CEIG – Spanish Computer Graphics Conference (2019)
A. Jarabo and D. Casas (Editors)

A Virtual Reality Front-End for Earthquake Simulation

Josep Lluis Fita, Gonzalo Besuievskyand Gustavo Patow

ViRVIG, University of Girona, Spain

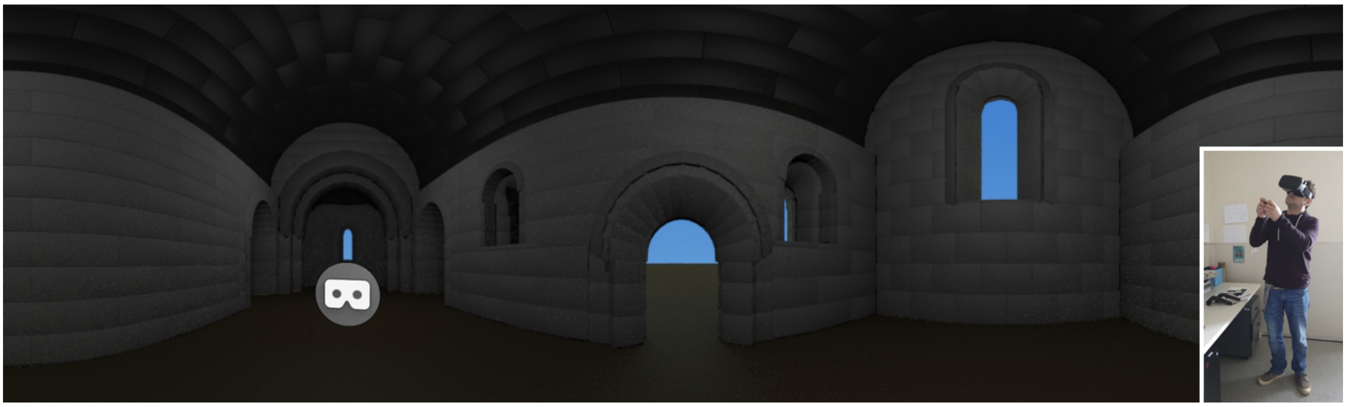

Figure 1: *A spherical view of the Romanesque church interior used in our virtual reality application, see the inset. Inset: a user testing it with the headset and controller.*

Abstract

Virtual Reality has traditionally been used in Cultural Heritage for giving to the final user an immersive experience over recreated scenarios, which usually have been designed and focused on static environment recreation. In spite of its importance for cultural heritage, we have found a lack of virtual reality applications to recreate structural and seismic simulations on historical buildings. In this paper we describe a low-cost virtual reality solution, affordable for all kinds of users that own a smart-phone. Through our application, the users can have an immersive experience that combines the ancient building recreation, its structural simulation and the natural phenomena simulation like earthquakes.

CCS Concepts

•***Computing methodologies*** → ***Physical simulation;*** •***Applied computing*** → *Virtual Reality;*

1. Introduction

Over the last years, research on new techniques in Computer Graphics has been of great benefit to fields like video-games, film and especially cultural heritage. It is on this last field where the efforts have been focused on the digital preservation of artifacts and architectonic structures through the development of methodologies that recreate ancient buildings as simple 3D textured objects. However, only a few of these research efforts have been focused on the combination of simulation of natural phenomena, like earthquakes, and 3D modelling of historical buildings. On the other hand, the application of Virtual Reality techniques to the Cultural Heritage field has given to the researchers a new way for the diffusion of past events and historical building recreation with 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 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

DOI: 10.2312/ceig.20191204

low-cost platforms and virtual reality applications with a certain geometry-complexity.

In this paper, we have focused on solving the presented issue by exporting the simulation in 360°camera video 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. Our 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 device like a smart-phone with headsets.

2. Previous Work

Ancient building recreation, Structural and Earthquake simulation: Virtual building recreation is not an easy enterprise in the Cultural Heritage, which is particularly true for reconstructions of the architecture of the middle ages, especially in civil buildings like Romanesque churches. There exist some efforts [Fit61, HCSZ*01] about the creation of methodologies used by ancient masonry buildings, which is very helpful to understand the construction process of the different parts of these buildings.

