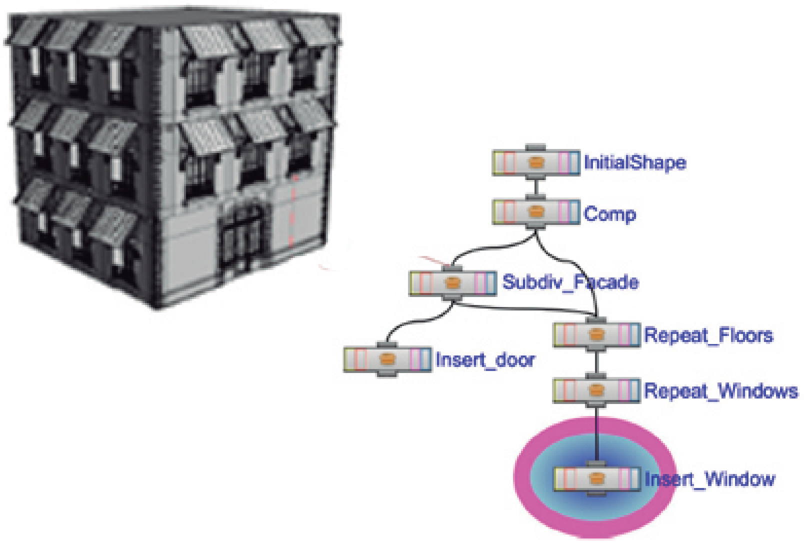


Figure 2.4: The visual representation of the graph generated by a set of procedural modelling rules (from Patow [12]).

building from scratch through a set of high-level primitives designed for this purpose. The authors demonstrated the usability of their technique among inexperienced users. Besuievsky and Patow [15] introduced a technique based on procedural modelling for the recreation of buildings and historical cities through a set of tools tailored for this purpose. For testing their technique, the authors have reconstructed some models based on real examples such as the *city walls of Carcassone* and *the Raccolet House*, both historical monuments located in France. Zhong-Qi and co-authors [16] proposed a modelling process whose main feature is the encapsulation of the structural components of the *Pailou* structures as icons that conform a graph. Through the manipulation of these icons, the users can model from scratch a *Pailou*, a Chinese structure similar to the Roman Triumphal arches. Musialsky and co-workers [17], presented a framework for modelling the façades of buildings through images. The framework is based on coherence-based editing that allows dealing and splitting the symmetries of any façade. Also, the authors have evaluated their approach against other façade modelling tools.

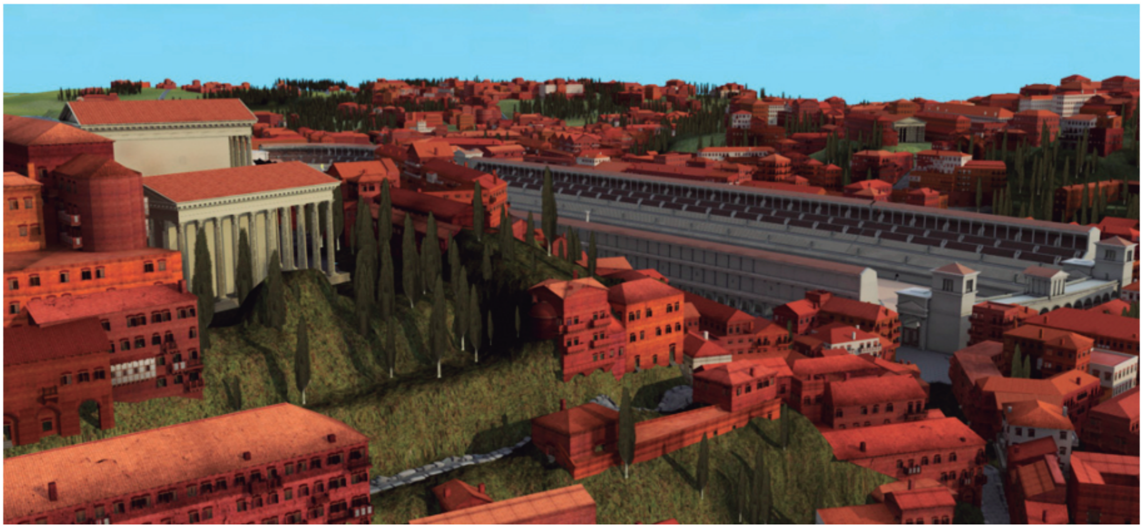


Figure 2.5: A recreation of the ancient Rome (Dylla and co-workers [20]).

2.2.3 Procedural Modelling and Cultural Heritage

Some researchers who developed procedural modelling techniques, have designed methodologies with the aim of showing the applicability of their techniques in the Cultural Heritage field. A first work that validates and applies procedural modelling techniques was presented by Muller et al. [18]. In this work, the authors introduced procedural generation of 3D *Puuc-Style* buildings of the pre-Colombian Mayan architecture for the reconstruction of a Mayan place through Geographical Information Systems (GIS) data from the building footprints or terrain elevation. Years later, Saldaña and Johanson [19] developed a similar technique that involves archaeological GIS data and 3D buildings generated procedurally with off-the-shelf tools. These tools were combined in a game development tool for the virtual reconstruction of two Roman temples and the respective environment of the ancient city of Magnesia.

Dylla and co-workers [20], used a procedural modelling techniques for a virtual reconstruction of the ancient city of Rome at 324 AD. For this purpose, the authors modelled the city in two ways. The first way consists of the reconstruction of the well-known buildings with a high level of detail because their main features, such as location, identification and design, are already known. The second way involves those buildings were the authors only have general knowledge, such as *insulae* buildings, warehouses, and single houses, see Figure 2.5. These buildings have been modelled with the procedural modelling software like *City Engine* [13], were the grammar rules have been designed by archaeological consultants with three levels of detail.

A technique based on procedural volumes designed for users such as curators and historians, whose main feature is the capability of recreating structures in a simple way such as a Roman aqueduct or a Gothic church as *La Sainte-Chapelle du Paris*, was

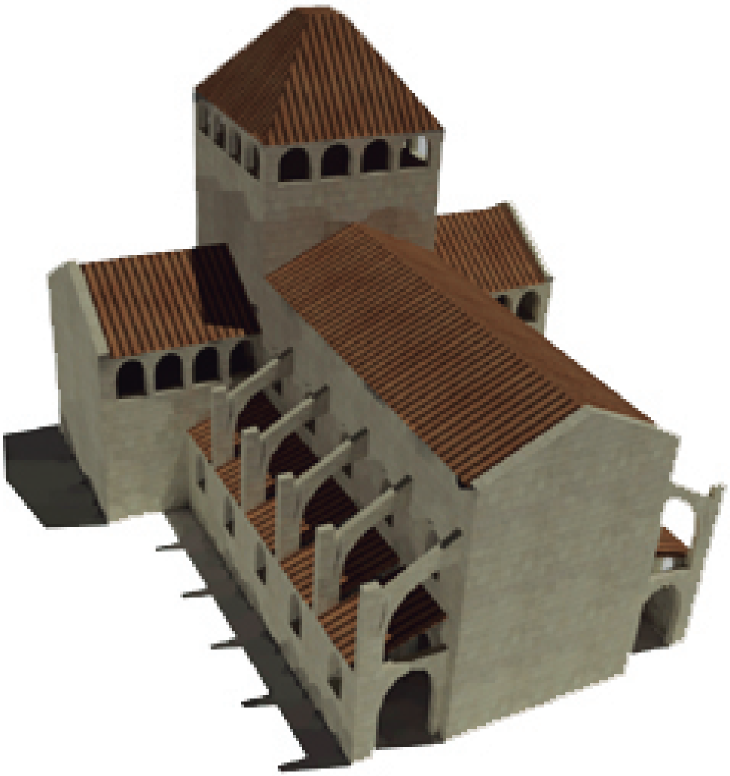


Figure 2.6: A recreation of the Cluny Abbey (from Whiting et al. [24]).

presented by Barroso and Patow [21]. Koehl and Roussel [22] presented a procedural modelling technique based on the reconstruction of the church of Turckheim, France. Through this procedural technique, the authors could make reconstruction hypothesis about the 12^{th} century chapel placed in the same terrain. Capellini and co-workers [23] developed a custom procedural technique based on the construction process followed in Roman times, which is capable of reconstructing 3D masonry structures from photogrammetric data.

However, all of the mentioned techniques have a common problem: that they have been designed only for the shape recreation of ancient buildings, without any combination with structural analysis. Thus, the users of these techniques have a precise shape of a certain building, but they can not know how the structure of the modelled buildings works.

2.3 Structural Simulation and Cultural Heritage

Structural stability of buildings has become a case study of some importance in Computer Graphics. Through structural simulations, researchers and users may know how an ancient structure works, especially masonry buildings such as churches and cathedrals, which were designed by the ancient architects for supporting vertical loads. Thus, this

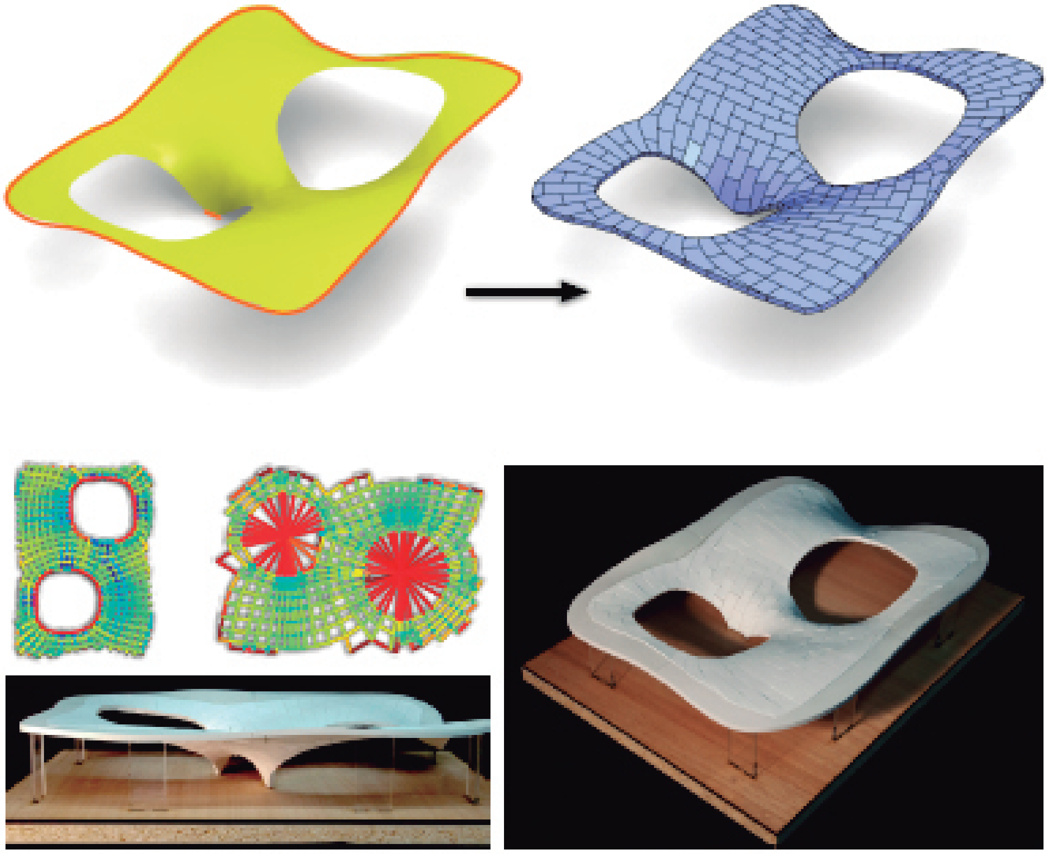


Figure 2.7: A new masonry shape is created, above. Then this structure is printed and assembled, below. (from Panozzo et al. [1]).

section focuses on those techniques focused on masonry buildings, especially those which combine procedural modelling techniques and structural analysis related to the cultural heritage field.

Whiting et al. [24] describe a procedural modelling technique based on stone masonry buildings such as churches and cathedrals, see Figure 2.6. This technique gives the non-experienced user the chance of modelling a masonry building through a set of text-based rules. Once the user has finished the creation process, the structural stability of the building is automatically computed by the calculation of quadratic equations. Later, Whiting and co-workers. [25] developed a method that allows the users to improve their masonry building design or the load of cable-stayed structures. This technique is capable of analysing the geometry of a masonry building, previously designed, through its vertex coordinates, and create a new stable structure according to the restrictions previously introduced by the user, such as block thickness.

Panozzo and co-authors [1] introduced an algorithm designed for those users who

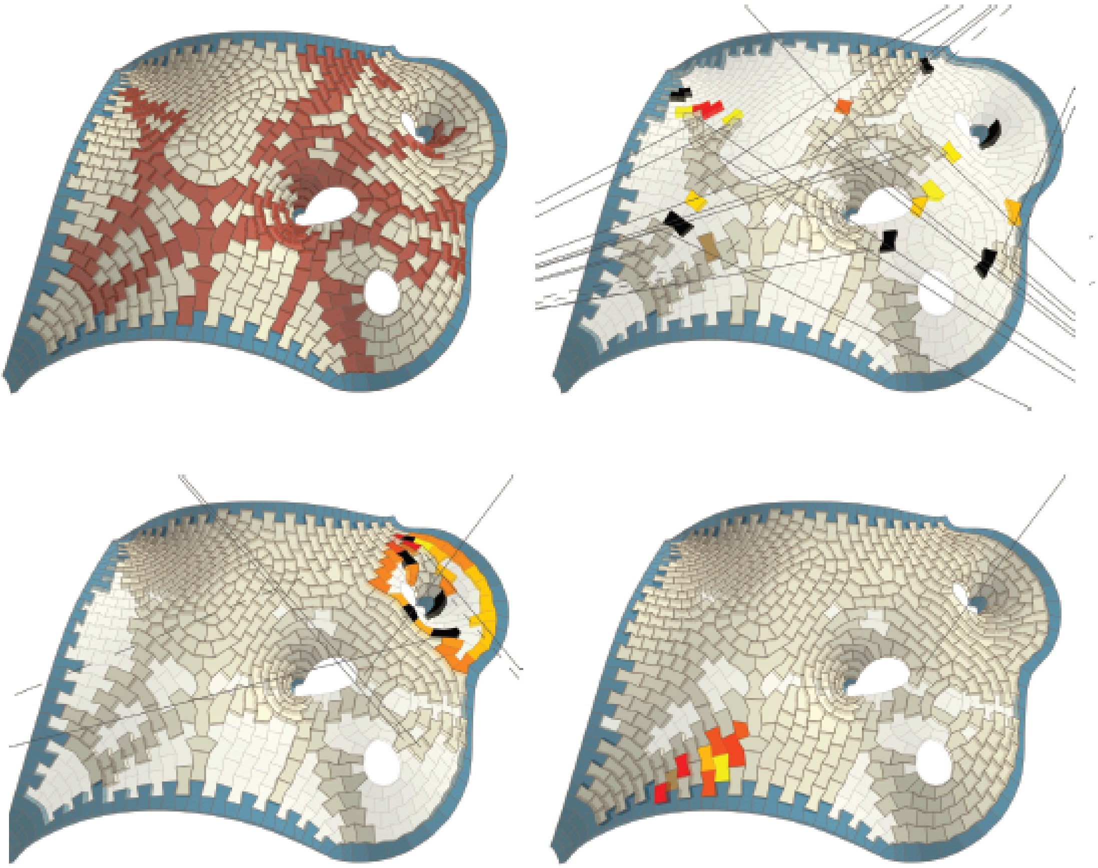


Figure 2.8: The detection of quasi-arches of a given building shape, above. Then the sequence generation, below. (from Deuss et al. [26]).

have no previous structural knowledge and that is capable of generating an unreinforced 3D masonry structure from a Non Uniform Rational B-Splines (NURBS) surface, as close as possible to the input shape. The main feature of the generated masonry structure is the capability of supporting itself without the combination of an extra glue component. Additionally, the authors printed and assembled a 3D masonry structure with the aim of validating the methodology, see Figure 2.7. Following the steps initiated by Panozzo et al., Deuss and co-workers [26] extended the research about self-supporting structures and generated a new algorithm based on the detection of quasi-arches of a given building shape. In their approach, a quasi-arch is a component that connects two blocks. This allows to model all kind of generated masonry buildings through a set of temporary chains and hooks for assuring the structure stability during the construction process, see Figure2.8.

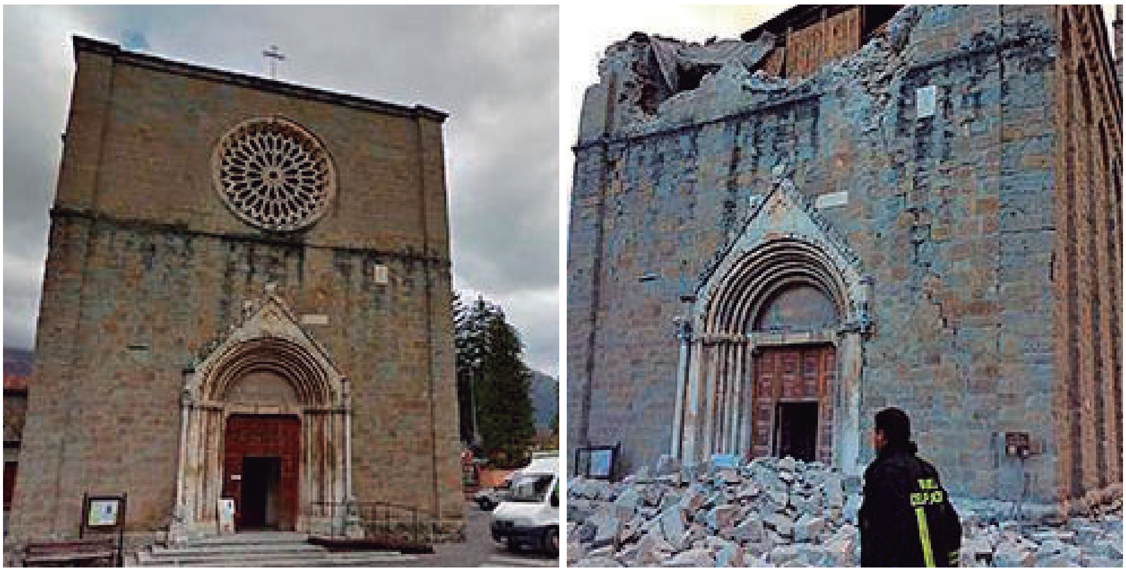


Figure 2.9: The St.Agostino church before and after the earthquake (from Wikipedia [27]).

2.4 Seismology

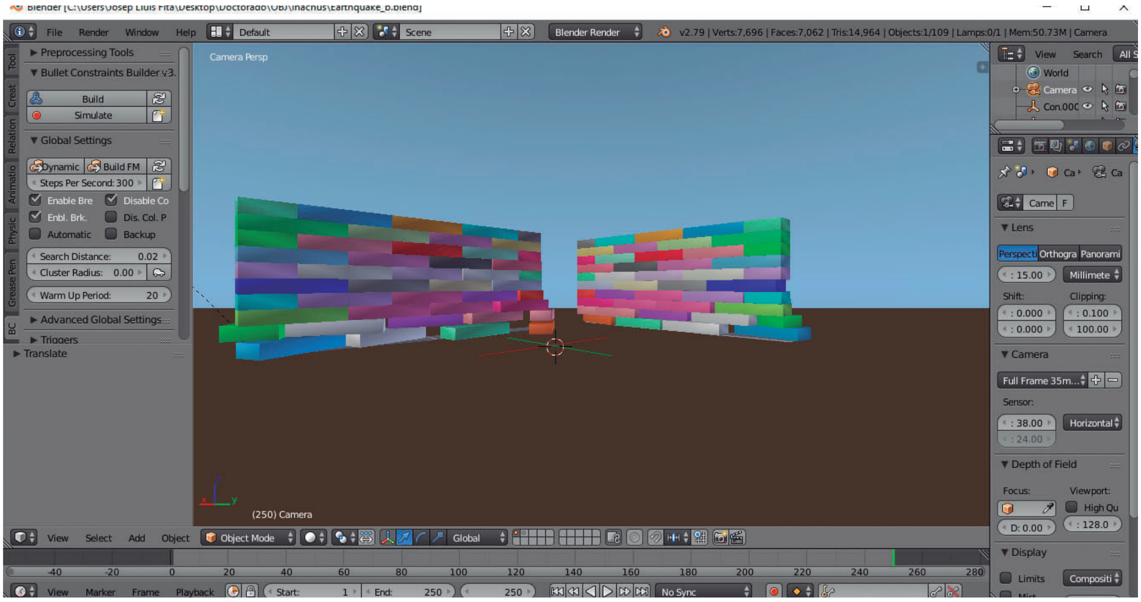
2.4.1 Seismology Theory

Historically, we can find a lot of cases where natural phenomena like volcanoes and earthquakes have been involved in the destruction of all kind of masonry buildings. We can mention, as an example, the earthquake happened in the year 2016 in the centre of the Italic peninsula with a Richter magnitude of 6 and which affected villages and towns such as *Accuamoli*, *Pescara del Tronto* and *Amatrice*, that were severely destroyed [28], as well as all their cultural heritage buildings, such as the church of *St. Agostino* of *Amatrice*, see Figure 2.9.

However, earthquake simulation is not a trivial subject. If we want to model this natural phenomenon as realistic as possible, we should know how a seismic event occurs for simulating it accurately. Nowadays, online resources such as U.S. Geological Survey [29] give all kind of information about earthquakes and the historical events where earthquakes were involved. However, we must appeal to other scientific sources [30, 31] for a deeper theoretical and mathematical description if we want to model the seismic movements in our simulations.

2.4.2 Software and Earthquake simulation

Among the literature about software designed for simulating seismic movements, we can find the ETABS Software [32], conceived for civil engineers and developed by an



Steps Per Second: 300
☑ Enable Bre
☑ Disable Co
☑ Enbl. Brk.
□ Dis. Col. P
□ Automatic
□ Backup
Search Distance: 0.02
Cluster Radius: 0.00
Warm Up Period: 20
Focal Length: 15.00
Shift: 0.000, 0.000
Clipping: 0.100, 100.00
Sensor: 38.00, 24.00
D: 0.00
128.0
Start: 1
End: 250
250

Figure 2.10: The INACHUS plug-in interface for earthquake simulation.

