



HAL
open science

Parametric Modelling Techniques for Rhine Castle Models in Blender

Etienne Sommer, Mathieu Koehl, Pierre Grussenmeyer

► To cite this version:

Etienne Sommer, Mathieu Koehl, Pierre Grussenmeyer. Parametric Modelling Techniques for Rhine Castle Models in Blender. 3D Virtual Reconstruction and Visualization of Complex Architectures, Feb 2024, Siena, Italy. pp.31, <10.3390/heritage8010031>. <hal-04944257>

HAL Id: hal-04944257

<https://hal.science/hal-04944257v1>

Submitted on 13 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Article

Parametric Modelling Techniques for Rhine Castle Models in Blender [†]

Etienne Sommer * , Mathieu Koehl  and Pierre Grussenmeyer 

ICube Laboratory, INSA Strasbourg, CNRS, Université de Strasbourg, UMR 7357, 67084 Strasbourg, France; mathieu.koehl@insa-strasbourg.fr (M.K.); pierre.grussenmeyer@insa-strasbourg.fr (P.G.)

* Correspondence: etienne.sommer@insa-strasbourg.fr

[†] In Proceedings of the 3D Virtual Reconstruction and Visualization of Complex Architectures, Siena, Italy, 21–23 February 2024.

Abstract: Recent advances in 3D modelling have greatly improved the digital reconstruction of historic buildings. Traditional 3D modelling methods, while accurate, are very time-consuming and require a detailed focus on complex architectural features. The use of Building Information Modelling (BIM) technology, adapted to historic buildings as Historic Building Information Modelling (HBIM), has made the modelling process easier. However, HBIM still struggles with a lack of detailed object libraries that truly represent the diverse architectural heritage, due to the unique designs of these ancient structures. This article presents a new method using Blender software, focusing on *Geometry Nodes* and modifier tools for parametric modelling. This method aims to efficiently reconstruct the Rhine region's castles, which are part of Europe's most heavily fortified areas with a history that goes back to the XIth century. Many of these castles, over 500 years old, are now ruins. Our method allows for quick changes and detailed customization to meet the specific needs of archaeologists and heritage researchers. Developed as part of the *Châteaux Rhénans-Burgen am Oberrhein* project, funded by the European Interreg VI programme, this approach focuses on digitizing and promoting the Rhine castles' heritage. The project aims to fill some gaps in parametric modelling by providing a flexible and dynamic toolset for heritage conservation.

Keywords: parametric modelling; Blender; Rhine castles; 3D/4D modelling; medieval heritage; HBIM; dynamic adjustment



Academic Editor: Fabio Remondino

Received: 21 November 2024

Revised: 10 January 2025

Accepted: 11 January 2025

Published: 16 January 2025

Citation: Sommer, E.; Koehl, M.; Grussenmeyer, P. Parametric Modelling Techniques for Rhine Castle Models in Blender. *Heritage* **2025**, *8*, 31. <https://doi.org/10.3390/heritage8010031>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Rhine originates in the Swiss Alps and flows north through Switzerland, forming a natural border with Austria and Liechtenstein before entering Germany. Over a stretch of approximately 180 km, it serves as the boundary between France and Germany. Finally, it flows into the Netherlands, where it splits into several branches before discharging into the North Sea. The Rhine is a significant waterway for Europe, and its banks preserve remnants of the cultural and historical heritage of the nations it traverses. Numerous castles and fortresses along its course reflect the region's rich medieval architectural tradition.

1.1. Historical and Heritage Context of Rhine Castles

These castles are characterised by towers and thick walls, with most occupying strategic positions atop hills. Initially built for defence and control, they later became symbols of power and nobility. Each castle represented the social standing and military strength of its inhabitants. Today, these castles continue to hold significant importance in the architectural

history of the regions along the Rhine. They attract not only historians and archaeologists but also tourists, drawn by the beauty of the buildings and the legends surrounding them. Despite their robust construction, these castles have suffered the effects of time and conflict. While some have been rebuilt or restored, many remain in ruins, contributing to the diversity of the structures still visible today.

Seeing the importance of these castles for the region, the last few decades have seen numerous works aimed at preserving them, but also at understanding them. This task can be very complicated, especially due to the limited number of period documents that remain. However, recent years have seen the emergence of 3D modelling techniques, including Historical Building Information Modelling (*HBIM*) and parametric modelling, which enhance the digitization and preservation of this rich heritage. The Rhine castles, rich in history and architectural varieties, are excellent examples of how to apply these new modelling and visualization methods. These tools make it possible to better study and understand the medieval structures of the region, while also enabling the diffusion of knowledge and representations to the general public through modelling and rendering software.

These studies also have a tourist purpose, enhancing the attractiveness of the region. It explores how parametric 3D modelling can be used to recreate, analyse, and visualise these castles with a high level of detail, contributing to the preservation and understanding of the region's past.

1.2. Challenges in 3D Modelling of Medieval Heritage

Due to its architectural diversity, medieval heritage brings several challenges in the 3D modelling process. These difficulties arise from the complex and varied styles of these ancient castles, as well as their uniqueness. Each castle may feature unique elements, not found on other sites. These elements require the creation of a parametric object library. This is already in place when discussing modern structures with *BIM*, but it is generally not the case when working on heritage with *HBIM* [1]. These challenges include, but are not limited to, the following:

- Unique architectural features: Modern buildings generally follow conventions in shape and size, while medieval structures are unique. Each castle thus has its own architectural details influenced by the era and location of construction, but also by the master builder in charge of the project.
- Complexity in meaning and shape: Rhine castles are complex in their architecture, but also in their meaning. Understanding the architecture of each castle is important for creating 3D models, but also to understand the reasons behind creating specific architectural details or particular defensive or offensive structures. *BIM* libraries related to current buildings are thus not really suitable because they do not take into account the historical and geographical context for object creation.
- Making custom parametric libraries: To successfully manage these architectural complexities, it is necessary to create parametric object libraries specifically for these buildings, taking historical and geographical considerations into account as factors influencing the various structures. These libraries must enable the easy and quick creation of new 3D models, while ensuring that the historical aspects unique to each castle are respected and preserved. This can be even more complicated, as castles often underwent several construction phases, resulting in mixes of materials and styles within the same building, which must still be taken into account and represented.

These elements clearly show that traditional 3D object and *BIM* libraries cannot meet the expectations of this project. It is necessary to propose a new modelling method that takes into account historical discoveries, current acquisition and modelling techniques to allow for the representation of these castles.

1.3. Objectives and Scope of the Study

The goal of this study is to demonstrate the capabilities of the Blender modelling software to create a parametric, detailed, and realistic model of the Birkenfels castle, presented in Figure 1. Starting from a few manually modelled elements, which are used as the initial objects for parametric modelling, this study utilises tools like Blender's *Geometry Nodes* and modifiers. This method makes it possible to create various objects that can be adapted in size, shape and number of duplications. In this way, it is possible to quickly and easily propose multiple possible reconstruction scenarios, based on historical resources and archaeological discoveries. It is also important to note that this method is not designed to create 3D models for accurate stone-by-stone documentation [2], but rather to support visualisation and broader analysis aimed at understanding the historical context and providing a comprehensive interpretation of the site while addressing the archaeological reliability of the reconstructions [3].

The parametric nature of the objects is crucial to propose new reconstructions if new historical data come to light (for example, following new excavation campaigns that have uncovered new findings). The Birkenfels castle is used as a case study, but this method is intended to be applied to other castles. Moreover, this study shows the potential of modelling software in the conservation of medieval heritage, which can be extended to other structures and architectural types. In sum, this study combines traditional 3D modelling methods with new, innovative methods that contribute to the digital reconstruction and preservation of historical monuments.



Figure 1. Aerial view of the current state of the Birkenfels castle [4].

2. Related Work

2.1. Literature Review on HBIM and Parametric Modelling

Parametric modelling and *HBIM* are essential tools for the conservation and modelling of heritage components. These structures were usually created with complex styles and details that vary across periods and locations, requiring an analysis and representation that are semantic, rich in details and sources. This enables not only the representation of the geometry of each element but also highlights and explains the historical aspect associated with it. The creation of object libraries and modelling through geometric primitives has shown its benefits, yet a complete understanding of the geometry and history of the objects or buildings under study is required beforehand [5,6].