One common approach for the recreation of ancient buildings is the use of procedural modelling techniques, such as the one presented by Muller et al. [MWH*06] using shape grammars. Later, following their steps, other authors have created extensions based on shape grammars such as as the visual language by Patow [Pat12], the extended expressions by Krecklau and Kobbelt [KK12], or the inclusion of primitives as first-class citizens by Schwarz and Müller [SM15]. For more information, please refer to the survey by Musialski et al. [MWA*13].

On the other hand, Saldana and Johanson [SJ13], presented a technique based on the virtual reconstruction of a building from GIS data. On the recreation of historical buildings as simple 3D objects, Capellini and co-workers [CSS*13] designed a technique for virtual recreation of Roman masonry structures. However, all these approaches only focus on the recreation of the exterior of the building, and do no combine it with any kind of structural simulation.

With respect to the techniques that combine building shapes and structural simulation, we can mention the work of Whiting and co-authors [WOD09], where they recreate an historical building like the *Cluny Abbey*, and apply a structural analysis based on shocks on the ground for studying the stability of the building. Later, Whiting et al. [WSW*12] based the building recreation through restrictions introduced by the users. After that, Panozzo and co-authors [PBSH13] presented an approach where a 3D structure is created from an input shape. After, Deuss et al. [DPW*14] extended this work for all kind of masonry shapes. More recently, and more specifically in the cultural heritage field, Fita and co-authors [FBP17] developed a methodology based on structural analysis. However, all these works and techniques presented were designed only for structural simulation, with the objective of analyzing and/or modifying the building geometry according to the users needs, but not taking historical requirements into account, and also disregarding computation times (their objective was analysis, not real-time performance).

On the other hand, the combination of the two fields (i.,e., structural simulation and earthquake simulation) has lead to some interesting works from the civil engineering field, as the one by Altunisik and co-workers [AAG*16], where the authors studied the behaviour of historical buildings under seismic conditions. Other similar works developed are the ones presented by Castori and colleagues [CBM*17], Fortunato and co-authors [FFL17] and Souami and colleagues [SZAM16]. Also in the civil engineering domain, we can mention two interesting works. The first, developed by Ramos and Lourenco [RL04], presented a methodology for rebuilding an historical city center situated in a seismic area. The second is the reproduction, with a great accuracy, of the Sumatra Earthquake, developed by Uphoff and colleagues [URB*17]. Recently, Fita and co-authors [FBP18], in the cultural heritage field, simulated the effect of an earthquake over masonry structures like churches, medieval walls and castles, at a large computational cost.

Virtual Reality and cultural heritage: Virtual Reality is used in the cultural heritage field with the aim of giving an immersive and realistic experience to the users. In the literature, we can find some approaches such as the work developed by Magnenat-Thalmann and Papagiannakis [TP05], where animated humans were presented over a recreation of ancient Pompeii. Recently, Gaugne and co-workers [GSN*18] presented a virtual reality methodology that shows the 3d artefacts as user interfaces. A research about the immersive experience given by 360°VR videos in cultural heritage was presented by Selmanovic et al. [SRH*18]. De Gasperis and co-workers [GCC18], presented a virtual reality methodology for the reconstruction of a building affected by an earthquake through the combination of 3D and photo-realistic modeling. Recently, Andujar and co-authors [ABB*18] introduced a VR-based project that uses CAD tools to model, together with VR tools to review, new structures for the Sagrada Family Cathedral to be able to evaluate these new elements in a proper context. For more information, please refer to the review by Mortara and co-workers [MECB*14] or the survey presented by Bekele and colleagues [BPF*18].