American enterprise. This software was designed for modelling all kind of structures for carrying out structural analysis. Also, this software has an option for seismic simulation.

We can also mention the INACHUS project [33]. This project is a software platform developed by an European consortium that involves privative and public entities such as the Netherlander University of Twente [34]. This project provides a free plug-in tool called *Constraint Builder* based on seismic simulation and designed to be managed for non-experienced users on the Blender software suite [35] and the physics engine Bullet [36]. Through this plug-in, the users can configure seismic features such as the time duration, the frequency, the wave direction and the earthquake magnitude. See Figure 2.10.

2.4.3 Earthquake Simulation and Cultural Heritage

Seismic simulation has become a case study of great interest in Civil Engineering for giving effective solutions to the inhabitants of high-intensity seismic areas. However, on the cultural heritage side, the structural and seismic simulation over buildings is barely treated.

On the literature about seismic simulation and Civil Engineering, we can find some works where historical masonry buildings or structures are involved. Some civil engineering researchers introduced a technique where the main objective is the analysis of the historical structures under seismic conditions. Altunisik et al. [37], examined the structural response of the Zaganos Bastion, in Turkey, through a Finite Element model combining experimental and numerical methods for obtaining the seismic analysis of the structure with satisfactory results. Castori and co-workers [38] analysed through three different models such as Equivalent Frame, Rigid Macro-Blocks and Finite Elements, the civic museum building of Sansepolcro, Italy, whose results highlighted some structural problems derived from the construction. Fortunato and co-authors [39] took real data through optical 3D measurements for modelling the baptistery of the San Giovanni church, Italy. Then, the modelled building was structurally analysed under seismic conditions for obtaining its structural response. Finally, Souami and colleagues [40], analysed the model of Algiers under seismic movements for detecting the vulnerabilities of the old buildings, specifically the buildings of the 1830-1930 period.

Since antiquity, the construction of masonry structures combined with timber has become a common way for reinforcing the buildings in those areas that usually have seismic movements. Kapos and Kouris [41] introduced and validated the results obtained by about non-linear static analysis over modelled structures based on timber framed masonry structures, whose behaviour dissipated the seismic energy efficiently. To reinforce their results, the authors applied the tests over a modelled façade of a real building located in the Ionian island, Greece. Mosoarca and Gioncu [42], developed a methodology based on modelling failures through blocks, focusing on orthodox churches where the test results predicted the failures of the modelled buildings under seismic movements. The authors compared the failure results obtained from the tests with a real case that affected the orthodox church of St. George-Birda, located in a region of Romania, for validating their study. Bosiljkov et al. [43], have tested different historical buildings from Slovenia through different evaluation methodologies for masonry buildings such as Non-Destructive tests, Minor-Destructive-Tests and Destructive-Tests, with the aim of knowing the structural behaviour of these structures. Ramos and Lourenco [44] developed a methodology based on finite elements and non-linear behaviour of materials with the aim of analysing the buildings of the 18th century located in the city centre of Lisbon under seismic movements. The authors carried out a set of tests that confirm the building failures observed on the terrain. Because the seismic simulation is not trivial, we would like to mention the research carried out by Uphoff and co-authors [45], whose main feature is the representation of the dynamic rupture of a detailed topography of the fault system that generates the Sumatra earthquake of 2004. The simulation shows the sea-floor surface displacements and allows to determine the initial conditions for the tsunami generation and its respective propagation.

Real scene
Virtual human
Virtual human
Real scene 'occludes' virtual human
MIRALab - University of Geneva
MIRALab - University of Geneva

Figure 2.11: A recreation of the Pompeii City (from Magnenat-Thalmann and Papagiannakis [46]).

2.5 Virtual Reality

2.5.1 Virtual Reality and locomotion

We can define Virtual Reality as a technology that provides the users an immersive experience in a 3D virtual environment, whose appearance can be realistic or synthetic, and that allows the interaction among users and the virtual environment. Walking on space is a well-known technique of locomotion for virtual environments that has a great benefit for the users because it gives the feeling that they are walking on the virtual world-ground, and allows reducing motion sickness. However, walking around has an issue because real space, in the majority of the cases, is smaller than the virtual environment. With the aim of correcting this issue, some researchers have developed some other motion techniques, such as redirect walking. This technique presented by Razzaque and co-workers [47] allows the user to walk large virtual scenes by rotating around the user position the virtual environment.

Teleportation is another technique of locomotion that allows to avoid motion sickness. A new locomotion technique based on Point and Teleport was developed by Bozgeyikli et al.[48]. This technique allows an easy displacement of the users by pointing to the desired place and the consequent teleportation. Also, in this work the authors compared their navigation technique with the other navigation techniques such as walk-in-place and joystick for Virtual Reality. Additionally, teleportation does not need extra hardware for being implemented. Thus, this is the most suitable locomotion technique for mobile devices such as smart-phones.

Figure 2.12: A spherical view of the Sagrada Family Cathedral environment (from Andujar et al. [52]).

2.5.2 Virtual Reality and Cultural Heritage

The main purpose of virtual reality technology applied to Cultural Heritage is the recreation of past events or realistic environments for giving an immersive experience when users interact with the objects in the virtual scenario.

Among the methods developed, we can find the VR/AR work of Magnenat-Thalmann and Papagiannakis [46], where they developed on a virtual model of Pompeii, the daily life recreation of their inhabitants, designed to be visualized on wearable devices. See Figure 2.11. Gaugne and co-authors [49] developed a methodology based on Virtual Reality for giving immersive experiences to cultural heritage users that allows them to interact with recreated 3D artefacts as user interfaces. Also, the authors based the development on various interpretations of an Iron Age urn discovered in the United Kingdom in the year of 1910 of the past century. Selmanovic and co-workers [50], introduced the study of user experience given by 360^0 Virtual Reality (VR) videos about the diving tradition of the Mostar Old Bridge, Bosnia Herzegovina. A Virtual Reality recreation of a church was modelled through photogrammetry and 3D techniques, because this historical building, Santa Maria Paganica of L'Aquila, located in Italy, suffered several damages in its structure by the 2009 earthquake. The virtual recreation of the building was assembled in a game tool, and the developers tested the interaction through two different VR desktop devices. This work was presented by De Gasperis and co-authors [51].

Andujar et al. [52] presented a VR-based project whose technology is a combination of 3D Computer-Aided-Design (CAD) tools, for modelling new parts of the building; and

VR tools, for the visualization of the Sagrada Family Cathedral existing parts, together with new ones designed for the cathedral completion, see Figure 2.12. X-Reality (XR) can be defined as an integration of Augmented, Mixed and Virtual Reality technologies. Carrozzino et al. [53] presented a study based on the comparation and classification, according to features such as Interativity and Usability among others, of different XR technologies. For more information, please refer to the survey by Bekele and co-authors [54] and the review presented by Mortara and colleagues [55].

2.6 The Seville principles

Virtual Archaeology is a scientific discipline that manages the virtual recreation of cultural heritage based on the London Charter and the Seville principles. According to the Seville principles for computer-based visualization of cultural heritage, we can mention these principles as:

- Interdisciplinary
- Purpose
- Complementarity
- Authenticity
- Historical Rigour
- Efficiency
- Scientific Transparency

The techniques presented in this thesis are aligned with the Seville principles given in the list above. These can be deployed by an *interdisciplinary* team of professionals such as historians, archaeologists, curators or other professionals that work in the cultural heritage field. Also, each methodology has a *purpose* or finality that can be *complementary* among these or other diffusion materials. The *efficiency* principle refers to obtain good visual results of the rendering, applying fewer resources. Thus, this thesis aims at presenting efficient methodologies based on low-cost recreation and diffusion of ancient masonry buildings and their environments to the interdisciplinary teams that develop Cultural Heritage projects. However, the level of *authenticity*, the *historical rigour* and *scientific transparency* in the building recreations depend on the team that follows our methods.

Chapter 3

Tools for Modelling and Structural Analysis

3.1 Introduction

In the middle ages, when a church was built by masons, especially when they were finishing its last stages, there was one particularly critical stage that had to be carefully planned and executed. We refer to the construction of the vault, where the habit of the sculptors of making each block of stone have the exact size that fitted in the final placement of the vault, resulted in complex production procedures, generally based on trial and error. This allowed to build the vaults without mortar because the vault self-supported its weight through friction. Thus, with the aim of learning about historical masonry buildings, we have analysed those techniques based on cultural heritage literature that combines building modelling and structural analysis. The analysis of these techniques allows to understand how these buildings were designed and conceived by their builders. However, most of these analysis techniques are based on customized solutions that the majority of the cultural heritage users cannot afford because their management is technically difficult. With the use of off-the-shelf tools, we have developed low-cost techniques designed for those cultural heritage users, and we show its viability. In section 3.2, we present a tool for modelling masonry buildings through procedural modelling techniques, which was chosen because it has the advantage of easy adaptability of the features of the modelled buildings. A first case study about procedural modelling and structural analysis is introduced in Section 3.3. In this analysis, we modelled through procedural techniques the walls that support the vault created by Panozzo et al. [1], where each stone was designed with the most suitable size for its corresponding place, such as was done in ancient buildings by ancient masons. In Section 3.4, a study of structural analysis based on brute-force optimization of masonry structures is introduced.

In our work we used off-the-shelf tools such as *Houdini* from SideFX [56], the open source *Bullet* solver [36] (incorporated by default in Houdini, along with other solvers), and custom *Python* scripts, with the aim of giving a low-cost tool affordable for all kind of cultural heritage users or professionals. This allows the virtual recreation of all kind of masonry buildings and structures, such as walls and churches that resemble the Romanesque, or medieval periods, although we can use this methodology for the virtual recreation of the architecture of any other period, such as classical architecture.

3.2 Interactive Modelling for Masonry Buildings

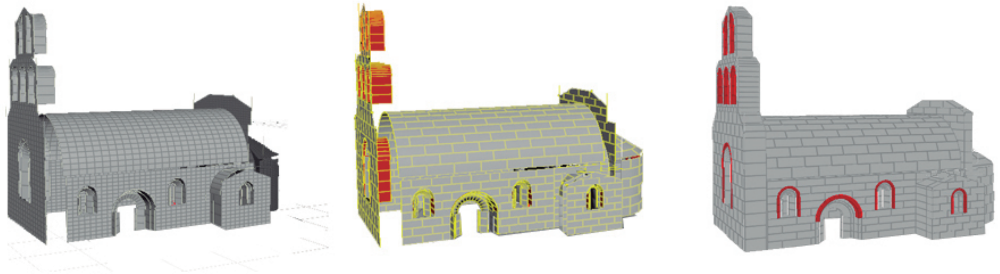


Figure 3.1: An input geometry of a shape mesh without volume like a church (left), (e.g., the belfry or the vault), is transformed in a mesh of stone bricks (middle). Then, the building with its physical features is ready for structural simulation (right).

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, see Figure 3.1. We show our pipeline at Figure 3.2. In the following sections we will present this process step by step, starting from the original shape, transforming it into a set of masonry bricks, and ending in the simulation itself.

3.2.1 Input Geometry

We assume that users start with a modeling tool, which can be either a traditional one such as Autodesk's Maya, Autodesk's 3DSMax or Blender, or a procedural tool such as Houdini, and design the building basic structure as a required mesh without volume. This model can be made of any kind of surface, from Non Uniform Rational B-Splines (NURBS) to simple polygonal representations. Our methodology does not support inputs based on scanned 3D models because it has been designed for working with single shape meshes,

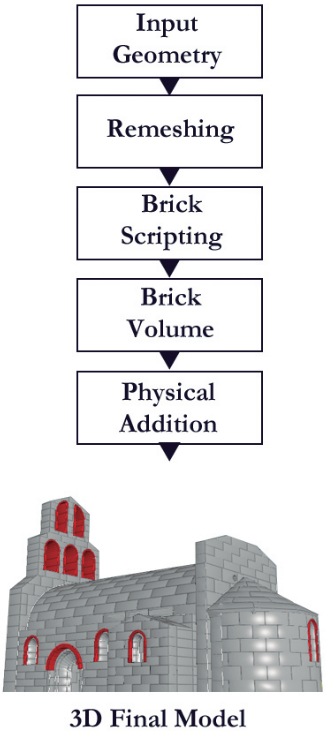


Figure 3.2: The Procedural Modelling Tool Pipeline.

(e.g., a canon vault or a single wall). Once users have finished this design stage, they can use our technique for easily transforming the created building into a masonry structure made of bricks and ready for structural studies.

3.2.2 Re-meshing

When users design a building, they usually focus on its shape and they do not think about the final masonry building, which is the main objective of our tool. The first step towards our model transformation is to convert the input shape based on a mesh without volume into a suitable mesh of quads, because this kind of mesh facilitates brick modelling. 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-tessellated surface. Once the input surface has been converted to a regular quad mesh, we need to re-mesh it again for generating a mesh of quads of low resolution suitable for our masonry model.

For the re-meshing 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

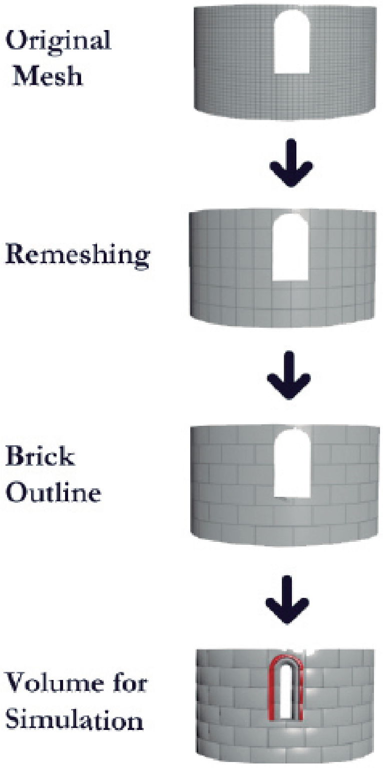


Figure 3.3: Starting with an original mesh shape, the process meshes and makes the brick outlines over the given mesh. Finally, it adds the volumes for structural simulation.

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 re-meshed the entire mesh. We can see a resulting re-meshed model at Figure 3.3 and the process on Figure 3.4.

However, our re-meshing technique is designed for supporting a fixed aspect ratio for the bricks of a wall. Thus, if we want to have in the same wall different aspect ratios for the bricks, we should divide this wall as two or three parts, and give for each part the respective brick aspect ratio. This procedure does not represent any limitation for recreating a building and has a low impact on the simulation results obtained with the tested model, whose belfry was modelled following this technique, see Section 3.2.5.

3.2.3 Brick shape

Once the re-meshing 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

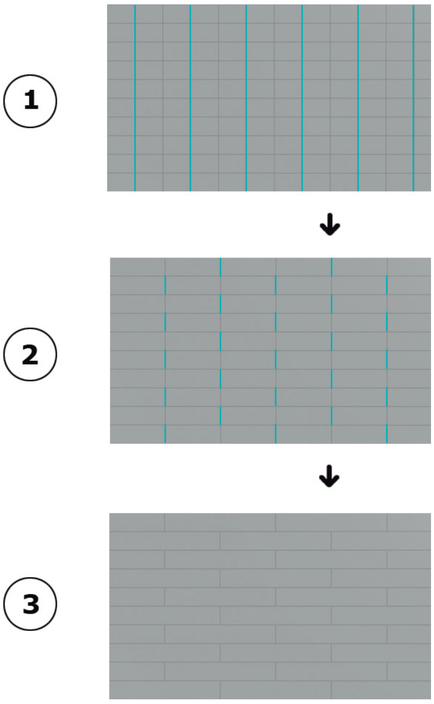


Figure 3.4: Brick pattern sequence creation; First, the re-meshing process selects and delete the edges of the mesh (cyan), for obtaining the brick size given previously by the user. Second, the method selects those edges (cyan) for obtaining the brick pattern. Third, the brick pattern is ready for receiving the volume and the physics features.

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 of the quad mesh. Then, the procedure iterates through all the edges of the loaded mesh, generating and saving a list of edges that contain those that have only two neighbouring 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 architectural structures 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. 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, see the second and third step in Figure 3.4.

This brick outline has been designed to reproduce the shape of the bricks without the infills of mortar because the stone buildings, churches or cathedrals, in the middle ages, were built practially with the minimum amount of mortar between stones, see Figure 3.5.

Furthermore, the brick outline in our implementation has a mason pattern for the

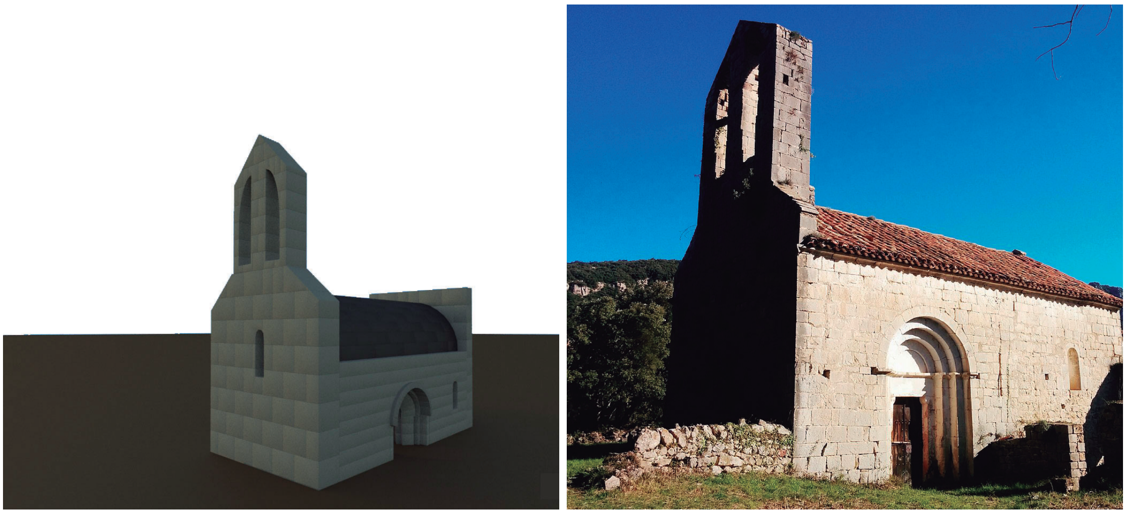


Figure 3.5: On the left, the 3D recreation inspired on 12th-century Romanesque church, *Sant Bartomeu de Pincaro*. On the right, the real church located in the village of Albanya [57]. In this picture, we can see the stone brick pattern of the church and how its walls were built with the minimum amount of mortar. Also, we can see the capability of our tool for the recreation of accurate models in comparison with reality.

walls that has the same size for the entire structure, in contrast with the real pattern shown in Figure 3.5, where we can see, in the same wall, different heights between the rows of stone bricks.

3.2.4 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 (in meters) 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. These bricks are created without irregularities or any other deficiencies.

The tool allows adding physical properties to the model, corresponding to an ancient building such as the density of the stone or other materials, the friction coefficients, and the environmental temperature, among others. The amount of mortar is assigned by the user through the glue component that affects the connection strengh among all the bricks. When our tool has finished these steps, the 3D model of the building is ready for structural simulation. See Figure 3.5. Also, we can see the capability of our tool for the recreation of accurate models in comparison with reality. We must remark that this version of our tool does not take into account a layer of plaster over the stone walls.

On the other hand, we have described our model based on perfect masons without any degradation process based on time. However, this current model, through the friction and stiffness parameters for the stone walls, will be able to reproduce the effects of these irregularities by lowering the parameter values. But, it is not sufficient for reproducing these effects in the simulations accurately. For this purpose, we should add geometric irregularities to the exterior surface of the stone walls, according to the friction and stiffness parameters, and the weather conditions such as wind or rain for reflecting these irregularities through scripts. Nevertheless, we will obtain simulation results as realistic as we have obtained without the application of this degradation process over the walls as its impact is minor considering the other aspects taken into account by our simulation.

3.2.5 Results

We have made a 3D prototype of an old 11th century Romanesque church, *Santa Maria de Agullana*. We have designed a set of tests for verifying the feasibility of our interactive tool. The software platform we used for the implementation has been described in Section 3.1. The physical features for our test are: 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 $9000 Nm/rad$, the roof of the apse has the same physical values as the vault roof. The glue value is low, 10, which is ideal for simulating the strenght of the ancient mortar used in the middle ages. For the vault, the glue value is 0. As a result, we can see on Figure 3.6 that our generated model satisfies the soundness with its current design the 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.

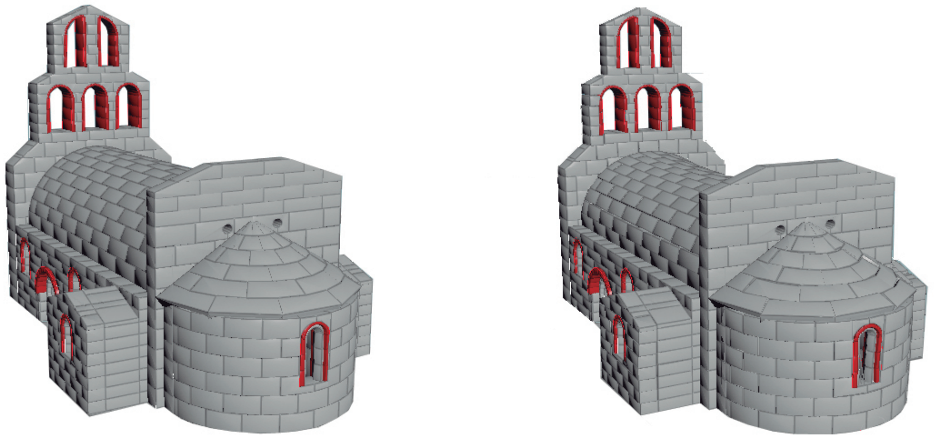


Figure 3.6: Our model, before (left) and after (right) the structural simulation test. Observe how the roof after the simulation is not a perfectly cylindrical shape anymore, but it has got a slight saddle shape.