To acquire such a level of knowledge, the data capture and processing tools must be suitably adapted. Photogrammetry and terrestrial laser scanning then become part of the acquisition and processing workflow [7,8]. The precision of the data allows for a faithful restitution of the studied object in its actual state, as well as an understanding of the historical context and the most accurate modelling of associated elements from the same period.

Furthermore, one of the advantages of parametric modelling is that it facilitates the creation of complex shapes, often necessary for significant buildings requiring conservation efforts. It is then possible to combine 3D acquisition techniques with parametric modelling [9,10]. This method is particularly effective for using objects that are adaptable over time, to continuously update the digital reconstruction based on possible new discoveries that may occur, thus facilitating conservation and restoration efforts. There are many parametric modelling methods. When selecting an approach, it is essential to consider the desired level of detail for the model as well as the additional functionalities that may be leveraged. Ref. [11] suggest using the *MEL* scripting language to have access to many parameters, while [12] utilise parametric techniques with *NURBS* curves to create complex surfaces from point clouds. A more comprehensive study of parametric modelling methods for heritage conservation is presented by [13].

In addition to the level of knowledge that can be linked to *HBIM* models, these can be integrated into a *GIS* workflow to add semantic data. The understanding of elements is thus enhanced [14,15]. Models enriched with data from various sources are then highly useful in the management and analysis of data related to a site.

Thanks to the combination of these elements, the monitoring and study of sites are improved. The use of 3D models allows simulating the temporal evolution of buildings and thus ensuring their integrity over the years. This is crucial for the safeguarding and security of heritage, regardless of the field. The literature then shows a convergence in ideas and opinions. It is necessary to establish processing chains that allow for an efficient acquisition of the geometry of historical buildings in order to put in place conservation and study methods. This ensures the preservation of the object's geometry, as well as its history, influence and genealogy.

2.2. Existing 3D Modelling Techniques for Heritage

Over the last few decades, the development of 3D acquisition methods had a significant impact on the preservation and modelling of heritage. These methods allow for the collection of precise and comprehensive data of the buildings studied. It is still important that the acquisition methods be adapted to meet the expectations of each project, to avoid overloading with data and processing time when it is not necessary. It can then be beneficial to combine different modes of acquisition and processing on the same site, depending on the importance and the level of detail of each element it consists of [10]. The data gathered in the field must then be sufficient, but not excessive, to convert these raw data

into parametric models. These models must then remain faithful to the historical reality and context [16,17].

Regarding the Rhine castles, several studies using different modelling software have already been carried out, such as Trimble SketchUp [18], Autodesk Maya [19], and Blender [20]. While these studies provide valuable insights, they primarily relied on manual modelling techniques, which required between 2 and 5 months to complete, depending on the prior 3D modelling expertise of the students involved in these projects as part of their master's thesis. The software packages all deliver satisfactory results. They all have points in common, but they also have differences and unique tools [21].

2.3. Gaps and Opportunities in Current Methods

HBIM and parametric modelling have been central research focuses in recent years. They have highlighted the importance as well as the difficulties of heritage modelling due to the variety of styles that exist. The methods have then shown the possibilities of combining geometric data with historical data. The complex shapes, the variety of materials, and the different structures therefore bring specific challenges, characteristic of historic buildings [22]. Automatic segmentation of elements into subgroups could automate and improve the already established modelling processes to more easily extract geometric primitives [15], which is difficult because of the complexity of the architecture of historic buildings [23].

Using these methods would make it possible to set up a more standardised and reusable processing chain. It would be easier to obtain similar results from different institutes and laboratories. In fact, there are always disparities in the acquisition and processing methods depending on the teams, which can lead to significantly different results [24]. The complexity of historical objects or buildings requires acquiring a large amount of data to represent them faithfully. It can then have different objectives: preservation in case of destruction or potential future deterioration [25], visualisation in virtual reality [26–28], or even 3D printing [29]. In all cases, the 3D models must be edited and adapted to have geometry detailed enough to meet the study's needs, while remaining lightweight to be manipulated and visualised correctly [30–32].

On the one hand, it is clear that the literature highlights many difficulties in the field of *HBIM* and parametric modelling, but also proposes several answers. On the other hand, 3D modelling software such as Blender or Maya, which are not initially intended to be used in *HBIM*, are gaining in popularity in the heritage conservation sector. These software packages are powerful, easy to use, and contribute to the conservation and visualisation of heritage, whether for scientific, tourist, or educational purposes. In any case, many gaps in terms of heritage modelling have not yet been filled. It is therefore necessary to try to obtain the best points from each method to arrive at conclusive results in the most efficient way possible.

3. History of the Birkenfels Castle

The history of the Birkenfels castle is difficult to recount due to divergent sources. Although its timeline and master builder are known, the reasons for its incomplete state remain uncertain. Bernhard Metz, a medievalist and historian who has carried out several studies on the history of Rhine castles, and the *Association for the Preservation of Medieval Architecture* (*Association de Sauvegarde de l'Architecture Médiévale*) conducted excavations in the 1970s under the access tower, providing insightful reports on material traces. No excavations have occurred since the 1980s.

Historians agree that the first documented reference dates back to 1289 [33], when Emperor Rodolphe de Habsbourg granted “Bergfels in the ban of Oberehnheim” to Burck-

hard Beger, compensating the city of Obernai for unjustly taken lands [34]. This indicates the castle was initially illegitimate, built on lands belonging to Obernai and the imperial crown. Burckhard Beger, supported by the bishop of Strasbourg, held a ministerial position starting with Henri de Stahleck and continuing with Walter de Geroldseck in 1260. After the bishop's defeat at the Battle of Hausbergen in 1262, Strasbourg retaliated by destroying bastions loyal to the bishop, possibly including Bergfels castle. Its abandon in the XIIIth century is agreed upon, but its construction period is still debated.

The first theory suggests that the construction began in 1246 under bishop Henri de Stahleck, who demolished the Hohenstaufen castle (near the city of Obernai) and rebuilt city fortifications [33]. This supports Rodolphe de Habsbourg's 1289 action to regularize the situation, considering Birkenfels castle illegal. The second theory is that construction began in 1260 when Walter de Geroldseck became bishop of Strasbourg. This was a time of rising tensions and he asserted control over imperial territories. Bernhard Metz suggests the Beger, after losing their urban residence, wanted to establish themselves outside the city [35]. The castle's incompleteness by the XIIIth century supports this, as the Beger likely lacked time to finish from 1260 to 1262.

Two main elements suggest the castle was incomplete in the XIIIth century. Metz's 1970s excavations found no human occupation traces before the XVth century [36]. Additionally, some courtyard stones, prepared for use, showed no lifting holes [37] or mortar [38], unlike those in the dungeon. This indicates construction halted after two years due to 1262 events and partial fire destruction.

The castle's ruin timeline is unclear. While XVth century activity suggests the Beger received the castle as an imperial fief [34], interpretations vary. The term "die zarge Birkenveltz" could mean "ruined" or "abandoned", not final abandonment. Metz's research showed the access tower dates to the early XVth century, possibly indicating the Beger's peak [39]. However, the 1441 castellany peace, not mentioning Birkenfels, suggests a decline. The Beger likely never resided there in the XIIIth century. The castle was sold and abandoned after the Beger line died in 1532, when Emperor Charles V sold it to Vice-Chancellor Mathias Held. Archaeological data confirm gradual abandonment starting in the early XVIth century, coinciding with 1532 events.

Thus, Birkenfels castle's history reflects the ministerial family's noble aspirations, halted by their protector's downfall, leading to the abandonment of their partially destroyed castle.