3. Seismology

We can describe an Earthquake as a movement that occurs between the tectonic plates inside the crust of the earth. The liberated energy travels inside of the earth through the medium in two ways. *Longitudinal* waves, whose movement by compression is back and forth and *transverse* waves, whose movement occurs by vibration in solid particles. When these two movements, known as body waves, reach the surface of the earth, create the surface waves which propagate along the earth's surface from the earthquake epicenter. The main features of the surface waves are the lowest velocity with respect to the body waves and the amplitude that becomes lower with the increment of the medium depth. These surface waves are the main responsible for the environment destruction and we can classify these according to their motion.

Here we will present only the basics of the surface waves formulation, a deeper description about general earthquake phenomena can be found elsewhere [FBP18], following the mathematical fundamentals of Lowrie [Low07] and Rawlinson [Raw08].

Rayleigh waves: are waves whose motion is a combination of the two types of body waves. The mathematical description of these waves is given by the position equation for the x-axis and z-axis at the free surface, assuming the wave travels in the y-axis.

$$\theta_x(x,t) = a\left(\frac{\omega^2}{2k\beta^2}\right)\cos(kx-\omega t), \tag{1}$$

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

where $\kappa_\alpha^2 = k^2-(\omega/\alpha)^2$ and $\kappa_\beta^2 = k^2-(\omega/\beta)^2$. Equation 1 and Equation 2, show the main features such as the angular frequency ω and wave-number k of the *Rayleigh waves* and describe the motion of these as retrograde and elliptical parallel to the direction of propagation.

The Love waves: The motion of these waves comes from the transverse body waves. These are faster than *Rayleigh* waves and travel along the Earth surface in groups of waves, described by the so called called *carrier* and *envelope* waves. The mathematical description of these waves is given by the position equation as a sum of two harmonic waves that can be described as a product of two cosine functions.

$$\theta_y(x,t) = a2\cos(kx-\omega t)\cos(\delta kx-\delta\omega t). \tag{3}$$

Equation 3 describes the *Love* wave position in a given time, where the carrier wave has an angular frequency ω and wave-number k and the envelope wave has a lower angular frequency $\delta\omega$ and wave-number δk.

We based our earthquake simulation in the reproduction of the surface waves following the mathematical description given by Equations 1 and 2 for *Rayleigh* waves and Equation 3 for the reproduction of the *Love* waves. The Earthquake magnitudes have been modeled by Equation 4, which is an adaptation of the original Richter formula [Raw08] as:

$$M = \lg A + 3\lg(8\Delta t) - Q, \tag{4}$$

where A is the amplitude recorded from the seismograph, Δt is the time between the two type of body waves, Q is the regional scaling factor and M is the magnitude number of energy released by the earthquake. In table 1 we show different intensities according the Richter Scale.

4. 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 equations 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.

	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 1: *The Richter magnitude scale that measures the energy released from a seismic event.*

In this paper 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 2. In this pipeline we can see the *Simulation* is computed first in a high-performance computer, rendered (with any selected renderer) 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.

5. Earthquake Simulation and Visualization

5.1. Earthquake Simulation

Our earthquake simulation tool uses the following elements, presented following a bottom-up perspective:

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 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 (in our case, 64) 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* [Lib16] solver library. See Section 6 for more details about the exact implementation.

An input geometry of an ancient masonry building, previously modeled, is imported in the simulator and connected through a dynamic network with the positions of the ground being controlled

Figure 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.*

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 modeled 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. 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 1 and 2 for simulating the *Rayleigh* motion, and Equation 3 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.

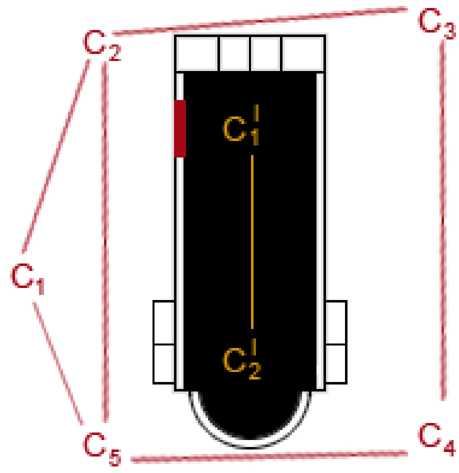

Figure 3: *The church map and the point-camera location has been distributed as example mode. In red, the exterior point-cameras. In orange, the interior point-cameras. The red rectangle is the linking point between the exterior and the interior of the church.*