3.3 Structural Stability: Direct Analysis

This section presents the study that has been carried out over a masonry structure. In this study we perform a simulation with the purpose of addressing the main features of structural simulation.

3.3.1 A general view of Structural Simulation

For carrying out the structural tests, the geometry of the recreated buildings and the ground must be connected through a dynamic network that allows configuring the linked bricks that conform the buildings to the ground. The dynamic network is the component that acts as a connector between the geometry and the physics engine. Through it, the physics engine process all the geometry such as buildings and ground as rigid bodies with features such as *density* and *friction*. The *density* is used by the solver to calculate the object's mass. Also, the geometry has the *collision shape*, which is the object volume used to detect collisions with other objects in the scene (i.e., the ground or other bricks). The physics engine is designed for performing the collision detection, resolving these collisions and their constraints, and finally performing the transformation of all objects affected by the simulation.

When a simulation starts, the physics engine creates the dynamic world with its rules, such as the gravity force, adds the physical objects such as the walls and the ground, and then performs the simulation with the following steps for each frame:

1. **Broad-Phase Collision Detection**: The physic simulator generates a list of possible pair of objects that could collide using *Axis-Aligned Bounding Boxes (AABBs)*. This step is performed automatically by the library without the need of fine tuning on our side.

2. **Narrow-Phase Collision Detection**: If two objects were candidates for a collision in the previous stage, this stage verifies the collision using the box geometries for each brick.

3. **Constraints**: The physic simulator updates a list with the new positions of the ground, for limiting the brick movement.

4. **Modifies Physics Parameters**: For each brick, the library checks whether the position of the brick should be updated with the previous list (i.e., the lower wall bricks, directly linked to the ground), or if its position needs to be updated as a result of a collision with another element (e.g., other brick). The final positions are used for rendering.

5. **Loop Rigid Body**: Return to Step 1 until the physic simulation finishes.

When the simulation has finished, then the physics simulator deletes the objects and the dynamic world created in the physics simulation process.

3.3.2 Testing the Vault

To perform the structural test, we must first build the geometry that supports the masonry vault. For this purpose, we built three supporting walls of *voussoirs* (blocks), each with a density of $2691 kg/m^3$, which corresponds to granite stone. Then, we gave a width of 2 m to the walls with the aim of building a structure resembling a Romanesque church. Although rarely necessary, a very fine, almost non-existent, layer of mortar was usually added simply as a security measure to the final stability of the building. Thus, we have added a bit of mortar, called *glue* from here on, as it is called the internal simulation parameter that simulates the mortar used between the blocks of stone. Given this basic structure and settings, we placed the same vault geometry used by Panozzo et al. [1], see Figure 3.7, and we gave it a density value of $2651 kg/m^3$ which simulates a variant of stone granite. In this case, the *voussoirs* were added without mortar, following Panozzo et al.'s specifications, see Figure 3.8. Once we run the simulation with this basic set of parameters, we can observe that the vault collapses, as shown in Figure 3.8. In order to improve stability, we adjusted (increased) the value of the so called *rotational stiffness* for the vault until the vault holds in the simulation. A higher value for this parameter would make the objects less liable to spinning; a lower value will make them more ready to spin. Basically, it is a scaling factor applied to the inertia tensor, lowering its rotational freedom.

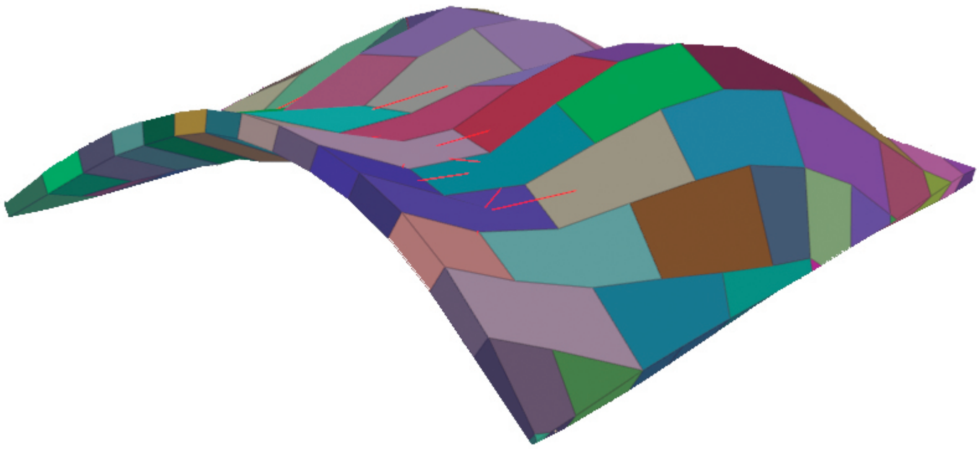


Figure 3.7: The unreinforced masonry model used as a roof (from Panozzo et al. [1]).

At this point, we begin reducing the width of the walls until 1.5m in an iterative way. We found out that the vault supports its weight with the new wall parameters, but we observed a few cracks (in our case, represented by some loose stones) on the walls. With these results, we continued decreasing the wall width with the aim of finding the minimum width that supports the entire structure. In this case, a value of 1.2m was found where the number of the cracks on the walls largely increased, as expected. Beyond this point, if we continued with the reduction of the width value of walls, the whole structure would fall. For this reason, we have decided to add six buttresses to the entire structure, as shown in Figure 3.9.

After adding the buttresses, we observed an improved stability: the vault now supported its own weight and the walls did not show any damage in their structure. At this point, we decided to reduce the width of our walls again, until 0.5 m. With these wall parameters, we still obtained good results for the stability of the structure (i.e., the vault and the walls).

As a consequence of our tests, we have a solid perspective on how the whole simulation works, specifically for the selected case of a vault over some walls. Then, we designed a new static test with the aim of improving the minimum number of buttresses per wall. Also, we incremented the wall buttress width to 1 m. Again, we observed severe instability and finally, the total collapse of the structure. Finally, during our last test, we added some glue (i.e., mortar) to our vault structure, in spite of the fact that this kind of vault had been designed to work without any mortar. We ran the simulation again and

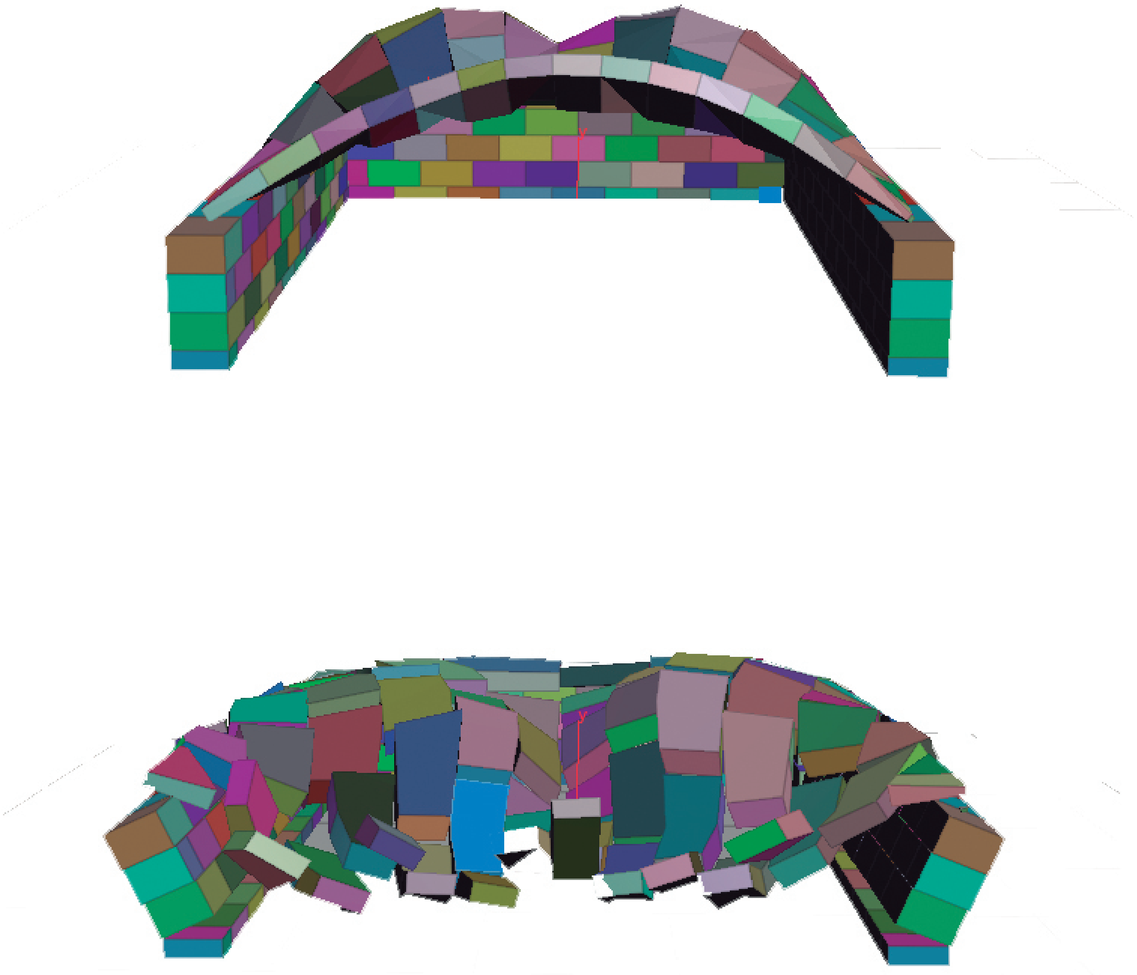


Figure 3.8: Above, our vault over the walls that simulate a Romanesque church. Below, our vault has fallen down during a test with the basic set of parameters.

the obtained results were similar to what happened to the last test, a building collapse. The reason is, probably, that the rotational stiffness plays a more important role than any mortar, as it prevents the *voussoirs* from rotating around their gravitational center of mass.

3.3.3 Analysis of This Test

We observe from the state-of-the-art in Section 2.3 the works that successfully merge procedural modelling and structural analysis on masonry buildings are, at most, scarce. Also, it can be seen that all of the presented techniques (i.e., Whiting et al. [24]; Whiting et al. [25]; Panozzo et al. [1]; Deuss et al. [26]) rely on custom, specifically tailored solutions that involve expensive development efforts to produce the required code. On

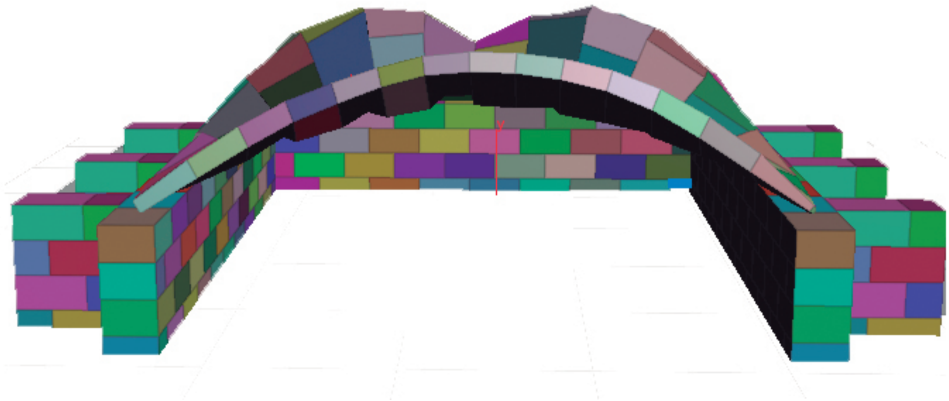


Figure 3.9: Our structure with buttresses.

the other side, our implementation relies on standard, off-the-shelf tools that are freely available and, quite probably, already part of the modeler toolbox. In spite of the lack of detailed studies, we consider that structural studies are a key element for achieving realistic masonry structures, in particular for both churches and cathedrals, and that their importance and usefulness cannot be denied in this context. Thus, we analyzed currently available, off-the-shelf procedural modelling tools and proposed, using one of these tools as described in section 3.1, a platform where to integrate modern procedural modelling techniques with freely available simulation tools (i.e., a physics solver). Both tools are either free or have freely available apprentice licenses. All this enabled us to build a practical soundness analysis tool that can be easily used by any stakeholder interested in masonry buildings, including, but not limited to, curators, architects, engineers, urban planers and even movie filmmakers and videogame designers. With all this, we can state that there is no reason, technical or whatsoever, to avoid the use of these tools that should become part of any modeler toolbox.

The test of the vault has had a duration of 30 minutes and shows that studies such as the ones carried out by Whiting et al. [24] are feasible on a low-cost implementation budget, without relying on expensive, custom implementations. Also, structural studies such as the one presented by Deuss et al. [26] are feasible also on a low-cost budget, but they still require the use of some clustering and optimization process that cannot be modelled without the introduction of an optimization library (e.g., the MOSEK library [58]), which is commonplace but still requires an expert's intervention.

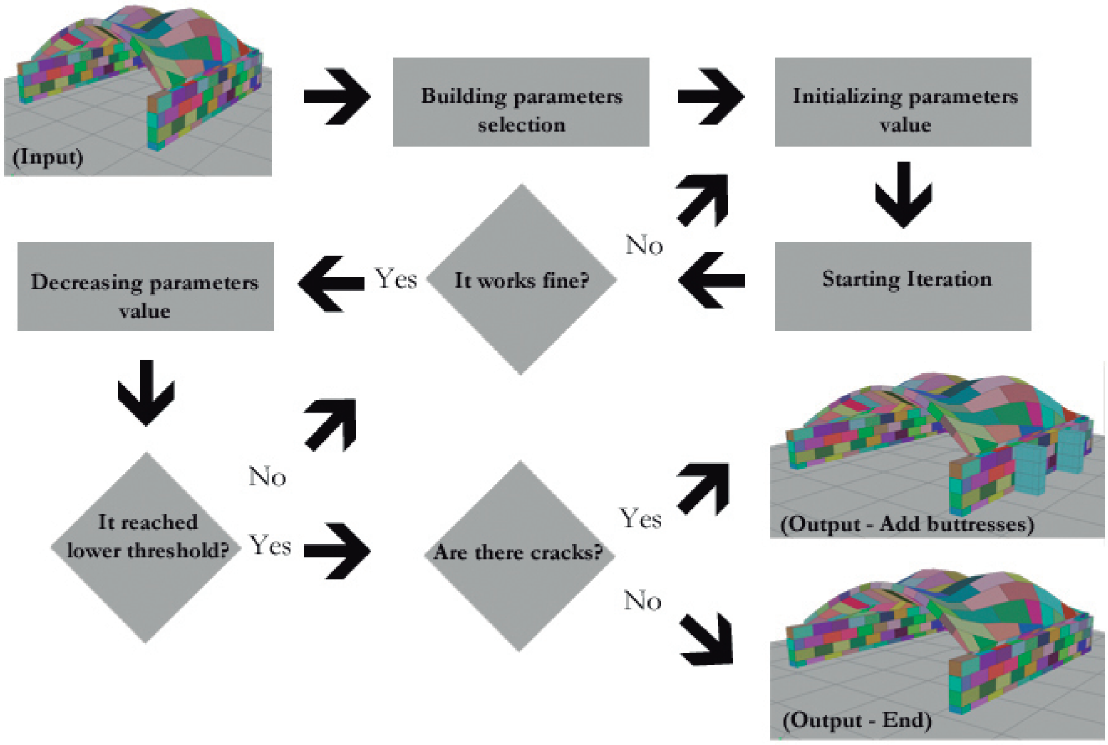

(Input)
Building parameters selection
Initializing parameters value
Starting Iteration
No
It works fine?
Yes
Decreasing parameters value
It reached lower threshold?
No
Yes
Are there cracks?
Yes
No
(Output - Add buttresses)
(Output - End)

Figure 3.10: Inverse Stuctural Analysis Pipeline.

3.4 Structural Stability: Inverse Analysis

This section presents the tool developed for the calculation of the configuration of the masonry structures necessary for supporting the weight of a given vault. In Figure 3.10 we can see the schema of our algorithm.

3.4.1 Parameter variations

In our algorithm we distinguish between two kind of parameters: intrinsic and constructive parameters. The first ones refer to all parameters related to the simulation and the numerical method used. These include the density for the bricks, $2691Kg/m^3$; friction, set to 0.7 corresponding to a non-polished granite surface; mortar strength, called *glue* in our system and set to 10, which represents a small value, as masonry buildings barely used any mortar to hold the bricks together because of the precision used to cut the stones; and rotational stiffness, which represents the resistance of the bricks to rotate in place, and that was given a value of $7000Nm/rad$. In general, intrinsic parameters are set as constraints, being left outside the optimization process, but there is no reason they cannot be included, if wanted, in the list of variables to find a value for. The constructive parameters refer to any geometric parameter, like height or thickness of the walls, the brick sizes, the size of the windows, and actually any parameter that could be defined in the input geometric parametrized model (in our case, created with procedural techniques) [24].

Our algorithm starts by letting the user select a few constructive parameters for some specified parts of the building (e.g., the wall thickness), previously configuring a maximum and minimum threshold with values of 2 meters and 0.5 meters, respectively. In case of existence or absence of certain elements (e.g., buttresses), the system tries with or without these elements to find a suitable solution. All numerical parameters are initialized to their maximum values. Then, in an iterative brute-force procedure, the algorithm varies the selected parameter determining at each step whether the structure falls down during the simulation or not, (e.g., the roof has fallen over a 1 meter from its initial position), see Section 3.4.2. In case the structure works correctly and stays in place, our algorithm optimizes the selected parameters, (e.g., the wall thickness decreases with a value of 0.5 meters in each iteration), always checking they have not reached the lower threshold limitation. After this optimization step, the algorithm simply outputs the results and provides analytic of the performance of the optimized resulting building. Please, refer to the scheme in Figure 3.10.

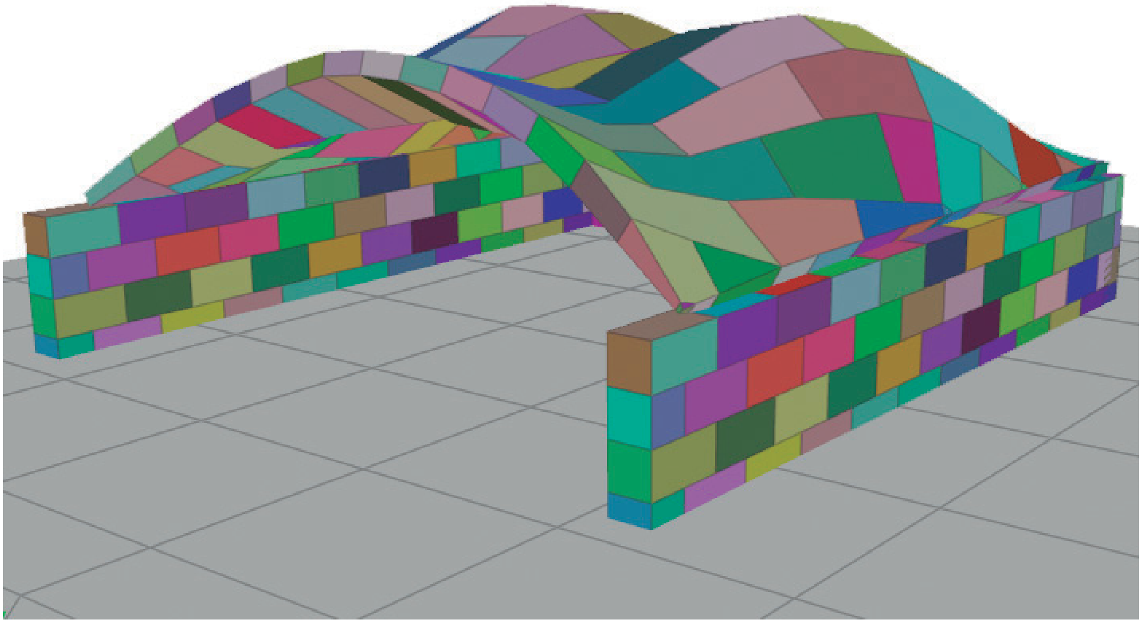


Figure 3.11: Structural Analysis Roof.

3.4.2 Structural Stability

At first, our algorithm takes the initial vertical position (the y-axis in our implementation) of the bounding box of the entire vault, and saves this initial value. At each iteration of the optimization process already described, a physical simulation is performed using the standard *Bullet* library. When this simulation has finished, the algorithm takes again the vertical position (Y coordinate) of the vault bounding box center, and analyses this with Equation 3.1.

$$(InitialRoofPosition - FinalRoofPosition) < Threshold \tag{3.1}$$

If the result of the difference between *InitialRoofPosition* and *FinalRoofPosition* is greater than the threshold, our algorithm understands that the vault is *not* supported by the walls. If it does, the algorithm repeats the same operation, but this time repeating the computation for each brick, which has a higher computational cost, and verifies the movement of the bricks with Equation 3.2.

$$\underset{allBricks}{Max}\ (InitialBrickPosition_i - FinalBrickPosition_i) < Threshold \tag{3.2}$$

Where *InitialRoofBrickPosition* and *FinalRoofBrickPosition* are the vertical positions of the bounding box centers of the initial and final position of a given brick, respectively. If they are greater than the user-defined threshold, then the algorithm assumes that the brick has moved along the vertical axis.

It is important to notice that, even if the vault (or the entire building) may hold its global position, there could be locality instabilities, resulting in cracks in the structure, which in our system are detected by the movement at the brick level. So, actually, our system is not only capable of detecting severe structural instabilities, but also to detect small movements.

3.4.3 Application Example: Walls and buttress

As we said, our system can easily accommodate a whole range of stability analysis, from global-level to brick-level assessments. This, of course, may include some external stresses that might threaten the whole building stability. As we compute displacements at the brick level, we traverse the walls or the roofs by visiting all bricks in order to detect some cracks (brick movement). In case that the detection is positive and surpasses the threshold of 0.05 meters from the initial position on the z-axis for the brick walls, the algorithm automatically adds buttresses and automatically calculates the most suitable number of buttresses for each wall, in a similar way as done before.