4. Introducing Blender

Blender is a powerful 3D modelling software capable of creating complete scenes thanks to its integrated toolset. It is a free and open-source software, boasting a very large user community, ranging from individuals using it as a hobby to professional animators, game developers, and visual effects artists. The software comes equipped with a comprehensive suite of modelling, rigging, animation, simulation, rendering and compositing creation tools. Its open-source nature makes it accessible to everyone, allowing users to develop and distribute their own plugins. The community plays a significant role in the continuous development of the software, by reporting or fixing bugs, creating online tutorials, sharing modelling methods, or even directly sharing modelled objects. The decision to use Blender for this study is based on its all-in-one, free aspect, compatibility with a wide majority of 3D object and mesh formats (*.obj*, *.fbx*, etc.), and its use by a vast international community. Its parametric modelling tools and rendering options offer methods well suited to meet the expectations of this project.

4.1. Description of Geometry Nodes

Geometry Nodes [40] are a relatively recent system in Blender, introduced in version 2.92, which was released in February 2021 (for context, the latest stable version is 4.2 at the time of writing in November 2024). *Geometry Nodes* are based on a parametric and procedural system to create various geometries through a network of nodes that are interconnected. These nodes can be of various types: geometry, integer or float number, vector, string, etc., to name the most common ones. Figure 2 shows how a *Geometry Nodes* setup looks in the Blender environment.

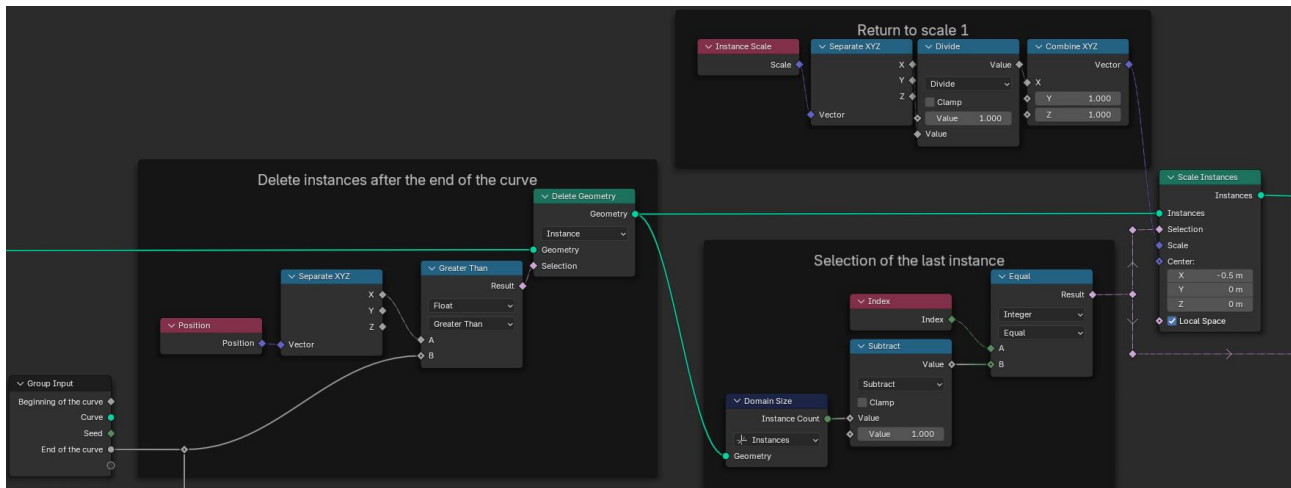


Figure 2. Example of a *Geometry Nodes* setup to delete some instances and scale others.

As for the geometries, the most frequent actions include deformation, duplication, translation, rotation, and scaling. By correctly connecting the nodes, it is possible to create complex and varied geometries from a simple base object. Thanks to the interconnection of nodes, modifying a parameter will automatically affect the entire structure, allowing for the quick creation of variations of the same object. Moreover, a specific *Geometry Nodes* setup can easily be created in one project and imported into another, significantly simplifying data management.

4.2. Description of Modifiers

In addition to *Geometry Nodes*, parametric modelling can also be achieved through the integrated modifiers in Blender [41]. These are non-destructive operations that can be applied to objects to alter their geometry in ways that would not be feasible manually. These tools enable the creation of complex shapes with precision while ensuring a correct topology. These features are therefore perfectly suited for modelling complex architectures in the context of heritage conservation.

5. Parametric Modelling Workflow

This paper aims to develop an efficient and reusable methodology for creating complex medieval architectural elements from a limited set of manually modelled objects. This approach requires only a few manual operations, resulting in significant time savings and reduced dependence on advanced modelling expertise. Particular emphasis is placed on the parametric modelling workflow, which allows castle models to be easily updated and adjusted as archaeological analyses evolve. This adaptability is especially important in collaboration with archaeologists, where new interpretations may necessitate the addition, modification, or removal of elements to ensure the models remain accurate and reflective of expert opinions. Additionally, the workflow is designed to be reusable across multiple

projects, enabling the efficient modelling of features unique to individual castles, while considering the shared architectural characteristics often dictated by similar construction periods. Some structures characteristic of Rhine castles will be presented, demonstrating how collaboration and multiple iterations have produced the most plausible and historically consistent models.

5.1. Creation of a Random Stone Wall

A significant feature of some Rhine castles is the appearance of the stones used to build the walls. Made from Vosges sandstone, these stones are referred to as “bossage stones” because the center of the stone is left uncut. This method offers several advantages:

- Time-saving, as there is no need to work on the central part of the stone.
- A demonstration of strength and power to opponents who would like to attack the castle.

These stones were typically used for walls that were most vulnerable to attack. Birkenfels castle has a wall made of this type of bossage stone, facing south, which was most likely to encounter enemies or projectiles due to the natural terrain. These stones are generally drilled with a hole, known as “lifting hole” (*trou de louve* in French), allowing for the use of a lifting clamp to raise the stone to the top of the wall during construction.

The stones making up the wall do not have a predefined size. In width, they generally range from 30 to 80 cm. As for the height, it is crucial to ensure that each stone in a row has the same height so that each stone level can be laid flat. One level may not necessarily have the same height as the next one.

A first method involves using a single manually modelled cube with sides measuring 1 m. A “Bevel” modifier is applied to slightly smooth the edges (which cannot be perfectly straight due to wear and tear), and a “Boolean” modifier is used to create the lifting hole. Next, a random value (between 30 and 80 cm) is assigned to each stone to determine its width. An “Accumulate Field” node is then used to sum each random value, placing the stones consecutively. Once a sufficient number of stones have been created, a second random value (between 30 and 60 cm) is assigned to the row to set its height. Similarly, another “Accumulate Field” node is used to stack the rows on top of each other.

By repeating this operation multiple times, a straight wall is obtained, as shown in Figure 3. Using a specific setup of *Geometry Nodes*, this wall can be deformed to follow a curve representing the castle’s ground plan.

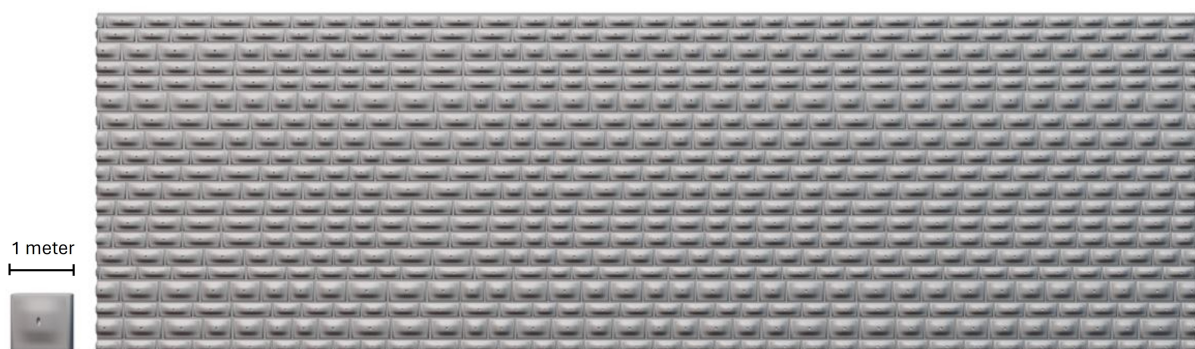


Figure 3. Input stone and automatic stone wall created using a specific *Geometry Nodes* setup.