5.2. Spherical Video

The implementation of the surface wave tool previously described allowed us to obtain 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° VR videos, which will allow us to display the simulation in real-time on a low-cost device, as shown in Figure 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 3. Also, each camera has been configured with a specific capture angle for rendering with a realistic *ray-tracing* algorithm, using an *equirectangular* projection, mono , see Figure 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 supports better lighter files.

Once the set of 360° 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 on the pipeline in Figure 2.

5.3. Virtual Reality App

In this project we have used the *Unity 3D engine* [Uni05], 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* [Rif10], *HTC Vive* [Viv10], or *Windows Mixed Reality* [Rea18], and also for the cheapest Head Mounted Displays for mobile VR, like *Google*

Daydream or Cardboard [VR18]. We decided to develop our cultural heritage application using *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 the *three-degrees-of-freedom* (3-DOF), fixed viewpoint position, for interacting with the 3D space. The video controller of the Head Mounted Display, *Trust GXT 720* [Tru18], 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°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 6.

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 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. See the supplemental video.

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 the first node of the graph represented by C_i, automatically the appropriate 360°VR video and neighbour nodes are loaded in the scenario. At this point, the user can interact with the 360°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 occur with all the nodes that conforms the graph and the graph located inside the church. This way, although we are dealing with 360°videos, we solve the visibility problem of hiding the nodes *behind* the building, which should not be been though its structure.

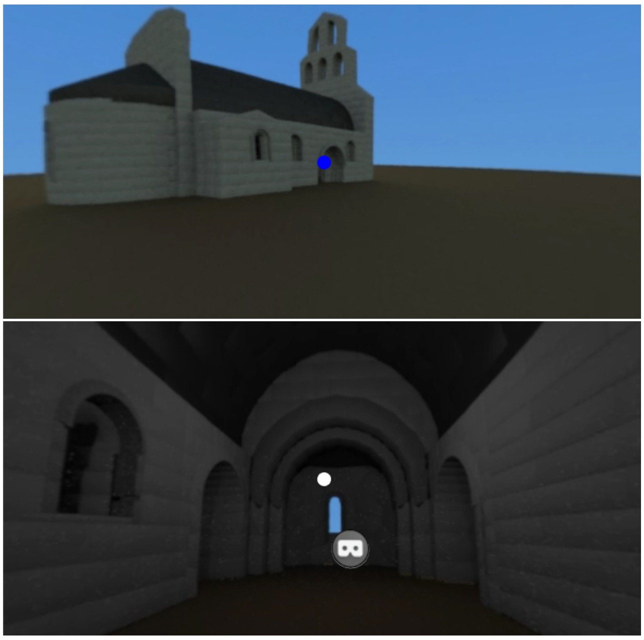

Figure 4: *Above, the exterior view of the church and the exterior access to the church indoor. Below, The interior church view near the door.*

An example of a graph is given in Figure 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 navigating through the exterior or interior graph and see the simulation from different points of view, see Figure 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 hide one graph or another, and presented in the exterior or interior church view, according to the predefined authored interactions. See Figure 4.

6. Results

Our surface wave simulation implementation is based on off-the-shelf tools such as *Houdini* from SideFX [Sid12], the Houdini Python libraries and its physics engine *Bullet* [Lib16]. This tool allows the user to reproduce the surface wave motion at different earthquake magnitudes (see table 1) in an easy way. We have performed a test over a set of stone structures for verifying the viability of exporting the 3D full 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. Finally, as a final test for the application and in order to validate the 360°VR system, we perform a full