The first thing that the algorithm does is to clean any additional structure (e.g., buttresses) that might have been previously created. Then, it starts by accommodating one buttress at each wall, and iteratively proceeds by increasing the number of buttresses. The calculation of the buttress positions follows the expression in Equation 3.3.

$$InitialPosition = (SpaceButtresses + WallPosition) \tag{3.3}$$

Where *SpaceButtresses* and *WallPosition*, are constants that our algorithm calculates from the longitudes of the walls through the expresion in Equation 3.4.

$$LongitudeRow = ((NumberOfPoints \times SizeOfBrick) + HalfBrick) \tag{3.4}$$

Where the parameter *NumberOfPoints* refers to the number of bricks in a wall row. The *SizeOfBrick* refers to the brick longitude, taking into account that each row of bricks has, at the end, a *HalfBrick*. The last parameter refers to the size of a half brick, which usually is computed as $SizeOfBrick/2$, but can differ for aesthetic reasons in a real building.

Once the algorithm has obtained the *LongitudeRow* parameter, then it calculates the distance among buttresses through the following equation:

$$DistanceAmongButtresses = \frac{LongitudeRow}{(\sum NumberOfButtresses) + 1} \tag{3.5}$$

Our algorithm finally repeats the procedure for computing the *InitialPosition* (See Equation 3.3) used for the first buttress, repeating it for each wall. It uses the user-defined constant *DistanceAmongButtress* in order to obtain the initial situation for the next buttresses, in a way such that the set of buttresses are centered with respect to the wall endings. To create the buttresses, the algorithm follows the same procedure as the one explained for the creation of the walls, with the difference that it uses necessary parameters such as *Orientation* are taken automatically (i.e., perpendicular) from the respective wall. The algorithm keeps track of the overall constructive parameters for the stability of the building, taking default values for the intrinsic parameters like the ones to setup the global structural simulation, the stone density or the strength of the mortar needed between the bricks.

Our algorithm iterates over the buttress thickness the same way as before, detecting failures and cracks in their structure. If the result is positive for each wall, it repeats automatically the steps described above until the cracks have been eliminated. As a result the outcome can be that some walls require a smaller number of buttresses than others for their stabilization. See Figure 3.12.

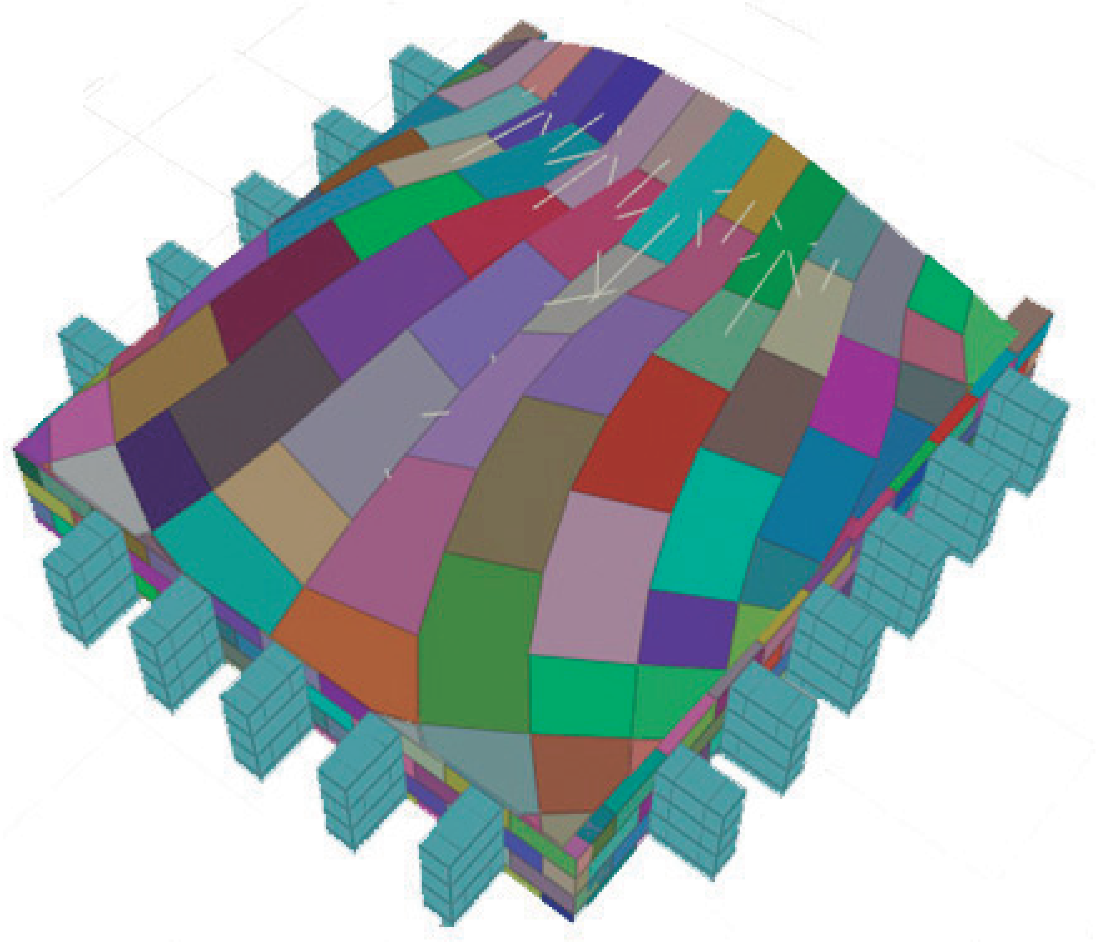
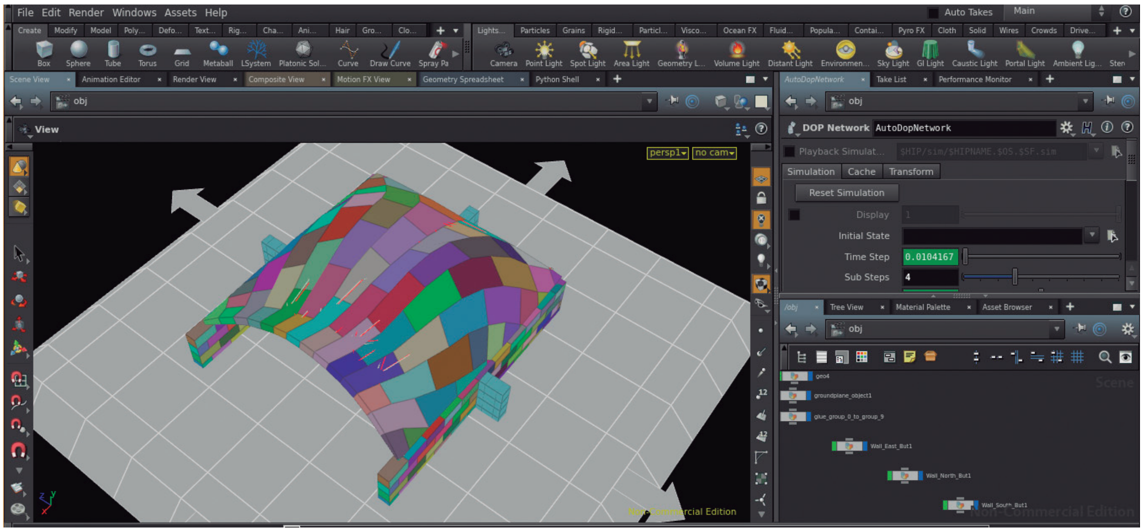


Figure 3.12: The walls with their respective buttress.

3.4.4 Results

We discuss here the results that we have obtained with our prototype implementation. For this purpose, we have designed two kinds of tests for our algorithm. For the first one, we tested the structure with a small value of stiffness for the building bricks (around $7000 Nm/rad$) and three different minimum thickness thresholds for the walls, *Maximum* (2m), *Medium* (1m) and *Minimum* (0.5m). This resulted in the addition of 6 buttresses in all walls except for the back one, which required 7 buttresses. For the second, we tested the structure with the same features but, in this case, with a larger value of the stiffness between the bricks (this time we used the value of $70000 Nm/rad$). We observed that the results depend heavily on the stiffness values and the amount of mortar (i.e., glue) added between the bricks. For a small value of the stiffness, the building is stable only for the thickest configuration, but for a large value of the stiffness the configuration is stable even for the thinnest wall, without requiring the introduction of additional supporting structures like buttresses. The reason is that, with the *Minimum Friction* value for the


File Edit Render Windows Assets Help
Auto Takes
Main
Create Modify Model Poly... Defo... Text... Rig... Cha... Ani... Hair Gro... Clo...
Box Sphere Tube Torus Grid Metaball LSystem Platonic Sol... Curve Draw Curve Spray Pa
Lights... Particles Grains Rigid... Particl... Visco... Ocean FX Fluid... Popula... Contai... Pyro FX Cloth Solid Wires Crowds Drive...
Camera Point Light Spot Light Area Light Geometry L... Volume Light Distant Light Environmen... Sky Light GI Light Caustic Light Portal Light Ambient Lig... Sten
Scene View Animation Editor Render View Composite View Motion FX View Geometry Spreadsheet Python Shell
AutoDopNetwork Take List Performance Monitor
obj
View
persp1 no cam
Non-Commercial Edition
Simulation Cache Transform
Reset Simulation
/obj Tree View Material Palette Asset Browser
obj
Scene
Non-Commercial Edition

DOP Network AutoDopNetwork

Playback Simulat... $HIP/sim/$HIPNAME.$OS.$SF.sim

Display 1

Initial State

Time Step 0.0104167

Sub Steps 4

geo4

groundplane_object1

glue_group_0_to_group_9

Wall_East_Butt1

Wall_North_Butt1

Wall_South_Butt1

Figure 3.13: The User interface of our application.

vault, the *Rotational Stiffness* has to provide the stabilization setting for the structure. If both parameters are set to low values (around $7000 Nm/rad$), the structure is unstable unless buttresses are added. In this case the maximum tolerable net force on the walls is smaller than the *Maximum Friction* value for the vault, resulting in an overall stability setting. See Figure 3.13.

With respect to timing, to perform a complete optimization our unoptimized prototype needs around 10 minutes to find the optimal values, even using our brute-force optimization method. Thus, it is expected that, with a more appropriate optimization algorithm (e.g., conjugate gradients) and with an optimized implementation, this time could be much reduced, in our experience up to the order of a few seconds. This would make this simulation feasible for near-interactive editing operations.

Chapter 4

An Earthquake Simulator

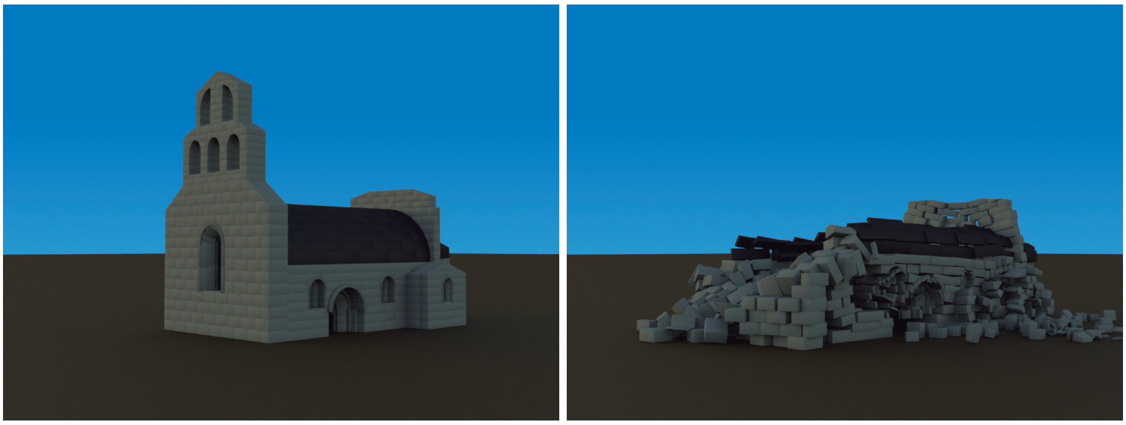


Figure 4.1: A church affected by an earthquake. On the left, a Romanesque church of the 11th century. On the right, the same church after the simulation of a large earthquake.

4.1 Introduction

The tools available that combine structural simulation of masonry buildings and seismic movements in the context of Cultural Heritage are testimonial. One possible reason about this gap in the literature is because this kind of simulations are not trivial and involve previous knowledge in seismology for carrying out the development of a believable tool for this.

However, we think that the combination of structural analysis of masonry buildings, such as churches, and seismic simulation, brings an opportunity for providing a methodology that allows researchers to better understand past events where an earthquake took place. To achieve our goal, a short description of a seismic movement when it takes place inside the crust of the earth and their propagation is given (see the appendices for a detailed derivation). Then, we continue describing our pipeline for the reproduction of

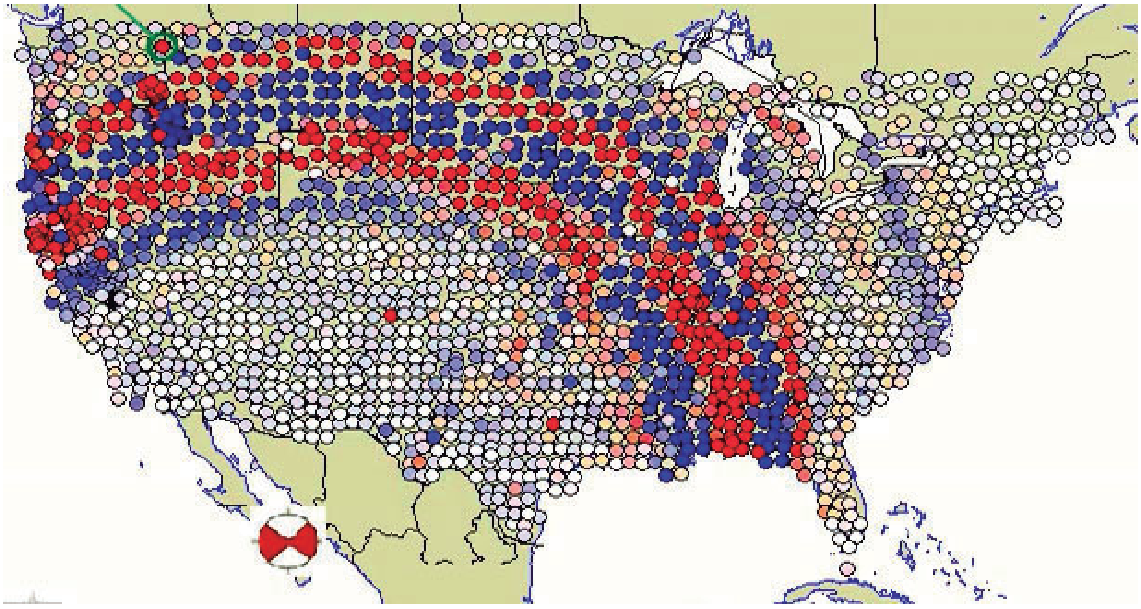


Figure 4.2: A seismic movement inside the Earth's crust takes place (see the big red-white point that represents the epicentre). The seismic movements known as *body waves* travel through the medium. When these waves reach the Earth's surface, they travel through it as surface waves (small red and dark blue points). (from U.S. Geological Survey [29]).

earthquake wave-fronts and the tests that have been carried out over masonry structures, see Figure 4.1.

4.2 Seismology

4.2.1 The seismic waves: Body waves

It is difficult to mathematically describe the propagation of a seismic movements through the Earth [30, 31], mainly because of their heterogeneity. Usually, this problem is solved by modelling seismic movements with a wave equation with homogeneous conditions. Thus, when an earthquake takes place inside the crust of the Earth, the liberated energy travels as a spherical wave-front by elastic displacements through the particles of the medium, i.e., as harmonic movements. The seismic movement under the earth surface is known as body waves. When it reaches the Earth's surface, the combination of the different motions creates surface waves, see Figure 4.2. When the wave-front is far from its origin, it starts decreasing its curvature progressively until it can practically be approximated by a plane wave.

Body waves travel through the medium in two ways. The first one is longitudinal waves, whose motion happens by compression, back and forth, of any particle that defines the affected area in the *x-axis*, parallel to the direction of propagation. The second type is known as transverse waves, whose motion happens only by vibrations in solid particles that define the medium through the *y-axis*, described as *SH-Waves*, and the *z-axis*, described as *SV-Waves*, both transversal to the direction of propagation, the *x-axis*. Transverse waves can be described similarly in both axis.

Both body waves, longitudinal and transverse, can be described mathematically in a similar way. When the wave-front passes through a particle that has a position x in the affected area A_x, it causes a displacement $\mathbf{u}$ of this particle with a force F_x for longitudinal waves, or a displacement with a force F_z for transverse waves. A particle of the same area in a nearby position $x+d_x$ with a length d_x and a mass $\rho d_x A_x$, experiences a displacement $u+d_u$ with a force $F_x+\Delta x$ for longitudinal waves, or experiences a displacement $u+d_u$ with a force $F_z+\Delta z$, for transverse waves. This force $F_x+\Delta x$ (or $F_z+\Delta z$) over the particle, is given by the particle stress σ_x. The equation of the waves for one dimension can be expressed as:

$$\rho\frac{\partial^2 u}{\partial t^2} = \frac{\partial \sigma_x}{\partial x}, \tag{4.1}$$

where σ_x is the particle stress that takes place in the affected area in one space dimension x, expressed as σ_{xx} for longitudinal waves and σ_{xz} for transverse waves, u is the second space dimension *(y or z)*, and time t. Thus, with Hooke's law, the equation for an elastic solid in one-dimension is $\sigma_{xx} = E\frac{\partial u}{\partial x}$ or $\sigma_{xz} = E\frac{\partial u}{\partial x}$, where E is the elasticity modulus of Young and $\frac{\partial u}{\partial x}$ is equivalent to the normal strain ε_{xx} or ε_{xz}. We can obtain, through substitution of the terms into Equation 4.1, the mathematical description of both body waves given by the general equation of a wave:

$$\frac{\partial^2 u}{\partial t^2} = C^2\frac{\partial^2 u}{\partial x^2}, \tag{4.2}$$

where C^2 is the wave velocity for longitudinal or transverse waves described as:

$$C^2 = \frac{E}{\rho}$$

However, the propagation of the body waves is in three-dimensions. In this case, when a disturbance takes place in a concrete area, the particle displacement is represented by its components in the respective axis, i.e., x, y and z. Thus, the equation of motion for compression or transverse waves in three-dimension can be written as:

$$C^2\nabla^2\theta = \frac{\partial^2 \theta}{\partial t^2}, \tag{4.3}$$

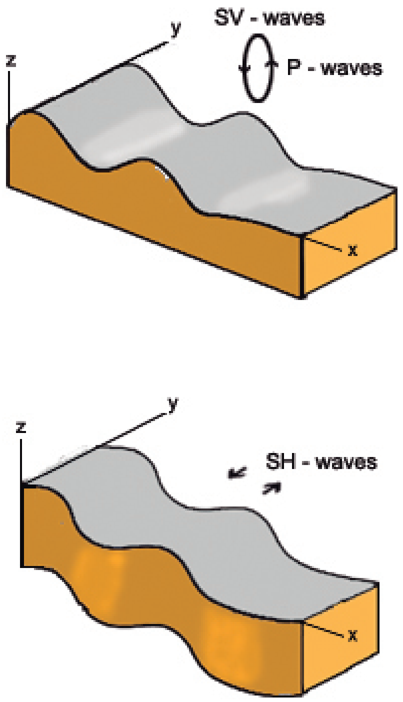


Figure 4.3: A graphical representation of the surface waves and their respective motions. Rayleigh waves motion (above). Love waves motion (below).

where θ is the dilatation or the shear disturbance of the particles and C is the wave velocity for both longitudinal waves and transverse waves. Thus, the longitudinal or compression waves are known as *P-waves* or Primary waves, because they are the fastest waves recorded on the seismograph. These waves can travel through the medium composed of solid, gas or liquid. The transverse waves are known as *S-waves*, or Secondary waves, because their velocity of propagation is slower than the primary waves and only travel through a solid medium.

4.2.2 The seismic waves: Surface waves

When the body waves composed of *P-waves* and *S-waves* reach the surface, their combination and their interaction with the free surface transform these into surface waves that propagate along the surface of the Earth from the earthquake epicenter. These are slower than body waves and their amplitudes decay with the depth of the medium. However, their effects are more destructive for human environments. Surface waves are distinguished by their motion, and they are grouped into *Rayleigh waves* and *Love waves*. See Figure 4.3.

4.2.2.1 Rayleigh waves

The motion of Rayleigh waves is a combination of *P-waves* and *SV-waves*, having a mathematical description given by the position equation for the x-axis and z-axis at the free surface. These equations can be described as:

$$\theta_x(x,t) = a\left(\frac{\omega^2}{2k\beta^2}\right)\cos\left(kx-\omega t\right), \tag{4.4}$$

$$\theta_z(x,t) = a\left(\frac{2k\kappa_\alpha}{k^2+\kappa_\beta^2}\right)\left(\frac{\omega^2}{2k\beta^2}\right)\sin\left(kx-\omega t\right), \tag{4.5}$$

where $\kappa_\alpha^2 = k^2-(\omega/\alpha)^2$ and $\kappa_\beta^2 = k^2-(\omega/\beta)^2$. Equations 4.4 and Equation 4.5, describe the motion of the *Rayleigh waves* as retrograde and elliptical parallel to the direction of propagation. See Figure 4.3, top. For a complete mathematical derivation of these equations, see the Appendix A.

4.2.2.2 Love waves

Their motion comes from the *SH-waves*, focused on the surface. These waves are dispersive and faster than *Rayleigh waves* and their propagation is along the surface. *Love waves* travel in groups of waves, resulting in both carrier and envelope waves. Thus, the mathematical description of these can be expressed by the position equation of a sum of two harmonic waves that can be described as a product of two cosine functions, as:

$$\theta_y(x,t) = a2\cos(kx-\omega t)\cos(\delta kx-\delta\omega t). \tag{4.6}$$

Equation 4.6 describes the *Love wave* position in a given time, where the carrier wave has an angular frequency ω and wavenumber k and the envelope wave has a lower angular frequency $\delta\omega$ and wavenumber δk. See Figure 4.3, bottom. For a complete mathematical derivation of these equations, see the Appendix B.