In Blender, each random value node has a seed value. This seed controls the distribution of random values for a given parameter. Figure 3 illustrates one possible version of the wall, and changing a single seed value would automatically generate a completely different version from the first. By adjusting this seed value, it is possible to create an infinite number of variations, all using the same initial stone and *Geometry Nodes* setup.

This method still has several drawbacks. Since the geometry is deformed along the entire curve, it is, for example, impossible to place specific stones at the castle's corners, as would be done in reality. Additionally, because each stone is first positioned along a single wall section and then deformed along the curve, individual access to each stone is lost, making it impossible to create openings, such as arrow slits. The most significant disadvantage of this approach is that it does not allow for the use of Blender's instancing system.

Instancing allows for duplicating objects while maintaining minimal memory usage, as each instance is merely a reference to the original object, rather than multiple unique objects. Regardless of how many instances are created, the face count of the scene remains constant. Typically, in Blender, one begins by creating a "master" object to be duplicated. Then, *Geometry Nodes* are used to handle the duplication according to the desired parameters and methods. This process not only saves time by avoiding manual placement of each element but also ensures realistic and random variation in the distribution of objects. Instances can be individually manipulated in terms of position, rotation, and scale, allowing for precise adjustments. Furthermore, Blender allows random variations to be applied to these parameters, resulting in a more natural and less uniform outcome.

In this case, instancing cannot be applied because deforming along a curve requires modifying the initial object's geometry.

To address these limitations, a second method was developed, showcasing the ongoing evolution and improvements in Blender. The previous method was designed before Blender version 4.0 (released in November 2023), which introduced the *Repeat Zone* within *Geometry Nodes*. The *Repeat Zone* functions like a for-loop in programming: performing a series of operations for a set number of iterations. Moreover, multiple *Repeat Zones* can be nested within others. This second method utilizes this feature by separating each segment of the curve that outlines the castle's ground plan. It operates as follows:

- **Inputs:** A curve representing the castle's ground plan and a cube representing the stones.
- **Iteration on each curve segment, representing a wall section:** A minimum and maximum stone width value is set. Since the *While* condition does not exist in *Geometry Nodes*, a condition must be found to generate enough stones along the segment. The segment length is divided by the minimum value to ensure that there are enough stones along the segment. At this stage, a row of stones is created, whose total length exceeds the segment length. This row is positioned at the center and created along the X-axis. Each stone's position can then be compared to the segment, allowing for the removal of those that exceed the segment's length. As this involves instance deletion rather than a Boolean geometry operation, it is executed almost instantly (under 0.1 ms according to Blender's built-in timer). The instancing system significantly reduces the computational complexity of the model, resulting in a structure with only 63,520 faces (corresponding to the quoin stones, located at the corners and deformed, which cannot be instantiated), compared to 1,067,040 faces when the instancing system is not utilised. These values are based on Figure 4, which contains a total of 1847 instances, further demonstrating the efficiency and scalability of the workflow. To leave space for quoin stones, 20 or 40 cm is subtracted from the curve length. This operation will be explained further in this paper. Among the remaining stones, the last one is selected, and its scale is adjusted to match the exact required length to complete the curve.
- **Creating the wall section:** Once a row of stones is obtained, a random value is assigned to determine the height of the row. Using the *Repeat Zone*, this operation can be repeated as many times as needed to achieve the desired wall height.
- **Orientation to match the segment:** At this point, the wall section is positioned at the center, oriented along the X-axis, with a total length that matches the length of the segment. This wall section is then translated to the first vertex of the pre-

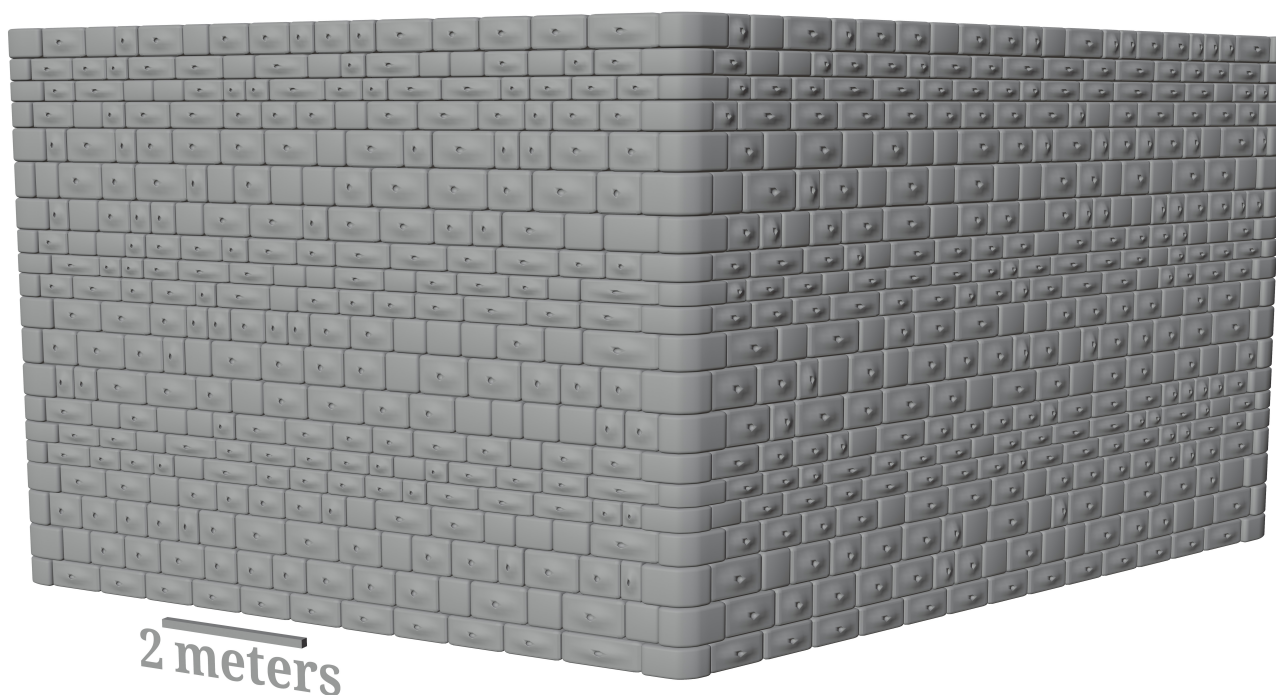


Figure 5. Automatic stone wall using the instance system.

5.2. Integration of Arrow Slits

Arrow slits are narrow openings made in the walls of medieval castles that allow defenders to shoot arrows or crossbow bolts while remaining protected from enemy attacks. Their narrow exterior and wider interior design provides defenders with a wide shooting angle, while minimizing the chances for attackers to strike those inside. This was a crucial feature of medieval defence, enabling defenders to repel attacks while staying shielded.

In this context, focused on the exterior of castles, these arrow slits are represented as thin slits. Within the established parametric modelling system, they are directly generated during the creation of the wall sections, as previously described. When creating the rows of stones, to be able to place an arrow slit, the segment in question is divided into several parts. For example, to place two arrow slits, the segment must be divided into three sections, as proposed in Figure 4. Each segment is then processed individually, leaving a gap at the beginning and end to match the width of the arrow slit. Additionally, a distinction is made to apply an offset of 20 or 40 cm for the quoin stones, but only at the two vertices of the initial segment. The position and size of each arrow slit can then be adjusted by adding a Boolean condition based on the index of the stone row being processed and the index of the segment of the original curve.

5.3. Parameterising the Parapet Walkway with Adaptive Sizing of Merlons and Crenels

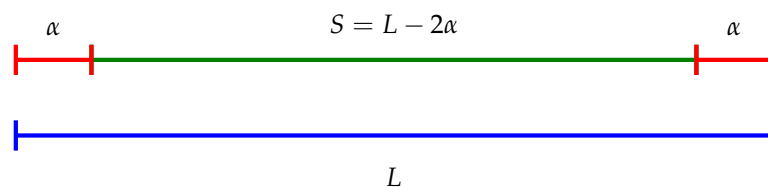
The previous two sections propose methods for creating castle walls and integrating elements by directly influencing the position of the stones. These defensive structures are generally accompanied by a parapet walkway, located at the top of the castles, allowing guards to move around and have an unobstructed view of the surroundings. Moreover, this elevated position provided a significant advantage over potential attackers positioned below. Since this parapet walkway is situated at the top of the walls, it seemed logical to reuse the same ground footprint curve as the castle itself for the parameterisation. Typically, a parapet walkway consists of a floor on which guards can walk and a stone parapet approximately 1 m high, featuring an alternation of the following two elements:

- **Merlons:** Stone structures providing full protection.