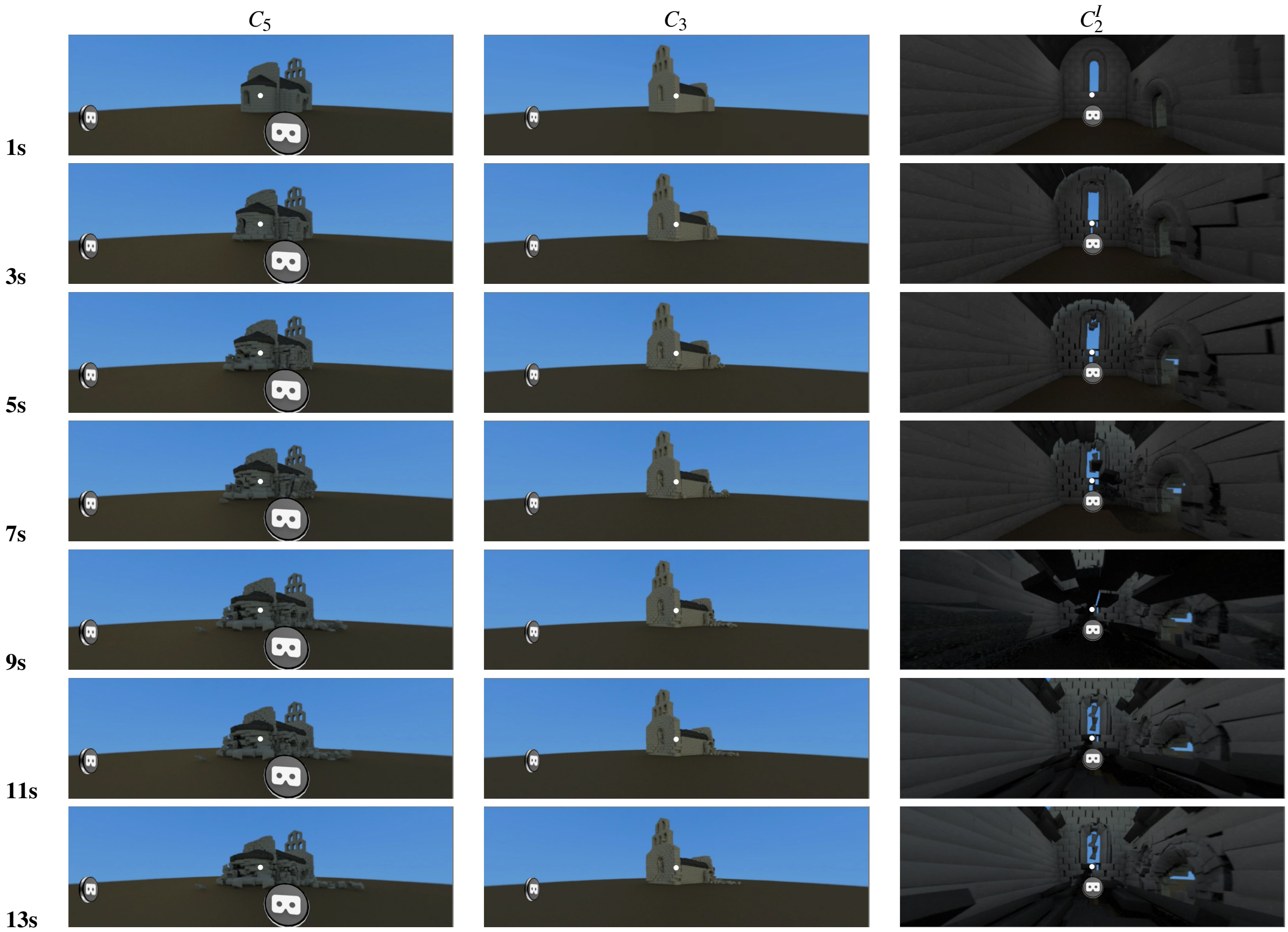

Figure 5: *The virtual reality simulation sequence from exterior and interior graph points such as C_5, C_3 and C_2^I given as a distribution example in Figure 3.*

case study of a masonry church where the rendering was computed with the same hardware and software as the first test.