4.2.3 The Richter magnitude scale

The seismograph is a device designed for recording the vibration of the ground motion. Based on the Woods-Anderson seismograph, the seismologist Charles F. Richter developed a formula that measures the magnitude of an earthquake where the magnitude is based on the amplitude recorded. The formula is based on a base-10 logarithm of the amplitude recorded in *mm* by a seismograph [29, 31]. The original formula is in disuse because the Wood-Anderson seismograph is not used for recording seismic events

	Magnitude
Minor	2.0 - 3.9
Light	4.0 - 4.9
Moderate	5.0 - 5.9
Strong	6.0 - 6.9
Major	7.0 - 7.9
Great	8.0 - 9.9

Table 4.1: The Richter magnitude scale that measures the energy released from a seismic event.

nowadays. In Equation 4.7 we can see an adaptation [31] of the original Richter formula for calculating the magnitude of an earthquake:

$$M = \lg A + 3\lg(8\Delta t) - Q, \tag{4.7}$$

A is the amplitude recorded from the seismograph, Δt is the time between the *P-waves* and the *S-waves*, Q is the regional scaling factor and M is the magnitude number of energy released by an earthquake, expressed in decimal numbers. Table 4.1 shows the values of earthquake magnitudes according to the Richter scale.

4.3 Implementation

Our method is based and developed on off-the-shelf tools such as *Houdini* from SideFX [56], Houdini's Python libraries and its plug-in *Bullet* [36]. Our tool has been designed for reproducing wave-fronts over a geometric grid that conforms the soil in a period of time determined by the user, assuming the wave-fronts originated at an epicenter located far away. For the wave-fronts, we focused on both surface waves, *Love* and *Rayleigh* waves, because they are the main cause of the damage that takes place during a seismic movement.

Our implementation follows the pipeline shown in Figure 4.4. First, we can see the *Input Geometry*, which is a geometric representation of a stone masonry structure, which must be connected to the ground through constraints defined by *Bullet*. The *Ground Geometry* is a grid that supports the generation of the seismic waves generated by the *Surface Wave Simulator*. The wave motion of the grid is combined with the building geometry through a *Dynamic Simulator* based on *Bullet*, which generates an *Output Geometry*.

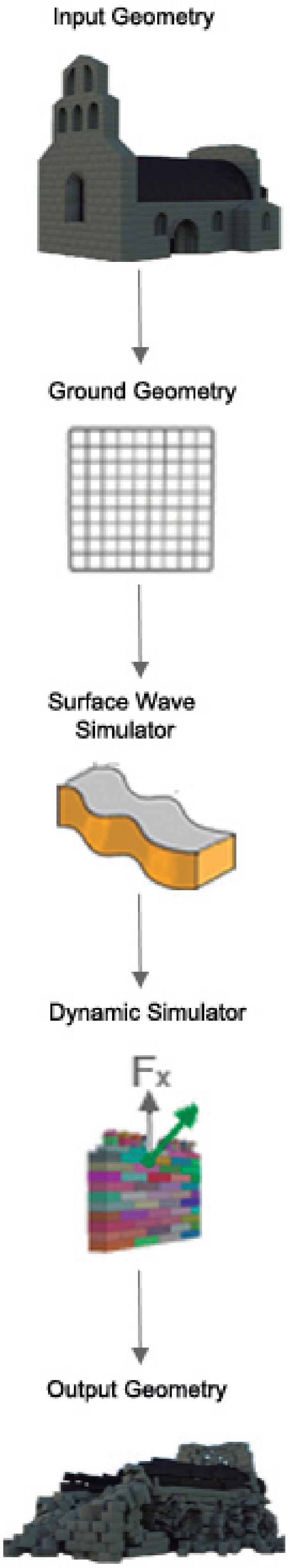


Figure 4.4: Our pipeline. The building geometry, is connected to the ground geometry which is controlled by the surface wave simulator. The dynamic simulator controls all the physics simulation and makes the building destruction possible.

4.3.1 Input geometry

The recreation of ancient masonry buildings has been implemented following the methodology described in Section 3.2, which takes an input building geometry given by the user, creates the bricks and finally gives them physical features. This procedure allowed us the creation of a set of ancient masonry structures. Specifically, buildings built with stone and designed for supporting vertical loads, as shown in Figure 4.1, such as walls, towers, churches and houses.

These buildings have been configured with the following physical features: *density*, with a value of $2691 kg/m^3$ for simulating the granite stone that conforms the building structure; *friction coefficient*, with a value of 0.7 that corresponds to the granite value; *bounce*, with a null value. The glue parameter has been activated for simulating the ancient mortar used by the masons in the process of building the stone structure.

The geometry of our buildings has been connected as a rigid body through a dynamic network in order to allow control the object and the features described before. The connection of our building geometry with the ground has been carried out through the dynamic network with the position process. This controls the point position of all points that conform the ground in the proximity of the building geometry, and set the new positions for the building in all those axes. This positioning process also allows the user to configure the bricks that are linked to the ground and that will follow its motion in a simulation process.

4.3.2 Ground geometry

The ground is implemented by a grid geometry that has a size of $1500m \times 1500m$ and it is composed by $8x8$ points. This grid has also physical features such as *bounce*, *friction coefficient* and *ambient temperature*. As mentioned, the motion of the ground is controlled by the surface wave simulator.

4.3.3 Surface Wave Simulator

The surface wave simulator has an interface oriented for non-expert users that allows the control of the earthquake strength based on the Richter magnitude scale. The strength can be categorized from *minor* to *great* earthquakes (around a value of 8.0), through the interface, according to the Richter magnitude, see Table 4.1. With this restriction, we avoid the saturation in the calculus of Richter magnitude that happens over values greater than 8.0. Our tool only allows the magnitude value adjustment before starting a simulation, which depends on two configurable parameters, *delay time* and *amplitude* for calculating its value through the Equation 4.6. The *delay time* given in *seconds*, refers to the time difference between the arrival of the *p-waves* and the *s-waves* registered on a seismograph; and *amplitude* given in *meters*, refers to the amplitude of the waves.

Our tool also allows the configuration of other parameters through the interface, such as the *time duration* of the earthquake, the *frequency* (given in *Hz*), the *phase* (given in radians), and the *wave direction* (given in degrees). The values of this last parameter correspond to $0-180$ degrees in the North-South axis and $90-270$ degrees in the West-East direction.

Internally, the surface wave simulator has been written in the Python language. When the user starts the simulation, our algorithm first reads all the wave parameters configured previously through the interface. The phase velocity of these waves has been modeled as $C_R = \beta\sqrt{0.8453}$ for *Rayleigh waves*, and $C_L < \beta$ for *Love waves*, where β is the *S-waves* speed for a granite ground, which is set to $3300m/s$. Also, it reads the time parameters, the *earthquake duration*, and it does the respective calculations for the *angular frequency*, among other parameters such as the *period of the wave* and *the wave length* (λ), needed for the correct application of the different wave motion.

Once the algorithm has made the parameter calculations, and for each point of the grid that represents the ground, it computes the *velocity* for each axis. Through this calculation, our algorithm determines if the wave-front has arrived at a concrete point. If the velocity values on the x and z axis are not equal to *0*, it applies the point position equations, Equations 4.3 and 4.4, respectively, which simulate the motion of the *Rayleigh waves*. The algorithm does the same for the y coordinate, where it applies the Equation 4.5, which simulates the motion of the *Love waves*. The surface wave simulator sends the data of the grid position to the dynamic network that is controlled by the physics engine [36].

4.3.4 Dynamic Simulator and Output Geometry

Because the geometry and the physics engine are not directly connected to each other, the *Dynamic Network* is the component that acts as a connector where all the geometry data of the grid that conforms the ground and the building geometry are processed by the physics engine, capable of performing the collision detection, resolve the collisions and their constraints, and finally perform the transformation of all objects affected by the simulation.

After defining the mentioned elements, we have added to the Dynamic Network the physical objects and the ground element that conform the simulated world. The Dynamic Simulator is capable of simulating physical objects such as rigid and soft bodies, although we did not use the latter ones. For the creation of a rigid body object, we have provided its physical features such as *density* and *friction*. The *density* is used by the solver to calculate the object's mass. Also, we have provided the *collision shape*, which is the object volume used to detect collisions with other objects in the scene (i.e., the ground or other bricks). For each brick we have used a box collision shape. We connected the lower bricks in the walls to the ground, by fixing their positions to the movements of the ground at their respective locations.

When the user starts the simulation, the physics engine creates the dynamic world with its rules, such as the gravity force, adds the physical objects such as the walls and the ground, and then performs the simulation with the following steps for each frame, a deeper description of these has been given in Section 3.3.1:

1. **Broad-Phase Collision Detection**.
2. **Narrow-Phase Collision Detection**.
3. **Constraints**.
4. **Modifies Physics Parameters**.
5. **Loop Rigid Body**.

When the simulation has finished, the physics simulator deletes the objects and the dynamic world created in the physics simulation process.

	Test 1	**Test 2**	**Test 3**	**Test 4**
A: amplitude (m)	0.002	0.02	0.2	0.5
$\triangle t$: delay time (s)	21	21	21	32
Q: regional factor (s)	-2.92	-2.92	-2.92	-2.92
Richter Magnitude Result	**4.06**	**5.06**	**6.06**	**7.0**
Rayleigh Frequency (Hz)	4.05	4.05	4.05	4.05
Love Frequency (Hz)	4.05	4.05	4.05	4.05
Phase (rad)	1	1	1	1

Table 4.2: Main settings applied for each test over the two walls. A is the amplitude value for both surface waves, $\triangle t$ is the time and Q is a constant, both values given for the calculation of the Richter Magnitude.

4.4 Results

We have tested our implementation through a set of examples based on different magnitudes in the Richter scale, applied to a set of ancient masonry buildings. In the first test, we applied the earthquake with different magnitudes over single walls. See Section 4.4.1. Next, in the second test case, we observed how different wave frequencies have affected the given structures.

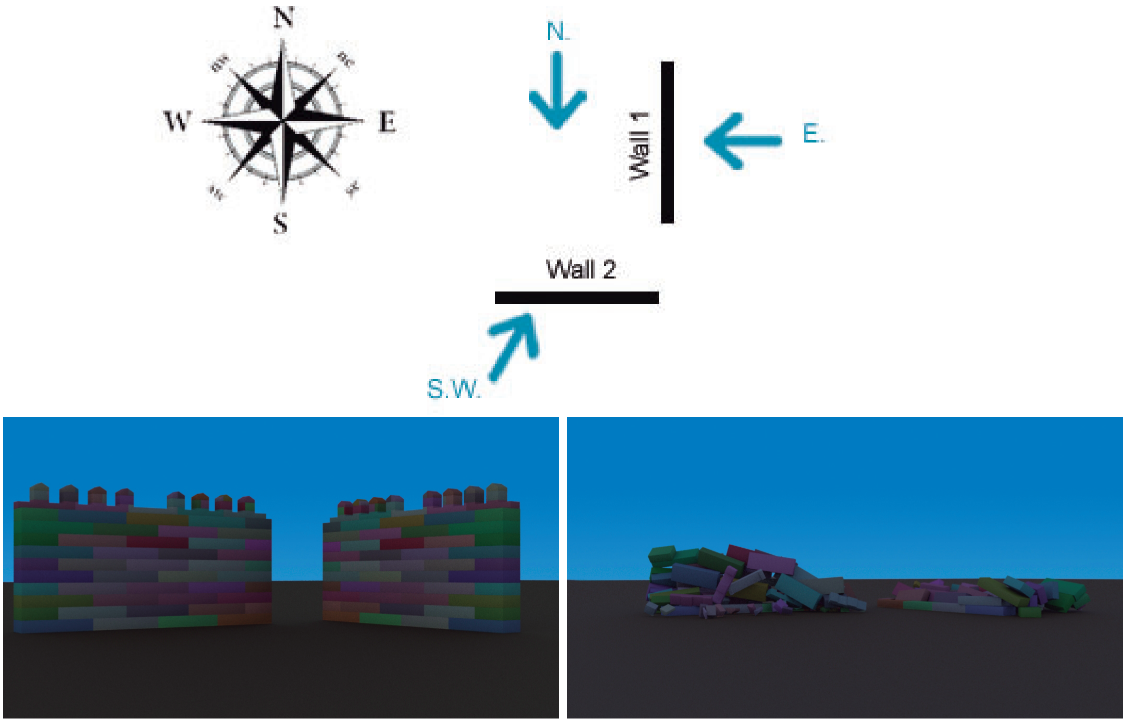


Figure 4.5: Top: Orientation of the walls over the grid and the seismic wave-front directions. Bottom: The walls before and after an earthquake of magnitude 7.0 from a South-West wave-front.

On the other hand, in order of validate our tool, we have made two visual validation tests. The first one is based on the reproduction of a real case. See Section 4.4.3. The second is based on the verification of the results against a similar software. See Section 4.4.4.

All tests have been performed on a CPU Intel-core i5-3210M with 12 Gbytes of RAM memory.

4.4.1 Test over two walls

The goal of this test is to analyze the behavior of our method when it applies the seismic wave on some basic masonry constructions with different Richter magnitude scales. For this purpose, in this test we placed a ground with features as described in the last section. Over this ground we placed two masonry walls composed of 74 bricks each, with their respective battlements and different orientations, North and East, see Figure 4.5. Through the dynamic network, we have configured their physical features such as the *density* of the wall material set with a value of 2691 kg/m^3 for simulating granite stone, and the *friction coefficient* with a value of 0.7, which corresponds to the *friction coefficient* of rock.

We have adjusted the parameters needed for each earthquake with the values shown in Table 4.2. We have made three tests with different wavefront orientations, such as

Richter Magnitude	4.06			5.06			6.06			7.0		
Wavefront	N.	S.W.	E.	N.	S.W.	E.	N	S.W.	E.	N.	S.W.	E.
Wall 1	0	0	0	0	0	0	10	69	69	29	69	69
Wall 2	0	0	0	0	0	0	40	58	69	37	69	69

Table 4.3: Test results that show the number of bricks displaced for each wall and earthquake strength.

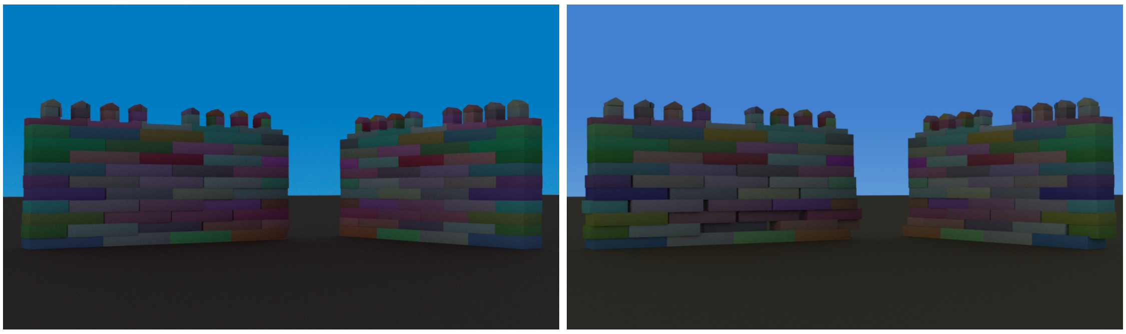


Figure 4.6: Frequency test. On the left, the walls under seismic motions with a frequency of 1.05 Hz. On the right, the walls under the same earthquake with a frequency of 5.05 Hz.

North, South-West and East for each Richter magnitude and an earthquake duration of 20 seconds. With the aim of quantifying the number of bricks affected in each earthquake, we set a threshold of 15 *cm* for the minimum movement of a brick on the considered wall. With this test we can analyze how the different magnitudes and wavefronts affect a human-made structure. The results of these tests are given in Table 4.3.

4.4.2 Frequency test

The purpose of this test is to verify the affectation of single walls, described in the previous test, by earthquakes that have different wave frequencies. We configured each earthquake with a magnitude of 6.06, a duration of 20 seconds and a South-West wave-front direction. Then, for each test, we have increased the frequency with the same value for both surface waves and counted the number of bricks that were displaced, as already explained. See Figure 4.6.

Frequency (Hz)	0.5	1.05	2.05	3.05	4.05	5.05
Wall 1	9	20	33	69	69	69
Wall 2	1	4	18	57	58	69

Table 4.4: Test results that show the number of bricks that were displaced for each wall and wave frequency.

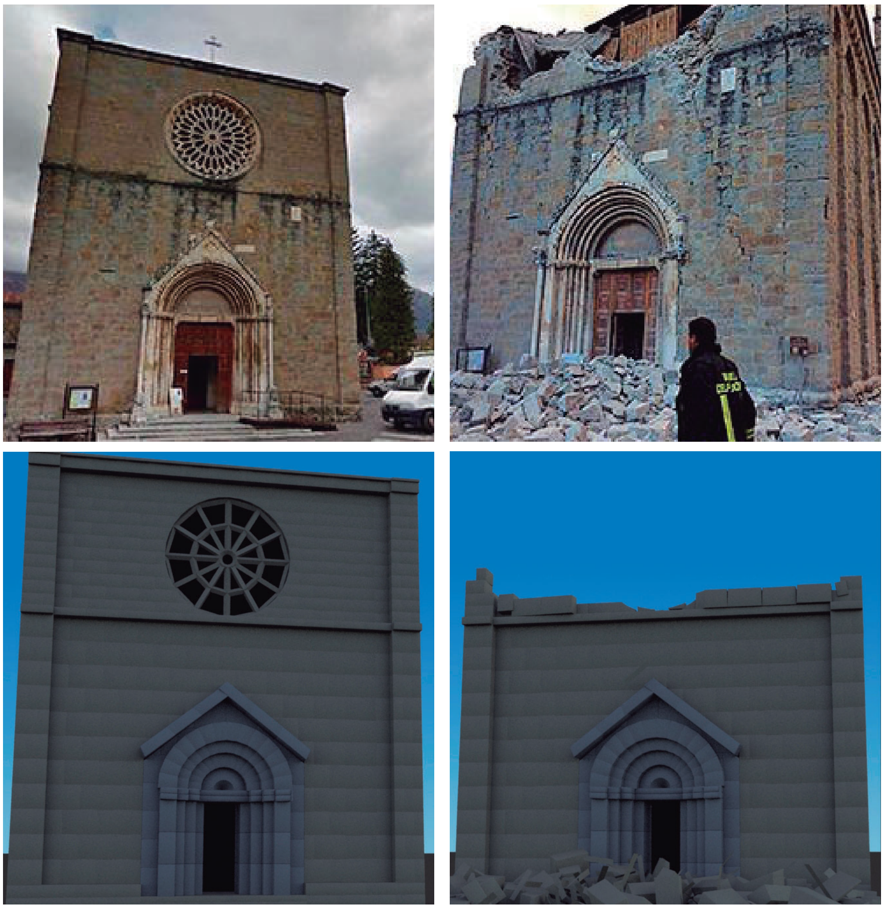


Figure 4.7: On the top, pictures from real St. Agostino church [27] before and after the earthquake took place. Below, the modelled church before and after the simulation of an earthquake with the similar properties as the real one.

4.4.3 Real case test

The August 24th of 2016 an earthquake of a Richter magnitude of 6 took place in the center of Italy, where villages and towns such as *Accuamoli*, *Pescara del Tronto* and *Amatrice* were severely destroyed [28]. As a qualitative test of our technique, we have reproduced the effects of this earthquake over an ancient masonry building located in one of these cities. The selected building has been the *St. Agostino* church of *Amatrice*, a church that had an architectural structure of the 13^{th} century combining both the Romanesque and Gothic styles, which was severely damaged by the earthquake.