- **Crenels:** Empty spaces between merlons, allowing for an unobstructed view downwards while protecting the lower body.

For Rhine castles, crenels were generally accompanied by small wooden shields, known as *huchette* in French, which could rotate to protect against arched arrows shot from below that might fall on soldiers from above their position.

To enable continuous and precise adjustment of the individual sizes of the merlons and crenels while maintaining an integer number of each element, a specific adjustment method was implemented. This method ensures that, regardless of the actual value assigned to the size of each merlon and crenel, these dimensions are automatically adjusted so that the entire curve segment under consideration can accommodate an integer number of these elements. Thus, the total length of the segment is considered to adapt the dimensions of the merlons and crenels, ensuring a coherent distribution and that each element fits perfectly without segmenting the curve. This approach preserves visual harmony while providing the necessary flexibility for customizing the structure. The method implemented is as follows:



- L = Total length of the curve between two control points;
- α = Length of half a quoin structure (the other half is on the neighbouring segment);
- $S = L - 2\alpha$ = Length of the segment on which the merlons and crenels will be placed.

The following values are also defined:

- M = Width of a merlon;
- C = Width of a crenel;
- n_m = Number of merlons;
- n_C = Number of crenels.

Thus:

$$\begin{aligned}
 S &= L - 2\alpha = M \cdot n_m + C \cdot n_C \\
 &= M \cdot n_m + C(n_m + 1) \\
 &= (M + C)n_m + C \\
 \Rightarrow L - 2\alpha - C &= (M + C)n_m \\
 \Rightarrow \frac{L - 2\alpha - C}{M + C} &= n_m
 \end{aligned}$$

Since n_m must be an integer:

$$\frac{L - 2\alpha - C}{M + C} = n_m \in \mathbb{N}$$

L , 2α , and C are fixed, so only M can vary to satisfy the condition $n_m \in \mathbb{N}$.

Let T be the truncated value of n_m . Thus, T represents the integer value corresponding to the number of merlons.

The aim is to obtain:

$$\begin{aligned} \frac{L - 2\alpha - C}{(M + \Delta_m) + C} &= T \in \mathbb{N} \text{ with } \Delta_m \in \mathbb{R} \\ \Rightarrow L - 2\alpha - C &= T(M + \Delta_m) + T.C \\ &= T.M + T.\Delta_m + T.C \\ \Rightarrow L - 2\alpha - C - T.M - T.C &= T.\Delta_m \\ \Rightarrow \Delta_m &= \frac{L - 2\alpha - C}{T} - M - C \end{aligned} \quad (2)$$

This value Δ_m , given by (2), is then added to the input value M to ensure $n_m \in \mathbb{N}$.

It is important to note that the above method of placing merlons and crenels is only possible if the segment length is sufficient, which means if condition (3) is respected:

$$L > 2C + M + 2\alpha \quad (3)$$

If this condition is not met, a single crenel with one *huchette* is placed between the two quoins stones, with a length equal to $L - 2\alpha$.

This method, illustrated with Figure 6, offers the advantage of allowing dynamic adjustment of the structure while remaining consistent with the rest of the castle, as the same ground footprint curve is utilised. The figure highlights that the two structures are different, adhere to the specifications outlined above, and were obtained by merely modifying numerical values. The merlons and crenels are wider in Figure 6b (resulting in a lower number of them), and the orientation of the *huchettes* can also be adjusted (according to the terrain surrounding the castle, ensuring the model's compatibility with all possible configurations). Furthermore, as in the previous cases, the instancing system was employed, ensuring optimal management of computational resources.

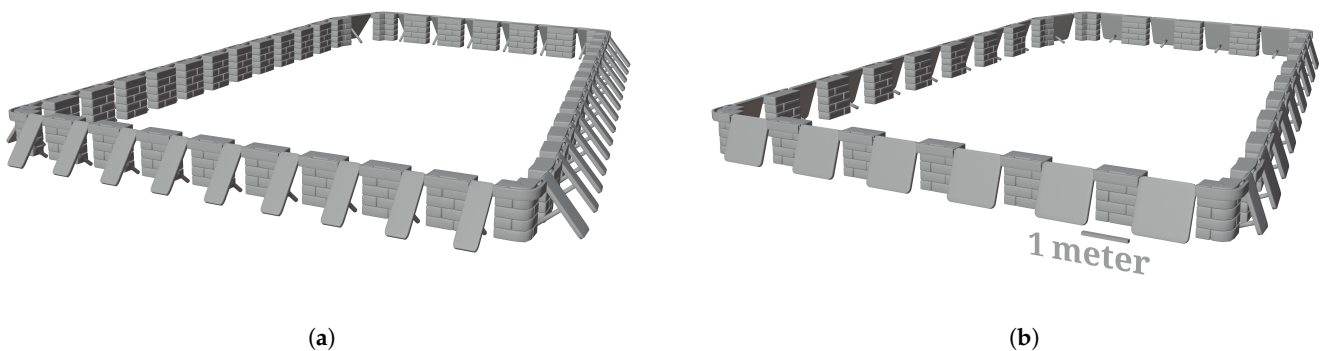


Figure 6. Adjustment of the size of merlons and crenels. (a) Small crenels. (b) Big crenels.

5.4. Adding a Roof to the Castle Structure

Once the elements of the previous sections have been added, the overall structure of the castle can be considered complete. The structure remains open, as the elements were created along the edges of a curve. The logical next step is to add a roof, thereby achieving a complete global structure. Following the same approach previously presented, the goal is to create a parametric model that can be adapted quickly for use in a wide range of configurations. To achieve this, it starts with a polygon composed as follows:

- A first set of vertices at an altitude Z_1 .
- A second set of vertices at an altitude Z_2 , with $Z_2 > Z_1$, all positioned within the surface formed by the first set of points when projected onto the same plane.

By connecting the vertices of these two sets, as shown in Figure 7, a 3D object can be created where each face is an inclined plane representing a roof slope.

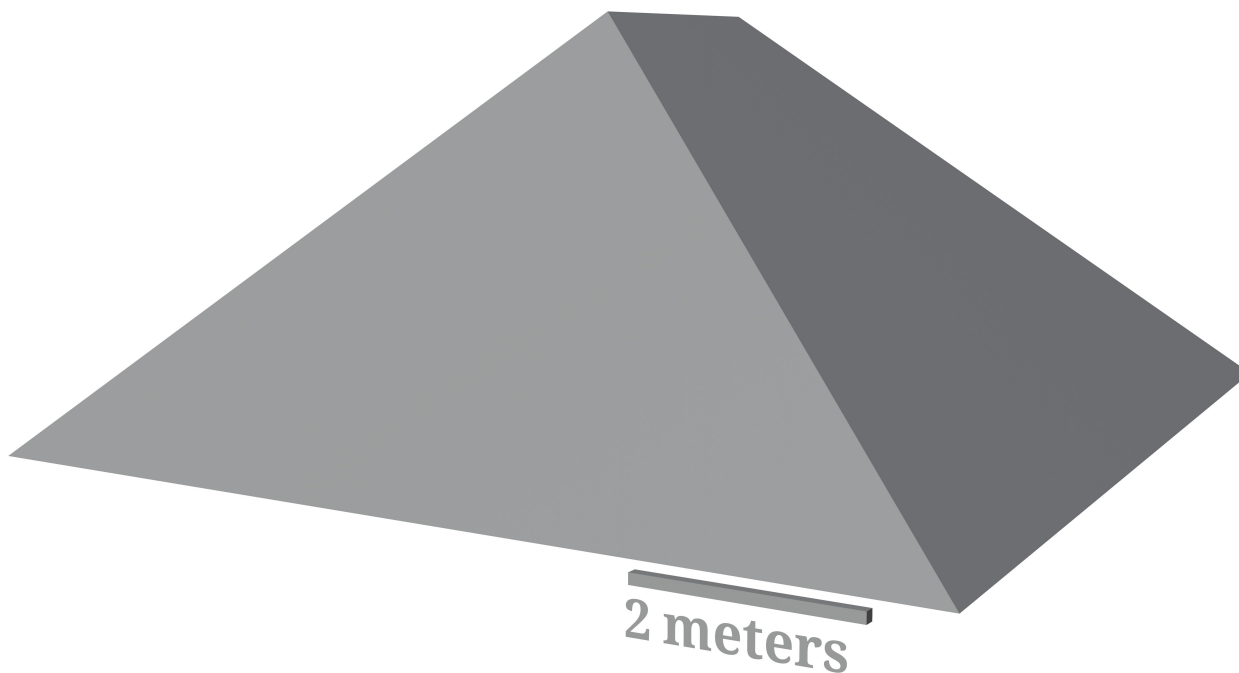


Figure 7. Input polygon giving the roof shape.