6.1. 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 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 6.

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 2 and plotted in Figure 7. They show the direct relation between the number of bricks and the time spent by the simulator to perform the computations. The time spent by frame of the simulation increases exponentially from values of $0.5s$ to $11.9s$, thus being unfeasible for the real-time requirements of virtual reality environments.

6.2. Full Rendering Test

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.

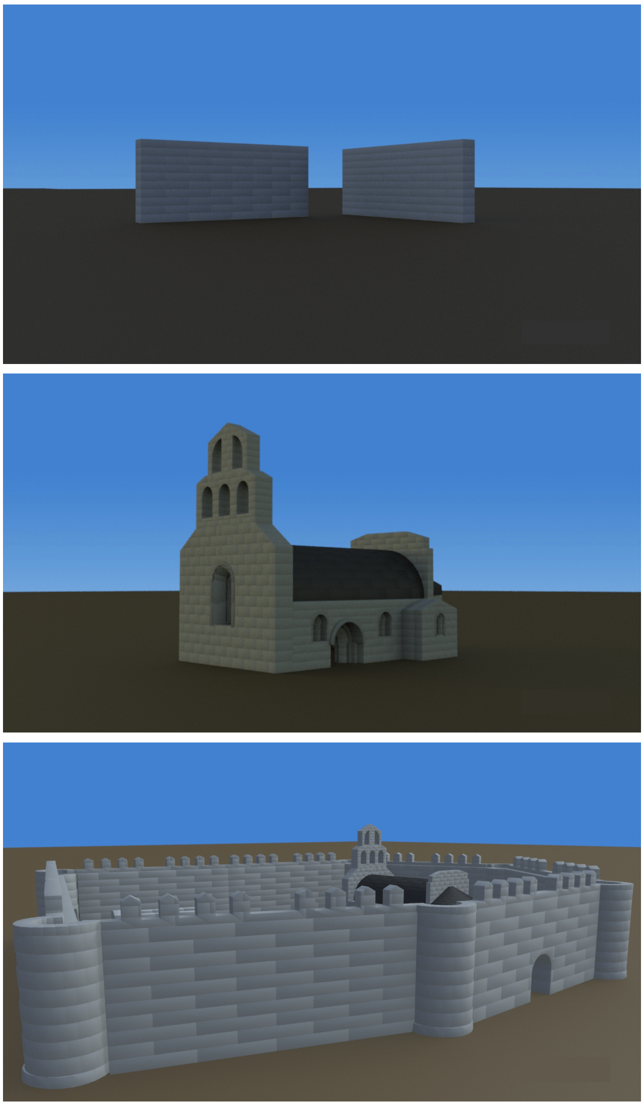

Figure 6: *The different masonry structures tested in the earthquake simulator: Two walls, a church and a village with their respective features.*

	Bricks	Time / Frame (s)
Single Wall	45	0.5
Two Walls	90	0.6
Church	1084	1.9
Church and Two Houses	1564	4.6
Village	3496	11.9

Table 2: *The number of bricks and the time per frame for each simulation and structure. See Figure 7.*

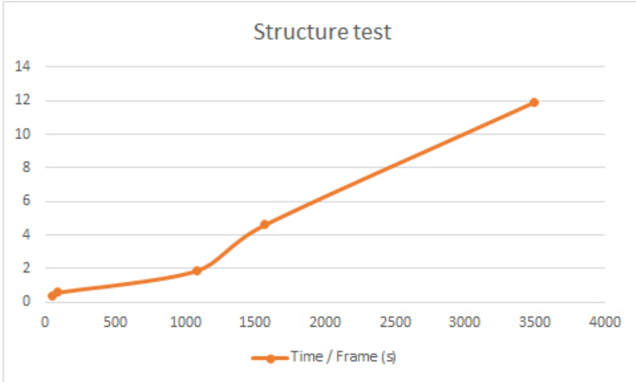

Figure 7: *The graphical results of the tests show the time spent by frame increases in relation with the brick number of the masonry structure tested. See table 2.*

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

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 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 3.