To recreate the building, as there are no accurate measures available, we resorted to traditional modelling techniques for the main facade, see section 3.2. We have applied the earthquake following the pipeline described before with a Richter magnitude of 6.06, see Table 4.2. The result of the test is shown on Figure 4.7. While it is not possible to obtain exactly the same result. There could be some possible reasons because our seismic methodology does not reproduce the building destruction accurately. On one hand, this

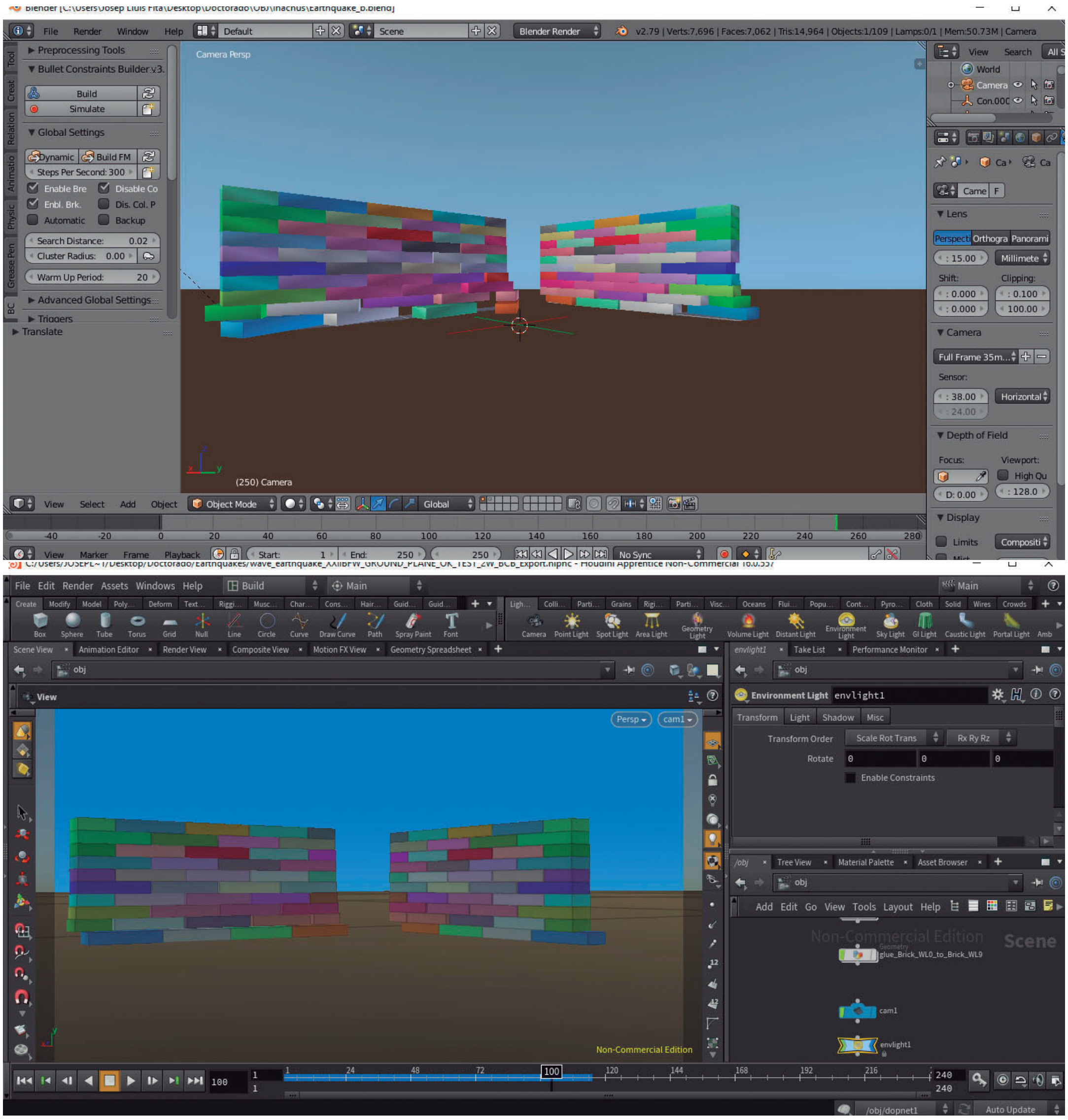


Figure 4.8: Results of the Simulation Builder for the *INACHUS project* (top) and the simulation performed with our tool (bottom).

has not been designed for reproducing the structural cracks of a building. On the other hand, the modelling tool does not recreate the infill of mortar between bricks, as the real case shows. But, brick by brick, by comparing both images we can appreciate a high degree of similarity in the building destruction. This allows us to be positive about the validity of our tool for successful earthquake simulations on ancient masonry buildings.

4.4.4 Software case test

The *INACHUS Project* [33] is a software platform implemented by a consortium of companies and public entities, such as the *University of Twente* [34] (Netherlands), sponsored by the European Union. The main aim of this consortium is to provide solutions for the detection and localization of the victims that have suffered a catastrophe. One of these solutions is a plug-in called *Constraint Builder* that provides a collapse simulation tool to the Blender software suite [35] using Bullet. This plug-in allows users without experience to manage Earthquake simulations.

The purpose of our test is to compare the behaviour of a similar earthquake with our tool and with the mentioned simulation package. For that purpose we used a model similar to the one used in Section 4.4.1. The main features of the test in both solutions are the *time duration*, which is set to 20 seconds; the *frequency*, that has a value of 4.05 Hz; the *wave direction*, aligned with North direction; and the same Richter magnitude, with a value of 6.06. In Figure 4.8 we can see we have used the same model in both simulations.

4.5 Results

Table 4.3 shows the number of bricks displaced during the simulations for the *two-wall test*. For a weak earthquake, with a Richter scale of 4.0, the walls resist without problem the earthquake in any wave-front direction. We can see that the number of bricks displaced after the earthquake simulation increases with the magnitude of the earthquake, where, for strong and major earthquakes, there was a larger number of bricks displaced, and the walls ended severely damaged, or even completely destroyed in some cases. Thus, from both tests we can observe that both the magnitude of an earthquake and the wave-front direction are important factors to be considered. Also, the wave frequency is an important factor too, and it is tightly related to the number of bricks displaced in each simulation.

On the other hand, we do not expect a perfect similarity in the results from our qualitative tests. If we observe Figure 4.7 from the *real case test*, we can observe that the destruction level of the church facade visually is very similar. Also, if we observe Figure 4.8 from the *software case test*, we can see the visual similarities in the final result of both pieces of software are practically the same, both showing the plausibility of our approach for a realistic earthquake simulation on ancient masonry buildings.

Chapter 5

A Virtual Reality Front-End for Earthquake Simulation

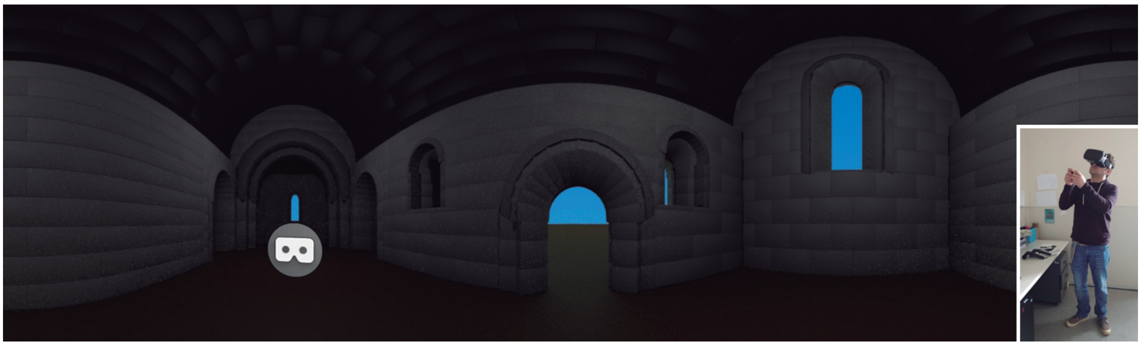


Figure 5.1: A spherical view of the Romanesque church interior used in our virtual reality application, see the inset. Inset: a user testing it with the headset and controller.

5.1 Introduction

The application of Virtual Reality techniques to the Cultural Heritage field has given researchers a new way for the diffusion of past events and historical building recreation among the general public. These Virtual Reality applications have been developed mainly on expensive hardware which is available only for a reduced number of people and usually at public institutions like museums.

In cultural heritage, most virtual reality applications have been designed mainly for architecture reconstruction or, at most, past event recreations. However, the design of applications that combine ancient building modelling, with structural and seismic simulation, and also accessible for all kinds of users, is practically testimonial. It must be taken into account that this kind of 3D simulation is unfeasible for low-cost devices such

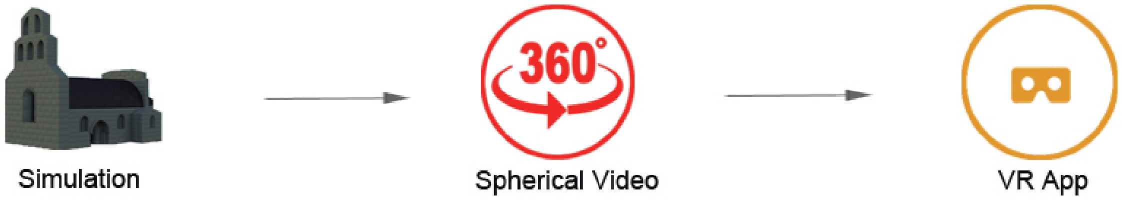


Figure 5.2: Our pipeline: A simulation is processed, and the resulting animation is rendered trough a camera tailored for Virtual Reality. The 360° video obtained is exported to the Virtual Reality application that displays it on a headset.

as smart-phones. During the process of finding a solution for this problem, we realized that exporting the simulation to a VR format based on 360° virtual reality video could be a suitable solution for resolving the issue between low-cost platforms and virtual reality applications with a certain geometry-complexity.

In this chapter, we have focused on solving this problem by exporting the simulation as 360° camera videos, where a recreation of a past event that simulates the effects of an ancient building, like a church under a seismic movement, is shown on a low-cost Virtual Reality system. The main contribution is providing an immersive experience for historians, curators and ordinary people. Our application is completely based on off-the-shelf tools and is designed to be visualized for everybody that has a Virtual Reality enabled device such as a smart-phone with a headset, see Figure 5.1.

5.2 Overview

The main problem with structural and earthquake simulations is that they are impracticable for real-time, mainly because a lot of computational resources are required to process the geometry by the physics engine, taking into account all physical constraints and equation setup. Additionally, exporting the full 3D scene is unfeasible for a low-cost virtual reality application that runs on a low-cost hardware platform, because the hardware of these devices is usually not powerful enough to handle the requirements of the whole geometry and animation, producing a latency between the frames that can cause the well-known motion sickness effect.

In this chapter we present one possible solution to the issues described before for situations where real-time simulation and low-cost devices are involved, which is shown in Figure 5.2. In this pipeline we can see the *Simulation* is computed first in a high-performance computer, rendered and then exported in a *Spherical Video* format. Once the 360° video has been created, it is exported to the *Virtual Reality application* to be visualized in real-time on a headset device.

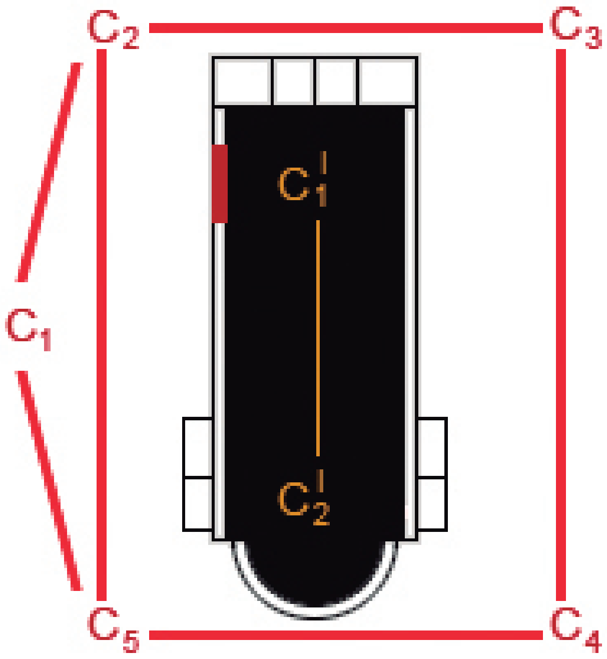


Figure 5.3: The church map and the point-camera locations have been distributed as example. In red, the exterior point-cameras. In orange, the interior point-cameras. The red rectangle is the linking point between the exterior and the interior of the church.

5.3 Earthquake Simulation and Visualization

5.3.1 Earthquake Simulation

Our earthquake simulation tool uses the following elements:

- **The surface wave simulator** has an interface oriented for non-expert users that allows reproducing earthquakes categorized both as minor and major magnitudes, according the Richter scale, see table 4.1. Through the user interface, the user can configure earthquake wave parameters such as *frequency*, *phase*, *time duration* of the seismic event and the *wave-front direction*, by defining the wave-front angle or with cardinal point values such as *North, North-West, North-East, East, South-East, South, South-West and West*.

- **The Ground** is represented with a grid with a size of $1500m \times 1500m$ and N points. The grid has been configured with physical features such as *bounce* and *friction coefficient*, that are used to react to the movements produced by the earthquake. This structure is directly controlled by the surface wave simulator.

- **The dynamic network** connects the geometry data from the ground with the building to be processed by the physics engine. This engine is capable of detecting the collisions among objects and resolve these for performing the object behaviour involved in the simulation. For this we have chosen the *Bullet* [36] solver library. See Section 5.4 for more details about the exact implementation.

- **An input geometry** of an ancient masonry building, previously modelled, is imported in the simulator and connected through a dynamic network with the positions of the ground being controlled by the surface wave simulation. This process is called *position process* and takes the positions of the points of the ground near each element of the building geometry, and transfers this information to the lowest layer of building bricks, which are directly linked to the ground and thus receive the full impact of the simulated earthquake. The building bricks have a component called *glue* that simulates the mortar and is used to give some cohesion between the building bricks. The modelled building has been configured with physical features such as *density*, with a value of $2691 kg/m^3$ for simulating the granite stone of the building; and the respective *friction coefficient* value, 0.7.

  When the simulation starts, the main parameters given by the user through the interface are used to calculate the *angular frequency*, the *phase velocity* and all relevant parameters required for the simulation of the surface wave motion involved in the earthquake simulation. Once these calculations are done, the points that conform the grid are loaded, see Chapter 4 4.3. The first step is to determine if the wave-front has arrived from its epicenter, computed as a simple linear velocity calculation. If the result is true, then the formula given by Equations 4.4 and 4.5 for simulating the *Rayleigh* motion, and Equation 4.6 for the motion reproduction of the *Love waves*, are applied. These steps are repeated for each ground point during the simulation time where the surface wave simulator sends the position data of the grid through the dynamic network, which in turn, is controlled by the physics engine.

5.3.2 Spherical Video

The implementation of the surface wave simulator previously described allowed us to obtain the result of an earthquake simulation over an ancient stone building. However, as mentioned before, the direct results of these simulations are unfeasible to be exported to a VR tool on a low-cost device, so we exported these results as a set of 360^o VR videos, which will allow us to display the simulation in real-time on a low-cost device, as shown in Figure 5.2.

The spherical video has been designed to be displayed inside of a spherical shell, where the system corrects any perspective-induced deformation. Spherical video rendering has been carried out through a set of specific cameras located around and inside of the building with the aim of recording different points of view of the same simulation, see Figure 5.3. Also, each camera has been configured with a specific capture angle for rendering with a realistic *ray-tracing* algorithm, using an *equirectangular* projection, see Figure 5.1. We must say that we have not used stereo projection because mono projection allows us saving time in the rendering process and the low-cost devices support lighter files better.

Once the set of 360^o VR videos have been recorded, loading them into a virtual reality application allows to see, in real time, simulations on a low-cost headset, as shown in the pipeline at Figure 5.2.

5.3.3 Virtual Reality Application

In this project we have used the *Unity 3D engine* [59], a well-known game engine designed for a large number of users, from apprentice to professional game developers, and available for almost all operating systems. This engine supports the development of Virtual Reality and Augmented Reality applications, so we can develop any application for all kinds of Head Mounted Displays available in the market, both for desktops such as *Oculus Rift* [60], *HTC Vive* [61], or *Windows Mixed Reality* [62], and also for the cheapest Head Mounted Displays for mobile VR, like *Google Daydream or Cardboard* [63]. We decided to develop our cultural heritage application using the *Unity 3D engine*, with the idea of generating an immersive experience inside a scenario that recreates an historical event where an earthquake takes place. As we target low-cost devices appropriate for exhibitions and mass events, our implementation was designed for running on the *Google Daydream or Cardboard* that only allows the user *three-degrees-of-freedom* (3-DOF), with fixed viewpoint position, for interacting with the 3D space. The video controller of the Head Mounted Display, *Trust GXT 720* [64], has been configured with functions such as play and pause, video forward or rewind, and teleportation node selection. A demonstration video for our application can be found at `https://drive.google.com/open?id=1d30WOQwrakJW3SyldV_qWoJLhLT9Pc2d`.

The application 3D space has been designed for displaying the imported and prerecorded 360^o Virtual Reality videos generated with *Houdini*. The videos are controlled by C# scripts and rendered as a shaders inside of a *photo-sphere* surface using an *equirectangular* image. In its center, we have located the main camera that gives an interactive user-environment for the immersive first person experience. This video rendering technique is a suitable solution for the proposed problem because requires less computing power than a full 3D scene rendering for reproducing real-time structural simulations, as shown in Section 5.4.

Immersed in this scenario, users can see interactive nodes, C_i, that are part of an undirected graph $G = (V,E)$ located on the exterior and interior of the church, where V is the set of the graph nodes C_i (we add an extra I superscript to refer to the nodes in the interior of the building C_i^I, just for clarity of presentation purposes), and E a set of edges, each defined as a pair of nodes $< C_i, C_j >$, where the user can move from node C_i to node C_j. From this graph, locomotion has been designed for user teleportation among specific nodes distributed over the space, as can be seen in Figure 5.3. In this figure we can see the red nodes that show the interactive nodes outside of the church, while the orange nodes conform the interior teleportation nodes. We have designed this kind of locomotion based on specific nodes with the aim of reducing the motion sickness that is caused by lags in screen updates, colliding with the user brains who expect synchronized changes. We have added to our motion system a gaze-based point for helping the user interaction. When the gaze point is over a node of interest, it changes its color from white to blue.

The navigation among the graph nodes is free and allows users to make their own paths through the graph nodes. Thus, when a user selects a first node of the graph represented by C_i, automatically the appropriate 360$^{\text{o}}$ VR video and neighbour nodes are loaded in the scenario. At this point, the user can interact with the 360$^{\text{o}}$ VR video by playing the recorded simulation, stopping it if wanted to see the destruction frozen at a given time. If the user selects another point of view, represented by node C_k, the corresponding video is loaded at the same time-step that the one that was stopped at the last node C_i. Also, the neighbour nodes of C_k are loaded and the nodes that were neighbours of C_i that are no longer needed are hidden. The same interaction and navigation occurs with all the nodes that conform both the exterior graph and the graph located inside the church. This way, although we are dealing with 360$^{\text{o}}$ videos, we solve the visibility problem of hiding the nodes *behind* the building, which should not be seen though its structure.

An example of a graph is given in Figure 5.3. On it, we can see how the vertices and edges can be distributed over the space. As we have explained before, when a user selects the C_1 point, the respective video and neighbours represented by the points C_2 and C_5 are loaded. This kind of interaction allows the user to navigate through the exterior or interior graph and see the simulation from different points of view, see Figure 5.5. Also, we allow some degree of interaction with the spherical video image, because the user can interact with hidden buttons located, for example, on the church door and are only activated when the points located in front of these are selected. These buttons are associated with each node, and are show or hidden on one graph or another, and presented in the exterior or interior church view, according to the predefined authored interactions. See Figure 5.4.

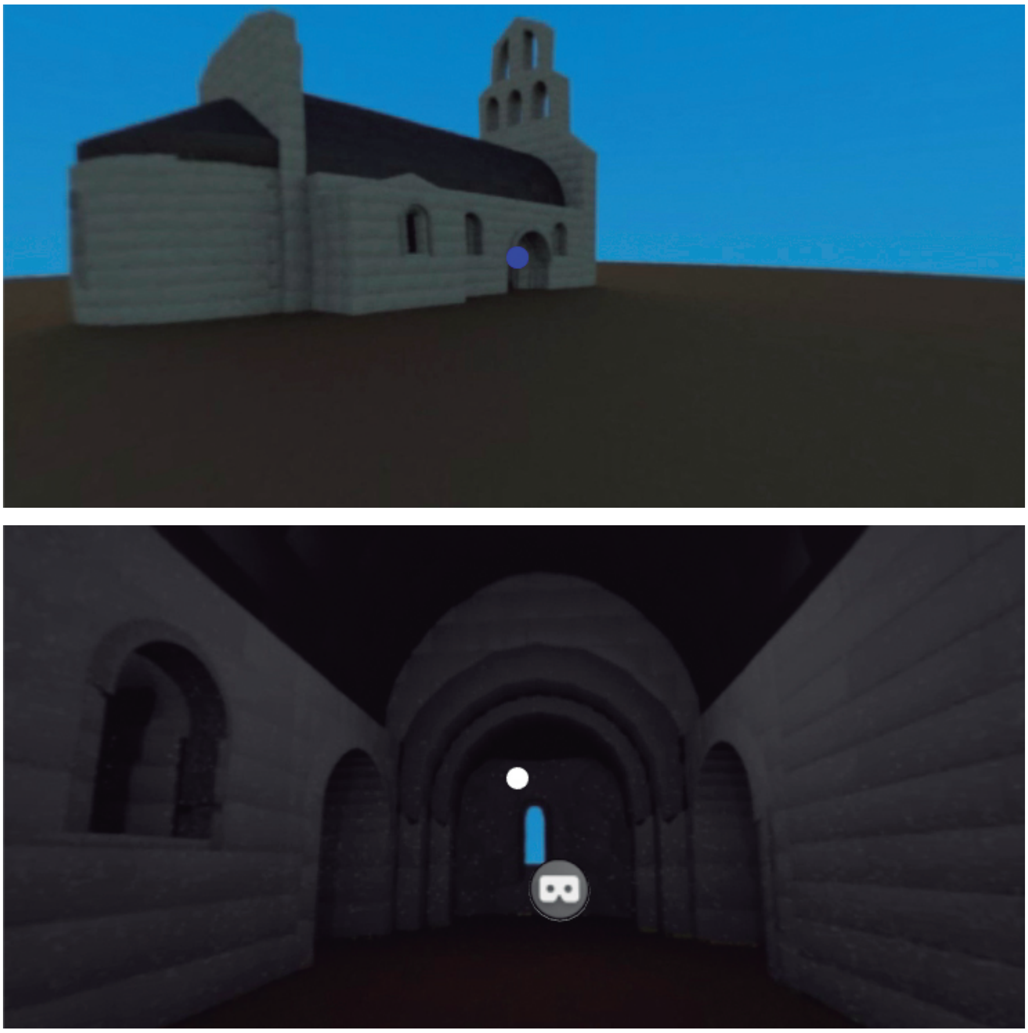


Figure 5.4: The exterior view of the church and the exterior access to the church indoor, (top). The interior church view near the door, (bottom).