This object is then used as an input for the *Geometry Nodes* to create the roof. The process is divided into two key steps; the first is to place tiles on the roof slopes, and the second is to place ridge tiles.

For the roof slopes, following discussions with our archaeological partner, the medieval tiles look like those found on the towers of the covered bridges in Strasbourg. These tiles are regular, placed side by side, with a regular alternation between two successive rows. This structure is relatively easy to create using adapted *Geometry Nodes*, but the challenge lies in adapting it to each roof slope to ensure the correct orientation and dimensions.

To iterate over each roof slope, a *Repeat Zone* is used, allowing each slope to be processed separately. Each slope can have two configurations; it can either be triangular or trapezoidal. In both cases, the two lower vertices (referred to as **A** and **B** hereafter) form a segment on which the orthogonal projections of the upper vertex are located, either as a single point **C** for a triangle or two vertices for a trapezoid. The idea is then to analyse the surface to correctly place the roof tiles. For simplicity, the method is discussed in the context of a triangular face, but the same approach can be applied to a trapezoidal face by considering one of its vertices with altitude Z_2 .

First, the tiles must be placed at the beginning of the roof slope. Because of the three-dimensional context, the notion of “beginning” can be interpreted in various ways. It is essential to define it precisely. Here, the “beginning” of the roof slope is defined as the vertex with the smallest local X coordinate in a local orthonormal coordinate system composed as follows:

- Local X -axis: The axis formed by the two vertices **A** and **B**.

- Local Y-axis: The axis formed by the segment between the third point of the triangle and its orthogonal projection on segment AB.
- Local Z-axis: Formed by the normal vector to the face oriented positively with respect to the global Z-axis, meaning the dot product between these two axes is positive.

To determine these axes, vectors \vec{AB} and \vec{AC} are firstly calculated with relation (4):

$$\vec{AB} = \begin{pmatrix} x_B - x_A \\ y_B - y_A \\ z_B - z_A \end{pmatrix}, \quad \vec{AC} = \begin{pmatrix} x_C - x_A \\ y_C - y_A \\ z_C - z_A \end{pmatrix} \tag{4}$$

The normal vector \vec{n} to the triangle’s plane is obtained via $\vec{n} = \vec{AB} \wedge \vec{AC}$.

Next, the normal vector is normalized. To ensure that the local Z-axis is correctly oriented relative to the global Z-axis, \vec{n}_{norm} is multiplied by the sign of its dot product with the global Z-axis. This ensures that the face’s normal vector is directed outward.

The height of the triangle is calculated by projecting point C onto line AB using $\vec{AP} = \frac{\vec{AC} \cdot \vec{AB}}{\|\vec{AB}\|^2} \vec{AB}$. Then, the projected point P is given by (5):

$$\mathbf{P} = \mathbf{A} + \vec{AP} = \begin{pmatrix} x_A \\ y_A \\ z_A \end{pmatrix} + \frac{\vec{AC} \cdot \vec{AB}}{\|\vec{AB}\|^2} \begin{pmatrix} x_B - x_A \\ y_B - y_A \\ z_B - z_A \end{pmatrix} \tag{5}$$

The triangle’s height h , calculated with (6), is the distance between C and P:

$$h = \|\vec{CP}\| = \sqrt{(x_C - x_P)^2 + (y_C - y_P)^2 + (z_C - z_P)^2} \tag{6}$$

A local orthonormal coordinate system is then constructed, with the origin at point P, the Y-axis aligned with \vec{PC} , the Z-axis adjusted as described, and the X-axis obtained via $\vec{x} = \vec{y} \wedge \vec{z}$. Local coordinates for points A and B are calculated using dot products with the local X-axis. Consequently, the point identified as “leftmost” is the one with the smallest x_{local} coordinate. This method can then be visualised graphically using two cases summarised in Tables 1 and 2, represented by Figure 8:

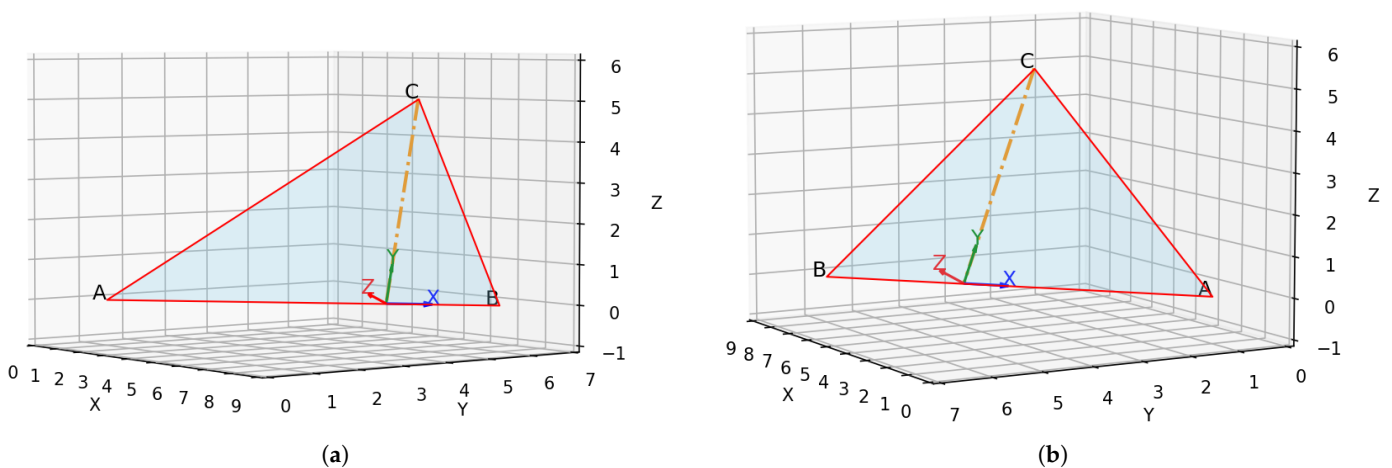


Figure 8. Triangles illustrating the 2 possible configurations. (a) Leftmost point: A. (b) Leftmost point: B.

Table 1. Summary of triangle 1.

Property	Value
Coordinates A	[1, 1, 0]
Coordinates B	[8, 6, 0]
Coordinates C	[5, 6, 5]
Normal vector	[0.55, -0.77, 0.33]
A in local frame	[-6.16, 0, 0]
B in local frame	[2.44, 0, 0]
Leftmost Point	A

Table 2. Summary of triangle 2.

Property	Value
Coordinates A	[1, 1, 0]
Coordinates B	[8, 6, 0]
Coordinates C	[7, 2, 5]
Normal vector	[-0.51, 0.72, 0.47]
A in local frame	[5.46, 0, 0]
B in local frame	[-3.14, 0, 0]
Leftmost Point	B

Using this method, the roof tiles are placed at the determined “left” vertex and oriented towards the second vertex following the same approach described in Section 5.1. A length equal to the distance AB and a height equal to the triangle’s previously calculated height are assigned to the global tiling. Each tile is then rotated around this segment to ensure its normal aligns with the roof slope’s normal.