7. Discussions

Table 3 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.

The first test, represented by Table 2 and Figure 7, 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 2, which is the base of our virtual reality application compatible with low-cost platforms.

8. Conclusions and Future Work

We have presented a low-cost virtual reality solution based on earthquake simulation events, which is an affordable VR solution for users such as historians, art historians, curators and even the general public, designed to run on low cost platforms.

Our future work focuses 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 want to improve the church rendering in a realistic view, including simulating a realistic appearance of granite stone. Finally, we would like to assess the practicality of our system by performing two studies, the first one for obtaining user feelings and impressions of the system usage, and the second is a usability study that involves cultural heritage researchers among other types of users, with the aim of obtaining technical feedback about the application.

Acknowledgments

This work was partially funded by the TIN2017-88515-C2-2-R project from Ministerio de Ciencia, Innovación y Universidades, Spain.

References

[AAG*16] ALTUNIŞIK A. C., ADANUR S., GENÇ A. F., GŬNAYDIN M., OKUR F. Y.: Non-destructive testing of an ancient masonry bastion. *Journal of Cultural Heritage 22* (2016), 1049 – 1054. 2

[ABB*18] ANDUJAR C., BRUNET P., BUXAREU J., FONS J., LAGUARDA N., PASCUAL J., PELECHANO N.: VR-assisted Architectural Design in a Heritage Site: the Sagrada Família Case Study. In *Eurographics Workshop on Graphics and Cultural Heritage* (2018), Sablatnig R., Wimmer M., (Eds.), The Eurographics Association. 2

[BPF*18] BEKELE M. K., PIERDICCA R., FRONTONI E., MALINVERNI E. S., GAIN J.: A survey of augmented, virtual, and mixed reality for cultural heritage. *J. Comput. Cult. Herit. 11*, 2 (Mar. 2018), 7:1–7:36. 2

[CBM*17] CASTORI G., BORRI A., MARIA A. D., CORRADI M., SISTI R.: Seismic vulnerability assessment of a monumental masonry building. *Engineering Structures 136* (2017), 454 – 465. 2

[CSS*13] CAPPELLINI V., SALERI R., STEFANI C., NONY N., DE LUCA L.: a procedural solution to model roman masonry structures. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2 (july 2013), 149–153. 2

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

[FBP17] FITA J., BESUIEVSKY G., PATOW G.: A perspective on procedural modeling based on structural analysis. *Virtual Archaeology Review 8*, 16 (2017), 44–50. 2

[FBP18] FITA J. L., BESUIEVSKY G., PATOW G.: Earthquake Simulation for Ancient Building Destruction. In *Eurographics Workshop on Graphics and Cultural Heritage* (2018), Sablatnig R., Wimmer M., (Eds.), The Eurographics Association. 2, 3

[FFL17] FORTUNATO G., FUNARI M. F., LONETTI P.: Survey and seismic vulnerability assessment of the baptistery of san giovanni in tumba (italy). *Journal of Cultural Heritage 26* (2017), 64 – 78. 2

[Fit61] FITCHEN J.: *The construction of Gothic cathedrals : a study of medieval vault erection /by John Fitchen*. Clarendon Press Oxford, 1961. 2

[GCC18] GASPERIS G. D., CORDISCO A., CUCCHIARA F.: Immersive virtual reality as a resource for unaccessible heritage sites. *IOP Conference Series: Materials Science and Engineering 364* (jun 2018), 012035. 2

[GSN*18] GAUGNE R., SAMAROUDI M., NICOLAS T., BARREAU J.-B., GARNIER L., RODRIGUEZ ECHAVARRIA K., GOURANTON V.: Virtual Reality (VR) interactions with multiple interpretations of archaeological artefacts. In *EG GCH 2018 - 16th EUROGRAPHICS Workshop on Graphics and Cultural Heritage* (Vienna, Austria, Nov. 2018), pp. 1–9. 2