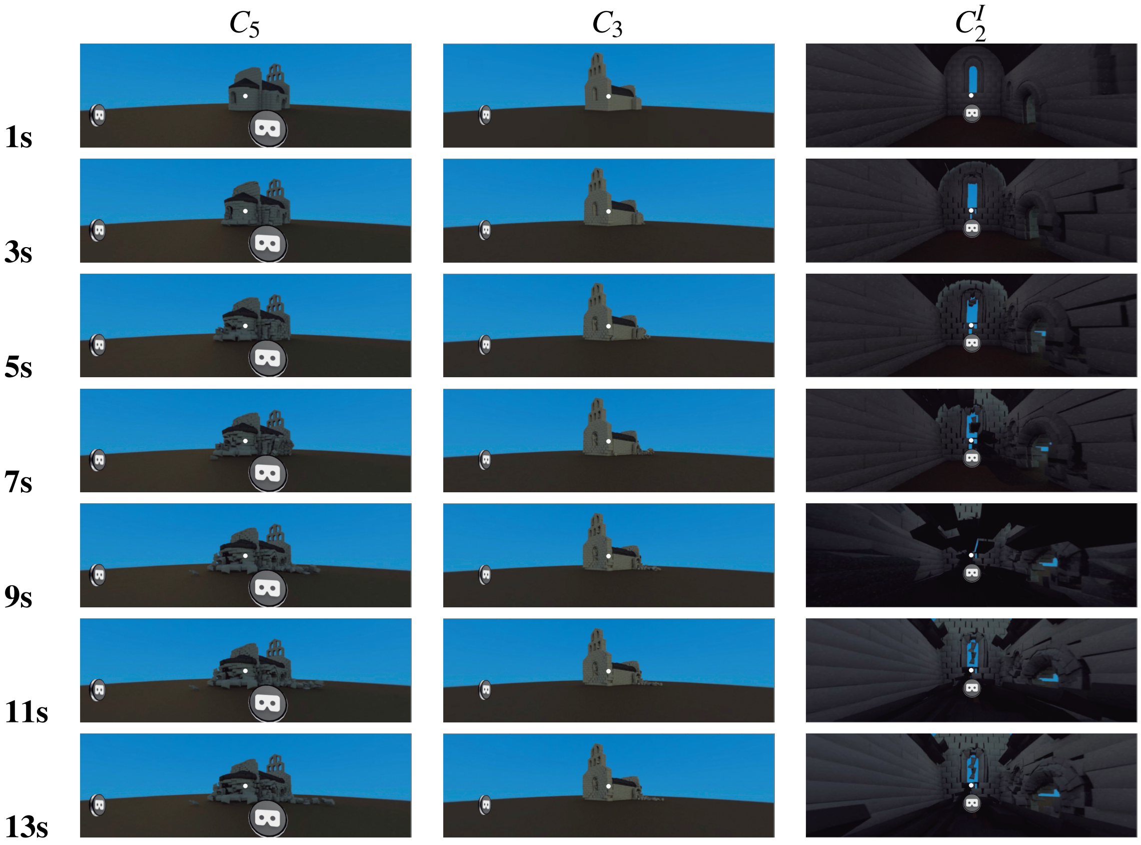


Figure 5.5: The virtual reality simulation sequence from exterior and interior graph points such as C_5, C_3 and C_2^I given as a distribution example in Figure 5.3.

5.4 Results

Our surface wave simulation implementation is based on off-the-shelf tools such as *Houdini* from SideFX [56], the Houdini Python libraries and its physics engine *Bullet* [36]. This tool allows the user to reproduce the surface wave motion at different earthquake magnitudes (see table 4.1) in an easy way. We have performed a test over a set of stone structures to verify the viability of exporting the full 3D scene into the virtual reality application. This test has been carried out on a CPU Intel-core i5-3210M with 12 Gbytes of RAM memory. As a final test for the application, and in order to validate the 360^{o} VR system, we perform a full case study of a masonry church where the rendering was computed with the same hardware and software as the first test.

	Bricks	**Time / Frame (s)**
Single Wall	45	0.4
Two Walls	90	0.6
Church	1084	1.9
Church and Two Houses	1564	4.6
Village	3496	11.9

Table 5.1: The number of bricks used and the time per frame needed for each simulation and structure. See Figure 5.6.

5.4.1 Structure Test

The purpose of this test is to analyze the efficiency of our 3D earthquake simulation scenario, computing the time spent per frame for different masonry structures. Through this test, we can see that the structural and earthquake simulation in real-time is not possible because of the high computational times involved. For the test, we set the Richter magnitude with a value of 7.0 and the *wave-front direction* pointing to the North.

We tested different ancient masonry structures such as a single wall without battlements, composed of 45 bricks; two walls, each with their respective 45 bricks; a church with 1084 bricks; a church and two houses of 1564 bricks and a village composed of a total of 3496 bricks. All these structures have been modelled with physical features such as *density* set with a value of 2691 kg/m^3 for simulating granite stone, and the *friction coefficient* with a value of 0.7, which corresponds to rock. See Figure 5.7.

The tests have been performed by testing each masonry structure for each simulation, where we have measured the number of bricks and the time spent by frame. The results are shown in Table 5.1 and plotted in Figure 5.6. They show the direct relation between the number of bricks and the time spent by the simulator to perform the computations. We

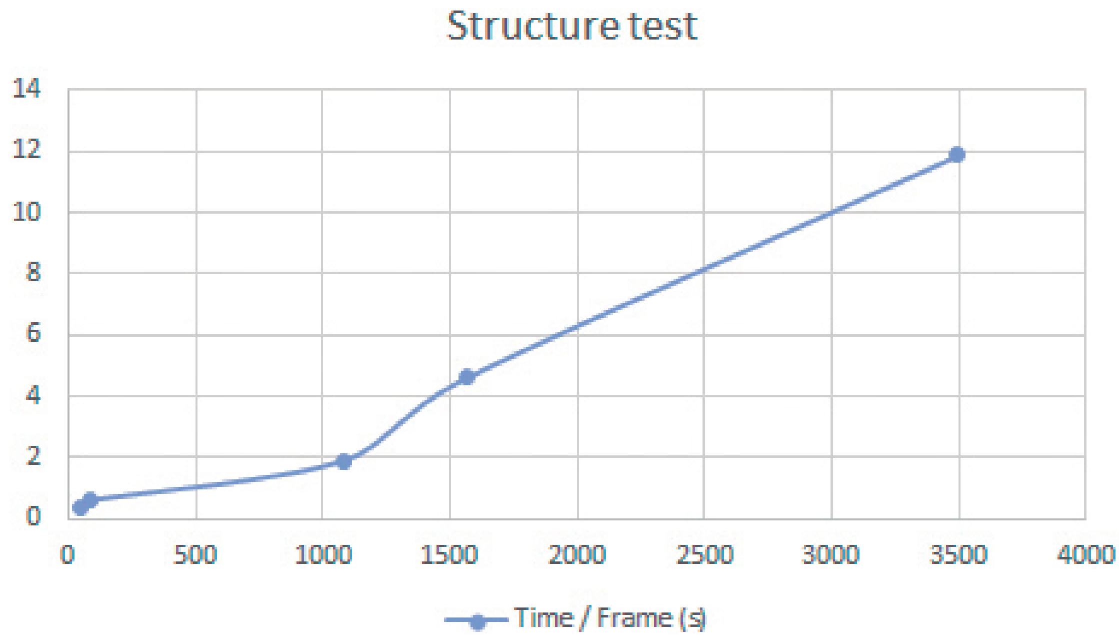


Figure 5.6: The graphical results of the tests show the time spent by frame increases in relation with the size of the masonry structure tested. See table 5.1.

can see, the time spent by frame of the simulation increases from values of $0.5s$ to $11.9s$, thus being unfeasible for the real-time requirements of virtual reality environments.

5.4.2 Full Rendering Test

	Rendering time
C_1	8.2 hrs
C_2	8.2 hrs
C_3	8.2 hrs
C_4	8.2 hrs
C_5	8.2 hrs
C_1^I	36.16 hrs
C_2^I	36.16 hrs

Table 5.2: Total computational time for each camera for the rendering process for a sequence composed of 240 frames.

The goal of this test is to evaluate the rendering process for a specific structure. For this purpose we chose the church, composed by 1084 bricks and with physical features of *density* of 2691 kg/m^3, for simulating granite stone, and the *friction coefficient* with a value of 0.7, as before.

The test has been performed by also adding the cameras located inside the simulated masonry building, following the distribution of the two graphs shown in the example in

Figure 5.3.

During the test, we have observed that the render engine has spent, on average and for the exterior graph cameras, 2 minutes and 5 seconds per frame in the rendering process. For the interior graph cameras, the same software has spent 9 minutes and 4 seconds on average. After a simulation of 240 frames, we can see the total time spent by the render engine for each camera on Table 5.2.

5.5 Results

Table 5.2 shows the total time spent by the software and hardware used in the video rendering process for each camera. It is important to remark that the times registered in this table have not any impact on the interaction and the efficiency of the virtual reality environment, because it uses only the resulting panoramic videos and not any other information from the simulation process. On the other hand, the performance of the virtual reality scenario is affected only by the time spent in loading a new video over the corresponding sphere surface, which is practically instantaneous even on the most modest hardware platform, simply because this video has been rendered in a previous stage.

Our first test, represented by Table 5.1 and Figure 5.6, shows what would happen if exporting directly the simulation into the virtual reality environment. The second test, shows that a solution to this problem is possible by designing a pipeline as the one shown in Figure 5.2, which is the base of our virtual reality application compatible with low-cost platforms.

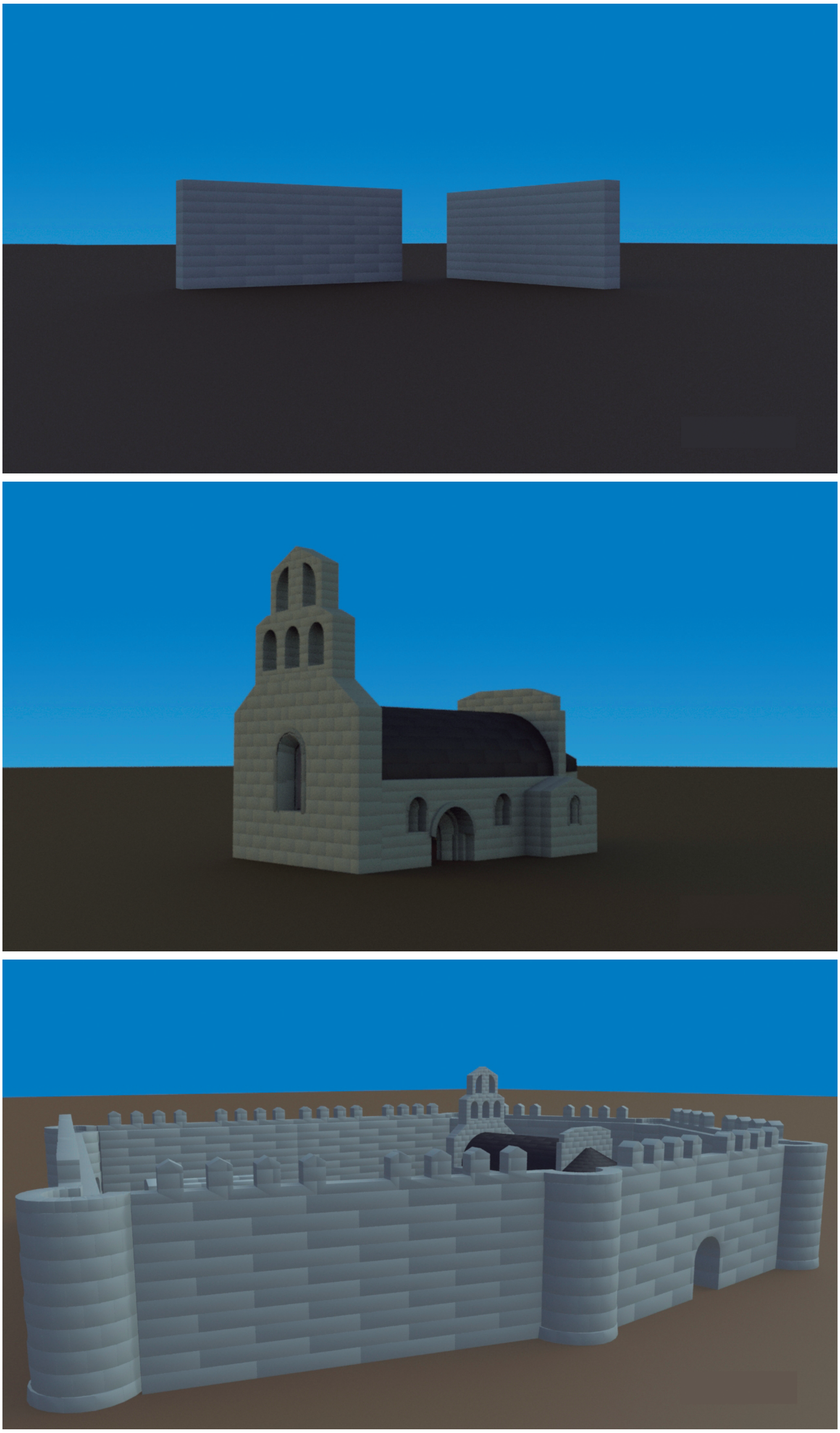


Figure 5.7: The different masonry structures tested in the earthquake simulator: Two walls, a church and a village with their respective features.

Chapter 6

Conclusions and Future Work

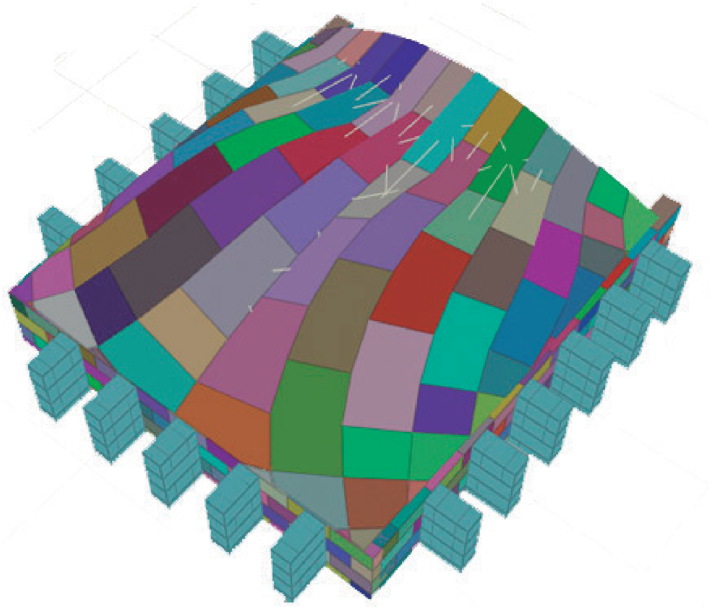


Figure 6.1: The structural stability test that results in the walls with their respective buttress.

6.1 Introduction

The review on literature has helped in the design and development of the thesis contributions for identifying and filling the gaps in the literature, and for pointing us to provide low-cost solutions affordable to all kind of cultural heritage users. After showing the test results of the contributions involved in this thesis, we present the conclusions and future work in the next sections.

6.2 Conclusions

In Section 3.2 we presented a pipeline for combining the procedural modelling of ancient masonry buildings and the structural simulation techniques. 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 or a Romanesque church from the 12th century. As a result, stakeholders (curators, art historians) will be able to better understand how the different elements of a masonry building (e.g., walls, arches, canon vaults, etc.) work in combination.

We tested the statics of a structure that combines the geometry of the walls and the geometry of an imported vault in section 3.3. We learnt how this kind of building that simulates an ancient vault structure works. We have learnt about the complex interplay of the different parameters affecting the simulation, in particular through the relationship between wall thickness and the number of buttresses. This allowed us to go a step forward and simulate the process in masonry building construction.

On the other hand, we have tested and validated the methodology proposed by Panozzo et al. [1]) and performed an in-depth structural analysis on a model resulting from their simulation, being able to validate their results under different simulation conditions, and determine the validity of their results in a generic, configurable environment, see Figure 6.1.

In Section 3.4, we presented a deeper case study of structural analysis. Given the importance of this kind of analysis in architecture, and the increasing use of procedural techniques, we decided to develop a pipeline based on off-the-shelf tools for the structural analysis of masonry buildings. We have tested the algorithm over masonry structures that combine the main elements found in a typical ancient masonry building: walls, buttresses and vaults. To test our pipeline with different settings, we created procedural walls and buttresses, but used a pre-computed roof with proven structural stability [1], and played with its constituent parameters. We have learned about the complex interplay of the different parameters affecting the simulation, in particular through the relationship among wall thickness, friction and rotational stiffness among the bricks; and structural elements like buttresses. This is important not only for entertainment purposes, but also for art historians, curators, conservators and other specialists related to ancient masonry buildings.

In Chapter 4 we have presented a methodology based on the reproduction of surface waves involved in an earthquake event. These kind of waves are the main responsible for the destruction of buildings. Following our methodology, the users can easily reproduce, at low cost, an ancient masonry building and the effects of seismic movements when they are recreating historical events about this natural phenomena that took place in the past.

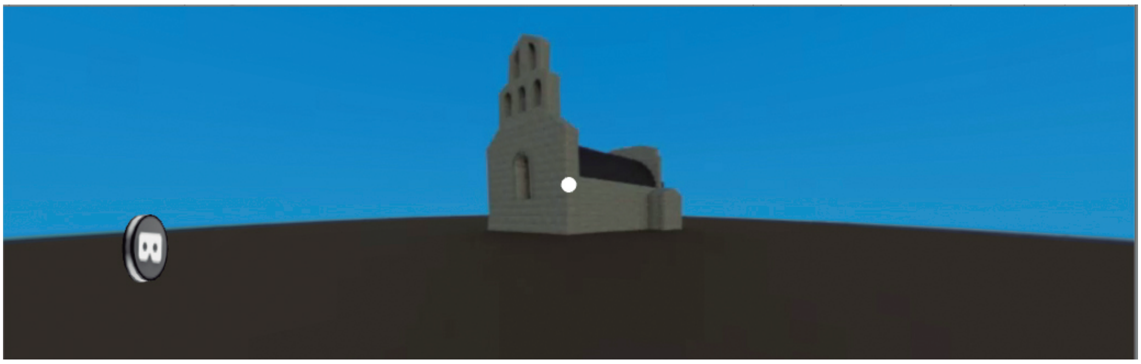


Figure 6.2: The church exterior of the Virtual Reality Application.

In Chapter 5 we have exposed the reasons of the infeasibility of the applications that have a certain geometry-complexity over low-cost platforms. We have presented a virtual reality solution based on 360^{o} VR video for giving an immersive experience of earthquake simulation over historical buildings for Cultural Heritage users, and even for the general public. This application has been designed for running on low-cost platforms, see Figure 6.2.

We think that the described contributions of this thesis are compatible with the Seville principles and fill the gap that exists in the cultural heritage literature for the recreation and diffusion of historical buildings, virtual environments and past events.

6.3 Future Work

6.3.1 Methodology Improvements

6.3.1.1 Building Modelling

We are aware that our tools and methods need some further improvements. For this reason, one possible improvement for our *tool for structural analysis*, based on a brute-force optimization algorithm, is to replace it by a better-suited optimization method, from gradient descent methods to simulated annealing [65]. Thus, through the implementation of an optimization method, our tool will become much more effective in time response and the resource consumption of the computer.

Concerning our procedural modelling tool, it could be further improved with a procedural library based on architecture styles such as Classical, Romanesque, Gothic, Baroque, Neoclassical and Modern. This library would contain masonry parts such as walls, buttresses, columns and more of all type of buildings for each architecture style. Thus, through this library, a user could select a determined architecture style, then select the type of building, civil or military, and build it step by step adding a building piece in each iteration. We think that this way, by simply assembling structural sound components, a non-expert user would not only be able to create sound masonry buildings, but also would be able to test a new hypothesis in a simple and efficient way. We hope that this library would become a reference tool among cultural heritage users for modelling all kind of historical buildings. We have thought another improvement for our procedural tool, that is the capacity of supporting 3D scanned models as an input shape.

Another possible way to improve our tool is by adding new physical features to the stone buildings such as stone degradation, brick infill of mortar, random height of the stone bricks for the same wall, or cracks effects. Also, we want our tool to be able to create a layer of plaster.

Furthermore, we would like to go a step forward and use our tools to simulate the complex construction process of a masonry building like an ancient Gothic cathedral. Also, we want to add the option for printing the model in the 3D version directly on a 3D printer.

6.3.1.2 Earthquake Simulator

A possible future work for the earthquake simulator, is to focus on its improvement, where we would like to include the possibility of reproducing the seismic secondary effects of the wave caused by the composition of the soil. Also, we want to add the possibility of putting the earthquake epicentre on the grid. Moreover, we would like to add more flexibility to the user interface, as for instance including seismic attenuation parameters.

On the other hand, we also want to improve the visualizations by adding the cracks of the walls as a result of the ground shocks. Also, we would like to make our tool more general and usable by the general public. For this, we would like to perform some sort of usability study involving heritage researchers and other stakeholders, to generate a user interface that would facilitate its usage by any lay person.

Also, we are studying to add other destructive natural phenomena for combining with our earthquake simulation, such as volcanoes. This combination allows simulating some historical events of the past, such as the destruction of the ancient Roman city of *Pompeii* [3].

We think that these improvements can be an important step to become a reference tool among cultural heritage users, in case these want to simulate a past event where an earthquake was involved.

6.3.1.3 Virtual Reality Application

For Virtual Reality applications, future work focuses on one hand on the improvement of the user interface by the inclusion of an earthquake magnitude selector and improving the presentation of additional information about the represented building. Also, we would like to assess the practicality of our system by performing a usability study that involves cultural heritage researchers among other types of users, with the aim of obtaining feedback about the application. On the other hand, we want to take advantage of the experience obtained with this virtual reality application and design a new one based on the process construction of a cathedral, as described in Section 6.3.1.1.

We think that these two applications can be a reference among non-expert users because they can learn step by step how a building of these characteristics performs under seismic movements. Also, we think that the users can learn and interact with the process followed by ancient masons when they built a church or cathedral.

6.3.2 Serious Games

Google Stadia [66] is an upcoming online platform developed by *Google* for gaming services around the world. The platform has been conceived for the users playing with all kind of devices, such as smart-phones, smart-TV or computers at high resolution without paying for extra hardware. At this moment, stadia users are planned to enjoy their games at 4k of resolution with a frequency of 60 Frames per Second. For the game developers, *Google* allows to create new games and uploading them to the platform. Because the philosophy of this thesis has some similarities with the characteristics of this platform, we have though in a near future in developing Virtual Reality-based serious games about Cultural Heritage, where the users can interact with our building recreations on all kind of devices.