A raycasting technique is used to filter out unnecessary tiles; a ray is cast upward in the +Z direction from each tile instance. If the ray intersects with the roof surface, the tile instance is retained; otherwise, it is removed. This method is more efficient than performing geometric intersection operations.

Finally, ridge tiles are placed along the roof’s edges using instances along the polygon’s edges. The geometry is converted into curve segments and sampled regularly, capturing attributes such as tangents and normals to ensure correct alignment. Figure 9 shows the effect of raycasting on the structure, allowing only the tiles corresponding to the size of the input geometry to be retained. It also shows the final result of the roof.

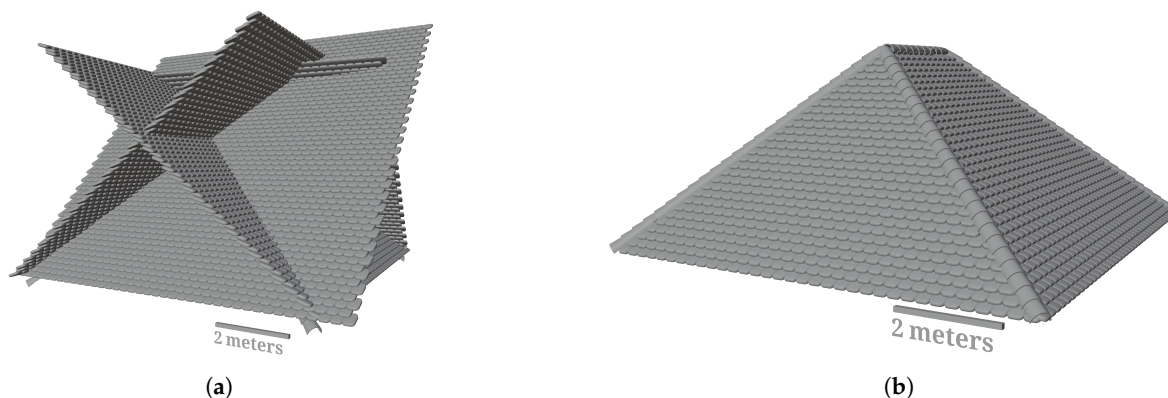


Figure 9. Tiles before raycasting and final roof model. (a) Tiling before raycasting. (b) Instance deletion by raycasting.

5.5. Creation of a Dynamic Hoarding Structure

Hoarding structures are defensive wooden constructs used starting from the XIIth century. They are typically placed on the walls of castles and have two main functionalities:

- Shooting arrows while being protected from enemy arrows by a roof.
- Dropping stones or boiling oil on assailants located below.

Depending on the region and country, numerous types of hoardings have emerged. A very well-known and visited model of hoarding can be found in Carcassonne, a fortified medieval city in the south of France. Figure 10 presents this model.



Figure 10. Example of hoarding structures in Carcassonne, France [42].

It is important to note that even if the period is similar, it does not mean that the types of hoardings are the same in the south of France as for the Rhine castles. At this point, discussion with specialists in Rhine medieval architecture becomes interesting and necessary. The parametric modelling of hoardings must align with what was actually done in the region. The creation process of these models is as follows:

1. Manual modelling of a minimum number of objects that will serve as inputs.
2. Parameterisation of several models, based on basic knowledge and the few elements modelled previously.
3. Presentation of the models to the specialized archaeologist for confirmation and modification.
4. Adjustment of the models according to the archaeologist's advice.

This workflow ensures efficient work with the specialist. Afterwards, the model is parameterized to be adaptable in number and dimension, while allowing it to be deformed along a curve that represents the footprint of the studied castle. Figure 11 presents the structure before and after modification, with major changes including:

1. Offset of the arrow slits: These must be located between two planks and not in the middle of one.
2. The planks that serve as the floor should not be visible; they should be concealed by the vertical planks.
3. The support beam for the roof above the planks should not be visible; it must be hidden by the vertical planks.

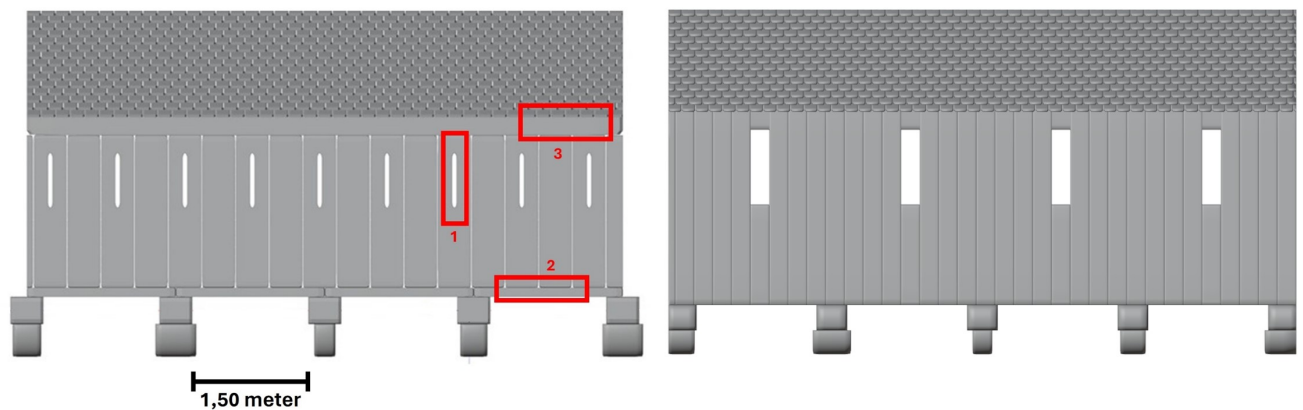


Figure 11. Orthographic front view of hoardings before and after consultation with the specialist.

Other minor changes, such as the size and overlap of the tiles with each other, as well as dimensions, have been corrected. The structure inside (thus not visible from the outside) has also been corrected.

6. Procedural Texturing

The models created through this workflow are intended to be used in tourist mediation and education. They are partly intended for a non-professional audience, and the visual aspect is therefore of great importance. The previously presented models (Figures 3–6, 9 and 11) are only geometries whose visualisation is intended to aid modelling. The applied grey colour does not match the colours found in reality. This is where texturing is important; it involves applying colours and light effects to the faces of a mesh in order to produce a colour render, enhancing the sense of realism and immersion.

Generally, there are two types of textures: photorealistic textures, derived from photographs of real elements, and textures entirely generated by algorithms using materials and physical light properties. The focus here is on the second method, also known as procedural texturing. It has several advantages:

- They are fully parametric and dynamic, meaning they can be adjusted at any time.
- They do not require the UV projection of the object. Thus, the duplication or deformation of a parametric structure is not a problem when applying textures.
- Unlike textures derived from photographs, these are not repeating images. This ensures diversity in the representation of an object, thereby improving the effect of realism.
- There is no resolution limit in the quality of the texture; whether one is close to an object or viewing it from a distance, the rendering will always be of high quality.

6.1. Procedural Textures Applied on a Stone Wall

A specific procedural texture, illustrated with Figure 12, has been developed to be applied to the stone wall created using *Geometry Nodes*. It combines noise textures and Voronoi textures to simulate the intrinsic differences in the type of stone used in reality. To this, parameters for ambient occlusion are added, simulating, for example, an impact on the stone, and some white spots caused by erosion. Finally, a combination of normal maps is used to simulate the micro-details that make up the stones. These elements result in a complex texture, which adds several visual details to the object while ensuring a geometry that remains very light and thus manageable.