[HCSZ*01] HENRY-CLAUDE M., STEFANON L., ZABALLOS Y., FOURNIER S., REGO A.: *Principles and Elements of Medieval Church Architecture in Western Europe*. Livre ouvert: Architecture. Les éditions Fragile, 2001. 2

[KK12] KRECKLAU L., KOBBELT L.: Smi 2012: Full interactive modeling by procedural high-level primitives. *Comput. Graph. 36*, 5 (Aug. 2012), 376–386. 2

[Lib16] LIBRARY B. P.: Bullet 2.83, 2016. http://bulletphysics.org. 3, 5

[Low07] LOWRIE W.: *Fundamentals of Geophysics*, 2 ed. Cambridge University Press, 2007. 3

[MECB*14] MORTARA M., EVA CATALANO C., BELLOTTI F., FIUCCI G., HOURY-PANCHETTI M., PETRIDIS P.: Learning cultural heritage by serious games. *Journal of Cultural Heritage 15* (05 2014), 318âĂŞ325. 2

[MWA*13] MUSIALSKI P., WONKA P., ALIAGA D. G., WIMMER M., VAN GOOL L., PURGATHOFER W.: A Survey of Urban Reconstruction. *Computer Graphics Forum* (2013). 2

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

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

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

[Raw08] RAWLINSON N.: Seismology lectures, 2008. http://rses.anu.edu.au/~nick/teaching.html. 3

[Rea18] REALITY W. M.: Windows mixed reality, January 2018. https://www.microsoft.com/en-us/windows/windows-mixed-reality. 4

[Rif10] RIFT O.: Oculus rift, January 2010. https://oculus.com. 4

[RL04] RAMOS L. F., LOURENÇO P. B.: Modeling and vulnerability of historical city centers in seismic areas: a case study in lisbon. *Engineering Structures 26*, 9 (2004), 1295 – 1310. 2

[Sid12] SIDEFX: Houdini 16, 2012. http://www.sidefx.com. 5

[SJ13] SALDANA M., JOHANSON C.: Procedural modeling for rapid-prototyping of multiple building phases. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-5/W1* (02 2013), 205–210. 2

[SM15] SCHWARZ M., MÜLLER P.: Advanced procedural modeling of architecture. *ACM Transactions on Graphics 34*, 4 (Proceedings of SIGGRAPH 2015) (Aug. 2015), 107:1–107:12. 2

[SRH*18] SELMANOVIC E., RIZVIC S., HARVEY C., BOSKOVIC D., HULUSIC V., CHAHIN M., SLJIVO S.: VR Video Storytelling for Intangible Cultural Heritage Preservation. In *Eurographics Workshop on Graphics and Cultural Heritage* (2018), Sablatnig R., Wimmer M., (Eds.), The Eurographics Association. 2

[SZAM16] SOUAMI M. A., ZEROUALA M. S., AIT-MEZIANE Y.: The impact of building proportions in the preservation of algiers architectural heritage against the seismic hazards. *Journal of Cultural Heritage 20* (2016), 686 – 693. Cultural HELP 2014 Special Issue. 2

[TP05] THALMANN N., PAPAGIANNAKIS G.: Virtual worlds and augmented reality in cultural heritage applications. In *EUROGRAPHICS Workshop on Graphics and Cultural Heritage* (01 2005), pp. 419–430. 2

[Tru18] TRUSTGXT: Trust gxt virtual reality glasses, January 2018. https://www.trust.com/es/product/21322-gxt-720-virtual-reality-glasses. 5

[Uni05] UNITY: Unity 3d engine, June 2005. https://unity.com. 4

[URB*17] UPHOFF C., RETTENBERGER S., BADER M., MADDEN E. H., ULRICH T., WOLLHERR S., GABRIEL A.-A.: 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* (New York, NY, USA, 2017), SC '17, ACM, pp. 21:1–21:16. 2

[Viv10] VIVE H.: Htc vive, January 2010. https://vive.com. 4

[VR18] VR G.: Google vr, January 2018. https://vr.google.com. 5

[WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 112:1–112:9. 2

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