Bibliography

[1] D. Panozzo, P. Block, and O. Sorkine-Hornung. “Designing Unreinforced Masonry Models”. In: *ACM Trans. Graph.* 32.4 (July 2013), 91:1–91:12. ISSN: 0730-0301.

[2] W. contributors. *Minoian civilizationWikipedia, The Free Encyclopedia*. 2019.

[3] W. contributors. *PompeiiWikipedia, The Free Encyclopedia*. 2019.

[4] W. contributors. *Buddhas of Bamyan — Wikipedia, The Free Encyclopedia*. 2019.

[5] R. Scott. *The Gothic Enterprise: A Guide to Understanding the Medieval Cathedral*. University of California Press, 2003. ISBN: 9780520939370.

[6] J. Fitchen. *The construction of Gothic cathedrals : a study of medieval vault erection /by John Fitchen*. English. Clarendon Press Oxford, 1961, xxi, 344 p.

[7] M. Henry-Claude et al. *Principles and Elements of Medieval Church Architecture in Western Europe*. Livre ouvert: Architecture. Les éditions Fragile, 2001. ISBN: 9782910685324.

[8] K. Ralls. *Gothic Cathedrals*. Nicolas-Hays, Incorporated, 2015. ISBN: 9780892541737.

[9] V. Sundstedt, A. Chalmers, and P. Martinez. “High Fidelity Reconstruction of the Ancient Egyptian Temple of Kalabsha”. In: *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*. AFRIGRAPH ’04. ACM, 2004, pp. 107–113. ISBN: 1-58113-863-6.

[10] T. Kelly and P. Wonka. “Interactive Architectural Modeling with Procedural Extrusions”. In: *ACM Trans. Graph.* 30.2 (Apr. 2011), 14:1–14:15. ISSN: 0730-0301.

[11] P. Müller et al. “Procedural Modeling of Buildings”. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 614–623. ISSN: 0730-0301.

[12] G. Patow. “User-Friendly Graph Editing for Procedural Modeling of Buildings”. In: *IEEE Computer Graphics and Applications* 32.2 (2012), pp. 66–75. ISSN: 0272-1716.

[13] City Engine. *ESRI City Engine*. https://doc.arcgis.com/en/cityengine/. 2016.

[14] L. Krecklau and L. Kobbelt. “SMI 2012: Full Interactive Modeling by Procedural High-level Primitives”. In: *Comput. Graph.* 36.5 (Aug. 2012), pp. 376–386. ISSN: 0097-8493.

[15] G. Besuievsky and G. Patow. “Procedural modeling historical buildings for serious games”. In: *Virtual Archaeology Review* 4.9 (2013), pp. 160–166. ISSN: 1989-9947.

[16] Z.-Q. Cai et al. “Interactive Iconized Grammar-Based Pailou Modelling”. In: *Computer Graphics Forum* 0.0 ().

[17] P. Musialski, M. Wimmer, and P. Wonka. “Interactive Coherence-Based FaÃ§ade Modeling”. In: *Computer Graphics Forum* 31.2pt3 (2012), pp. 661–670. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2012.03045.x`.

[18] P. Müller et al. “Procedural 3D Reconstruction of Puuc Buildings in Xkipché”. In: *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*. Ed. by M. Ioannides, D. Arnold, F. Niccolucci, and K. Mania. The Eurographics Association, 2006. ISBN: 3-905673-42-8.

[19] M Saldana and C Johanson. “Procedural modeling for rapid-prototyping of multiple building phases”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-5/W1 (Feb. 2013), pp. 205–210.

[20] K. Dylla et al. “Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques”. In: *Computer Graphics World* 16 (Jan. 2010), pp. 62–66.

[21] S. B. Juan and G. Patow. “Recreation of architectural structures using procedural modeling based on volumes”. In: *Virtual Archaeology Review* 4.9 (2013), pp. 76–81. ISSN: 1989-9947.

[22] M. Koehl and F. Roussel. “Procedural modelling for reconstruction of historic monuments”. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5/W3 (2015), pp. 137–144.

[23] V. Cappellini et al. “a Procedural Solution to Model Roman Masonry Structures”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (July 2013), pp. 149–153.

[24] E. Whiting, J. Ochsendorf, and F. Durand. “Procedural Modeling of Structurally-sound Masonry Buildings”. In: *ACM Trans. Graph.* 28.5 (Dec. 2009), 112:1–112:9. ISSN: 0730-0301.

[25] E. Whiting et al. “Structural Optimization of 3D Masonry Buildings”. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 159:1–159:11. ISSN: 0730-0301.

[26] M. Deuss et al. “Assembling Self-supporting Structures”. In: *ACM Trans. Graph.* 33.6 (Nov. 2014), 214:1–214:10. ISSN: 0730-0301.

[27] W. contributors. *Church of Sant’Agostino, Amatrice — Wikipedia, The Free Encyclopedia*. Online; accessed 13-December-2018. 2018.

[28] Geographical Association. *Italy earthquake 2016*. https://www.geography.org.uk/Italy-earthquake-2016. 2018.

[29] United States Geological Survey. *Measuring the size of an earthquake*. https://earthquake.usgs.gov. 2018.

[30] W. Lowrie. *Fundamentals of Geophysics*. 2nd ed. Cambridge University Press, 2007.

[31] N. Rawlinson. *Seismology lectures*. http://http://rses.anu.edu.au/ nick/teaching.html. 2008.

[32] Computers and Structures Spain, INC. *ETABS Software)*. https://www.csiespana.com/software/5/etabs. 2001.

[33] INACHUS Components. *Bullet Constraints Builder (BCB)*. https://www.inachus.eu/inachus-components. 2015.

[34] University of Twente. *ICT Faculty*. https://www.itc.nl/. 2015.

[35] Blender. *Open Source 3D Creation Suite*. https://www.blender.org/. 2010.

[36] Bullet Physics Library. *Bullet 2.83*. http://bulletphysics.org. 2016.

[37] A. C. Altunişik et al. “Non-destructive testing of an ancient Masonry Bastion”. In: *Journal of Cultural Heritage* 22 (2016), pp. 1049 –1054. ISSN: 1296-2074.

[38] G. Castori et al. “Seismic vulnerability assessment of a monumental masonry building”. In: *Engineering Structures* 136 (2017), pp. 454 –465. ISSN: 0141-0296.

[39] G. Fortunato, M. F. Funari, and P. Lonetti. “Survey and seismic vulnerability assessment of the Baptistery of San Giovanni in Tumba (Italy)”. In: *Journal of Cultural Heritage* 26 (2017), pp. 64 –78. ISSN: 1296-2074.

[40] M. A. Souami, M. S. Zerouala, and Y. Ait-Meziane. “The impact of building proportions in the preservation of Algiers architectural heritage against the seismic hazards”. In: *Journal of Cultural Heritage* 20 (2016). Cultural HELP 2014 Special Issue, pp. 686 –693. ISSN: 1296-2074.

[41] L. A. S. Kouris and A. J. Kappos. “Detailed and simplified non-linear models for timber-framed masonry structures”. In: *Journal of Cultural Heritage* 13.1 (2012), pp. 47 –58. ISSN: 1296-2074.

[42] M. Mosoarca and V. Gioncu. “Failure mechanisms for historical religious buildings in Romanian seismic areas”. In: *Journal of Cultural Heritage* 14.3, Supplement (2013). Science and Technology for the Safeguard of Cultural Heritage in the Mediterranean Basin, e65 –e72. ISSN: 1296-2074.

[43] V. Bosiljkov, M. Uranjek, R. Zarnić, and V. Bokan-Bosiljkov. “An integrated diagnostic approach for the assessment of historic masonry structures”. English. In: *Journal of Cultural Heritage* 11.3 (2010), pp. 239–249.

[44] L. F. Ramos and P. B. Lourenço. “Modeling and vulnerability of historical city centers in seismic areas: a case study in Lisbon”. In: *Engineering Structures* 26.9 (2004), pp. 1295 –1310. ISSN: 0141-0296.

[45] C. Uphoff et al. “Extreme Scale Multi-physics Simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC ’17. ACM, 2017, 21:1–21:16. ISBN: 978-1-4503-5114-0.

[46] N. Thalmann and G. Papagiannakis. “Virtual Worlds and Augmented Reality in Cultural Heritage Applications”. In: *EUROGRAPHICS Workshop on Graphics and Cultural Heritage*. 2005, pp. 419–430.

[47] S. Razzaque, Z. Kohn, and M. C. Whitton. “Redirected Walking”. In: *Eurographics 2001 - Short Presentations*. Eurographics Association, 2001.

[48] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. “Point &#38; Teleport Locomotion Technique for Virtual Reality”. In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY ’16. Austin, Texas, USA: ACM, 2016, pp. 205–216. ISBN: 978-1-4503-4456-2.

[49] R. Gaugne et al. “Virtual Reality (VR) interactions with multiple interpretations of archaeological artefacts”. In: *EG GCH 2018 - 16th EUROGRAPHICS Workshop on Graphics and Cultural Heritage*. Vienna, Austria, 2018, pp. 1–9.

[50] E. Selmanovic et al. “VR Video Storytelling for Intangible Cultural Heritage Preservation”. In: *Eurographics Workshop on Graphics and Cultural Heritage*. Ed. by R. Sablatnig and M. Wimmer. The Eurographics Association, 2018. ISBN: 978-3-03868-057-4.

[51] G. D. Gasperis, A. Cordisco, and F. Cucchiara. “Immersive Virtual Reality As A Resource For Unaccessible Heritage Sites”. In: *IOP Conference Series: Materials Science and Engineering* 364 (2018), p. 012035.

[52] C. Andujar et al. “VR-assisted Architectural Design in a Heritage Site: the Sagrada Família Case Study”. In: *Eurographics Workshop on Graphics and Cultural Heritage*. Ed. by R. Sablatnig and M. Wimmer. The Eurographics Association, 2018. ISBN: 978-3-03868-057-4.

[53] M. Carrozzino et al. “COMPARING INNOVATIVE XR SYSTEMS IN CULTURAL HERITAGE. A CASE STUDY”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W11 (May 2019), pp. 373–378.

[54] M. K. Bekele et al. “A Survey of Augmented, Virtual, and Mixed Reality for Cultural Heritage”. In: *J. Comput. Cult. Herit.* 11.2 (2018), 7:1–7:36. ISSN: 1556-4673.

[55] M. Mortara et al. “Learning cultural heritage by serious games”. In: *Journal of Cultural Heritage* 15 (May 2014), pp. 318–325.

[56] SideFX. *Houdini 16*. http://www.sidefx.com. 2012.

[57] Viquipèdia. *Sant Bartomeu de Pincaró — Viquipèdia, l’enciclopèdia lliure*. 2019.

[58] Mosek ApS. *Mosek library*. https://www.mosek.com. 1997.

[59] Unity. *Unity 3D engine*. unity3d.com. 2005.

[60] Oculus Rift. *Oculus Rift*. oculus.com. 2010.

[61] HTC Vive. *HTC Vive*. vive.com. 2010.

[62] Windows Mixed Reality. *Windows Mixed Reality*. https://www.microsoft.com/en-us/windows/windows-mixed-reality. 2018.

[63] Google VR. *Google VR*. https://vr.google.com. 2018.

[64] TrustGXT. *Trust GXT Virtual Reality Glasses*. https://www.trust.com/es/product/21322-gxt-720-virtual-reality-glasses. 2018.

[65] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd ed. New York, NY, USA: Cambridge University Press, 2007. ISBN: 0521880688, 9780521880688.

[66] Google Stadia. *Google Stadia Platform*. https://stadia.dev/. 2019.

Appendix A

Surface waves

A.1 Rayleigh waves

In this appendix, we give the mathematical description of a *Rayleigh wave* as a wave that propagates in the *x-axis*, and his displacement depends of a function based on depth defined by $f(z)$ that can be described as:

$$\theta(x,z,t) = f(z)e^{i(kx-\omega t)}, \tag{A.1}$$

where k is the wave vector, ω is the angular frequency and t is the time. If $\theta(x,z,t)$ has a phase velocity of $c = \alpha$ or $c = \beta$, we can make the substitution of terms into Equation 4.3 that is the wave equation for longitudinal or transverse wave and obtain an equation described as:

$$-\omega^2 f(z) = c^2[-k^2 f(z) + f'(z)], \tag{A.2}$$

where the Equation A.2 has the following solution described as:

$$f(z) = c_1 e^{(k_c z)} + c_2 e^{(-k_c z)}. \tag{A.3}$$

Equation A.3 describes an exponential function, $f(z)$, where c_1 and c_2 are constant. Thus, the propagation of the *Rayleigh waves* have a velocity through the surface slower than *P-waves* and *SV-waves* where the propagation of the *Rayleigh wave* can be described through potential functions.

$$\phi = -iae^{[k_\alpha z+i(kx-\omega t)]} \quad \psi = be^{[k_\beta z+i(kx-\omega t)]}, \tag{A.4}$$

as the *Rayleigh waves*, have not the *SH-waves* component, from the Equation A.4, we can consider the b vector as scalar. The relative amplitudes a and b are determined by the free surface, $z = 0$. Thus, We can describe the displacements related as the potentials in the *x-axis* as:

$$\theta(x,z,t) = \nabla\phi + \nabla x\psi, \tag{A.5}$$

where $\theta_x(x,z,t)$ and $\theta_z(x,z,t)$ are described as:

$$\theta_x(x,z,t) = kae^{[k_\alpha z+i(kx-wt)]} - k_\beta be^{[k_\beta z+i(kx-\omega t)]}, \tag{A.6}$$

$$\theta_z(x,z,t) = -i\kappa_\alpha ae^{[k_\alpha z+i(kx-wt)]} + ikbe^{[k_\beta z+i(kx-\omega t)]}, \tag{A.7}$$

from Equation A.6 and Equation A.7, we can obtain the relation of the tangential stress condition σ_{xz}:

$$\sigma_{xz} = \mu\left(\frac{\partial\theta_x}{\partial z} + \frac{\partial\theta_z}{\partial x}\right) = 0, \tag{A.8}$$

where $\frac{\partial\theta_x}{\partial z}$ can be described as:

$$\frac{\partial\theta_x}{\partial z} = k\kappa_\alpha ae^{[k_\alpha z+i(kx-wt)]} - k_\beta^2 be^{[k_\beta z+i(kx-\omega t)]}, \tag{A.9}$$

and $\frac{\partial\theta_z}{\partial x}$ can be described as:

$$\frac{\partial\theta_z}{\partial x} = k\kappa_\alpha ae^{[k_\alpha z+i(kx-wt)]} - k^2 be^{[k_\beta z+i(kx-\omega t)]}, \tag{A.10}$$

if we make the substitution of the terms of the Equation A.9 and Equation A.10 into Equation A.8, we obtain an equation for σ_{xz} that can be described as:

$$2K\kappa_\alpha a - \left(k^2 + \kappa_\beta\right)b = 0, \tag{A.11}$$

once we have deduced the tangential stress condition for the free surface, $z = 0$, we can describe the normal stress condition σ_{zz} as:

$$\sigma_{zz} = \lambda\left(\frac{\partial\theta_x}{\partial x} + \frac{\partial\theta_z}{\partial z}\right) + 2\mu\frac{\partial\theta_z}{\partial z} = 0, \tag{A.12}$$

where $\frac{\partial\theta_x}{\partial x}$ is described as:

$$\frac{\partial\theta_x}{\partial x} = ik^2 ae^{[k_\alpha z+i(kx-\omega t)]} - ik\kappa_\beta be^{[k_\beta z+i(kx-\omega t)]}, \tag{A.13}$$

and $\frac{\partial\theta_z}{\partial z}$ can be described as:

$$\frac{\partial\theta_z}{\partial z} = -i\kappa_\alpha^2 a e^{[k_\alpha z + i(kx-\omega t)]} + ik\kappa_\beta b e^{[k_\beta z + i(kx-\omega t)]}, \tag{A.14}$$

thus, for $\rho\alpha^2 = \lambda + 2\mu$ and $\rho\beta^2 = \mu$, we make the substitution of the terms into Equation A.12, we obtain an equation for the normal stress, σ_{zz}, that can be described as:

$$\left(\alpha^2 - 2\beta^2\right)\frac{\partial\theta_x}{\partial x} + \alpha^2\frac{\partial\theta_z}{\partial z} = 0, \tag{A.15}$$

if we make the substitution of the terms of the Equation A.13 and Equation A.14 into the Equation A.15, we obtain an equation for σ_{zz} that can be described as:

$$\left(K^2 + \kappa_\beta^2\right)a - 2k\kappa_\beta b = 0, \tag{A.16}$$

the Equation A.11 and Equation A.16 are a pair of homogeneous equations whose solution is the dispersion relation where the terms κ_α and κ_β have been substituted. The equation can be described as:

$$\left(1 - \frac{\omega^2}{2k^2\beta^2}\right)^4 = \left(1 - \frac{\omega^2}{k^2\alpha^2}\right)\left(1 - \frac{\omega^2}{k^2\beta^2}\right), \tag{A.17}$$

the substitution of terms into Equation A.17, where $\zeta = \omega^2/k^2\beta^2$ and $\eta = \beta^2/\alpha^2$, give the equation of the *Rayleigh waves* speed that can be described as:

$$\zeta^3 - 8\zeta^2 + 8\zeta(3 - 2\eta) - 16(1 - \eta) = 0 \tag{A.18}$$

Thus, for a *Poisson solid*, where $\lambda = \mu$, give the phase velocity independent of the frequency of the *Rayleigh waves* as:

$$C_R = \frac{\omega}{k} = \beta\sqrt{0.8453}, \tag{A.19}$$

through Equation A.11 and Equation A.16, we can obtain the relative amplitude for the *P-waves* and *SV-waves* when the dispersion relation is satisfied. Thus, the position equation for the *Rayleigh waves* for *x-axis* and *z-axis* respectively at the free surface, $z = 0$ are:

$$\theta_x(x,t) = a\left(\frac{\omega^2}{2k\beta^2}\right)cos\left(kx - \omega t\right), \tag{A.20}$$

$$\theta_z(x,t) = a\left(\frac{2k\kappa_\alpha}{k^2 + \kappa_\beta^2}\right)\left(\frac{\omega^2}{2k\beta^2}\right)sin\left(kx - \omega t\right), \tag{A.21}$$

where $\kappa_\alpha^2 = k^2 - (\omega/\alpha)^2$ and $\kappa_\beta^2 = k^2 - (\omega/\beta)^2$.

Appendix B

Surface waves

B.1 Love waves

The mathematical description of the *Love waves*, whose motion is given by *SH-waves* and requires a velocity that varies with the depth. Thus, like *Rayleigh waves*, the *Love waves* are dependent of a depth function , $f(x)$, whose solution is a depth exponential function that can be described as:

$$f(x) = Ae^{(K_l z)} + Be^{(-k_l z)} \tag{B.1}$$

But, from the Equation B.1 and in the case of *Love waves* all the solutions for a plane layer or a half-space are present. Thus, the propagation in the $x - axis$ direction for a plane layer can be expressed as:

$$\theta_y(x,z,t) = [Ae^{(\kappa_1 z)} + Be^{(-\kappa_1 z)}]e^{[i(kx-\omega t)]}, \tag{B.2}$$

the propagation in the $x - axis$ direction for a half-space where the growing exponential has been omitted, can be expressed as:

$$\theta_y(x,z,t) = Ce^{(-k_2 z)}e^{[i(kx-\omega t)]}, \tag{B.3}$$

where κ_1 and κ_2 are the vertical scale components. The equations that give the solution for these components are:

$$\kappa_1^2 = k^2\left(1 - \frac{c_L^2}{\beta_1^2}\right) \quad and \quad \kappa_2^2 = k^2\left(1 - \frac{c_L^2}{\beta_2^2}\right), \tag{B.4}$$

the free stress at the upper surface, $z = 0$, is given by $\sigma_{zy} = 0$. Therefore, the Equation B.2

of propagation for a plane layer can be rewritten as:

$$\frac{\partial \theta_y}{\partial z} = \kappa_1 [Ae^{(\kappa_1 z)} + Be^{(-\kappa_1 z)}]e^{[i(kx-\omega t)]}, \tag{B.5}$$

and for the half-space the Equation B.3 is rewritten as:

$$\frac{\partial \theta_y}{\partial z} = \kappa_2 Ce^{(-k_2 z)}e^{[i(kx-\omega t)]}, \tag{B.6}$$

from the Equation B.5 and Equation B.6 the displacement of θ_y at the interface $z = H$ can be written as:

$$2A\cos(i\kappa_1 H) - Ce^{(-\kappa_2 H)} = 0 \tag{B.7}$$

Thus, the shear stress at the interface $z = H$ where $\sigma_{yz} = 2\mu e_{yz}$ is written through the Equation B.5 and Equation B.6 as:

$$2Ai\mu_1\kappa_1 \sin(i\kappa_1 H) - \mu_2 C\kappa_2 e^{(-\kappa_2 H)} = 0, \tag{B.8}$$

the Equation B.7 and Equation B.8 are a system of homogeneous equations whose solution can be expressed as:

$$\tan(i\kappa_1 H) = \frac{\mu_2\kappa_2}{i\mu_1\kappa_1}, \tag{B.9}$$

if we take the κ_1 and κ_2 values from the Equation B.4 and makes the substitution of terms into the Equation B.9. Indeed, if we introduce the variable for the phase velocity C_L given by $\zeta = H\sqrt{\beta_1^{-2} - C_L^{-2}}$, we obtain the equation of the dispersion relation for the *Love waves* as:

$$\tan(\omega\zeta) = \frac{\mu_2}{\mu_1}\frac{\sqrt{H^2(\beta_1^{-2} - \beta_2^{-2} - \zeta^2)}}{\zeta}, \tag{B.10}$$

the Equation B.10 shows that the phase velocity C_L depends on the frequency ω for the *Love waves*. Thus, the position equation as a sum of two harmonic waves, can be described as a product of two cosine functions as:

$$\theta(x,t)_Y = a2\cos(kx - \omega t)\cos(\delta kx - \delta\omega t) \tag{B.11}$$