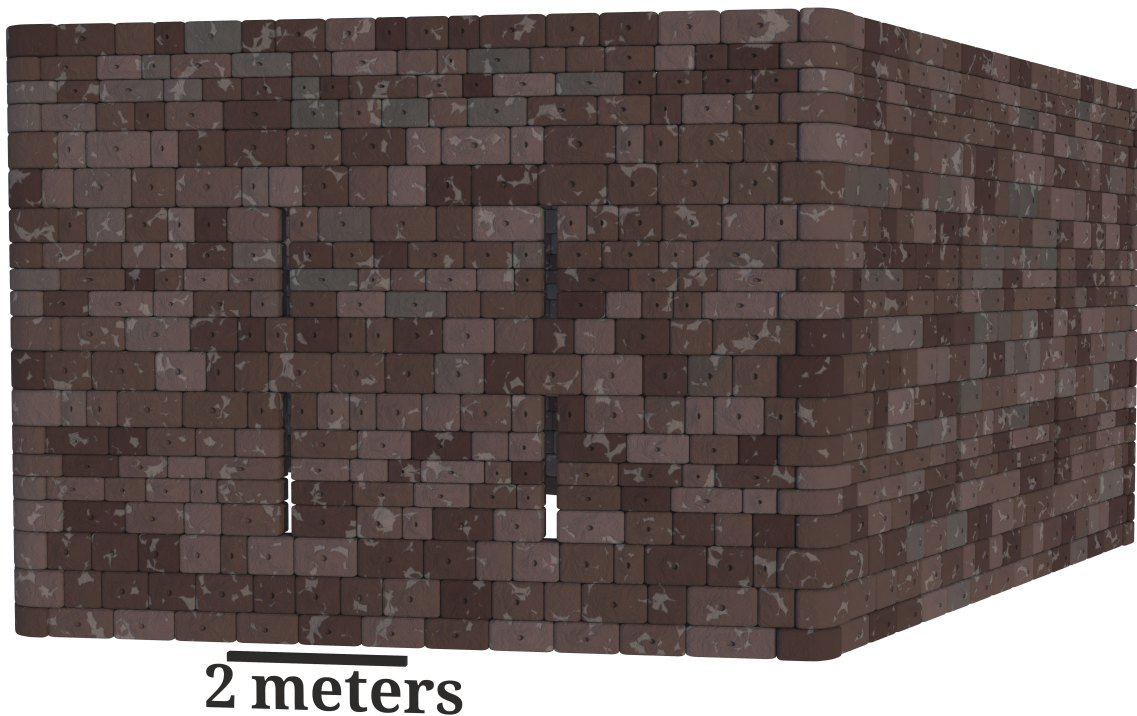


Figure 12. Application of procedural textures on a random stone wall.

6.2. Procedural Wood Textures

Creating a procedural texture that simulates wood is much more complex than creating the stone texture mentioned in the previous paragraph. Indeed, the wood characterising a plank includes many characteristic elements that need to be recreated:

- **Knots:** These are circular or oval marks left by the branches that were attached to the main trunk of the tree.
- **Fibers:** They are aligned in the direction of the tree's growth and define the wood grain.
- **Wood grain:** This is the pattern of wood fibers that naturally forms as the tree grows. The grain can be straight, wavy, or swirling.
- **Growth rings:** These indicate the age of the tree and the annual growth conditions.
- **Colour:** It can vary from very light to very dark, and can also change over time and under the effect of sunlight.
- **Texture:** The texture of the wood can be smooth or rough, depending on the type of wood used.
- **Imperfections:** Knots, cracks, resin inclusions, or other natural imperfections add uniqueness to each piece of wood.
- **Shine:** Some types of wood have a shinier or more matte appearance.

To properly parameterise the texture, each of these elements must be individually adjustable, while ensuring that the others are adjusted accordingly. The whole must be interconnected to ensure a texture applicable to all encountered geometries and to all forms of duplication or deformation. Generally, the textures used are:

- **Wave texture:** Creates bands with distortion, here used to give nuances in the texture.
- **Noise texture with Fractal Brownian Motion:** Allows combining multiple levels of noise to create details, here combined with the wave texture.
- **Voronoi texture:** It is the distance to the closest feature point as well as its position and colour. Here, it is combined with a wave texture to create the knots in the wood.

- **Musgrave texture:** It generates so-called organic textures, but they are very variable and can adapt to many uses. Here, this texture is used to control the roughness of the material.
- **Noise texture:** This is a completely random noise, different from Perlin Noise [43]. It is used to add randomness to the various textures mentioned above.

These textures are then combined to produce the desired result. They are associated with a colour ramp, which gives the colour nuances that appear in the wood, but they are also linked to a normal map and a roughness map to parameterise how the material reacts to light sources. Figure 13 shows how the texture can be modified by changing just one parameter related to the wood knots and the colours of the colour ramp. The camera position for rendering and the light settings are exactly the same. These changes can be made in a few seconds, and allow for a completely different result.

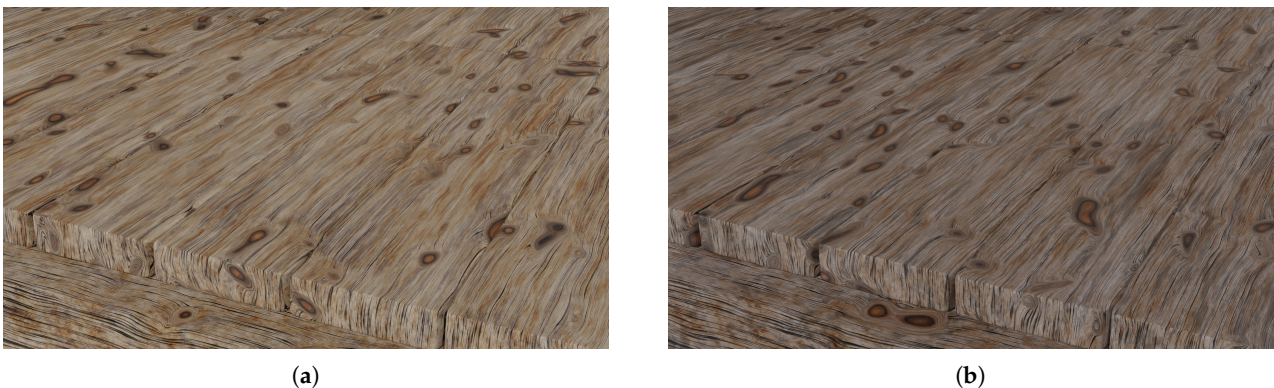


Figure 13. Comparison of two different settings of the same wood texture. (a) Light wood texture. (b) Dark wood texture with variations in the knots.

This method thus allows for extensive reuse of this texture, even within the same project, while ensuring that there will be no repetitiveness in the scene. This texture can then be applied to the hoardings, as shown in Figure 11, to achieve a realistic result of the structure, as depicted in Figure 14. This approach allows for easy and rapid adaptation to various reconstruction scenarios.



Figure 14. Textured hoarding structure.

7. Integration of the Model into Its Natural Environment

Once the castle modelling is complete, it is necessary to integrate the model into its actual environment. A Digital Terrain Model (DTM) could be created using *LiDAR* mounted on a drone, but this method is very costly, requires time on-site, and can also be restrictive due to flight permissions depending on the castle's location. (In France, some castles are close to a military airbase, making flight permissions difficult to obtain). Instead, it was chosen to use High-Density *LiDAR* data made available for free by the *National Institute of Geographic and Forest Information (IGN)* [44]. Measurements are still ongoing across the French territory, but Alsace is a region that has already been mapped.

Point clouds can then be retrieved for free in the form of tiles with a ground coverage of 1 square kilometre. *CloudCompare* [45], a free and open-source software, has a plugin using the *CSF* algorithm [46] that allows for the extraction of ground points. However, for Alsace, the cloud classification has already been performed by *IGN*. The classified ground points can therefore be directly retrieved.

The ground points are then separated into two different clouds: the environment near the castle, and the distant environment. It is then possible to resample the two clouds differently, depending on the details necessary for visualisation. The nearby cloud has been resampled to keep a point every 50 cm, while the distant cloud has been resampled to have a point every 2 m. Subsequently, a mesh was created from the two clouds using *CloudCompare* through the plugin utilising the *Poisson* algorithm [47]. Each mesh is then translated to be brought close to the origin of the coordinate system and ensure smaller coordinates for better management of graphical resources. The meshes thus allow for a good modelling of the terrain, but to integrate them into Blender and ensure good optimization, the mesh must be regular, closed, and topologically correct. For this purpose, the open-source algorithm *Instant Field-Aligned Meshes* was used [48]. The differences between the two meshes are shown with Figure 15.

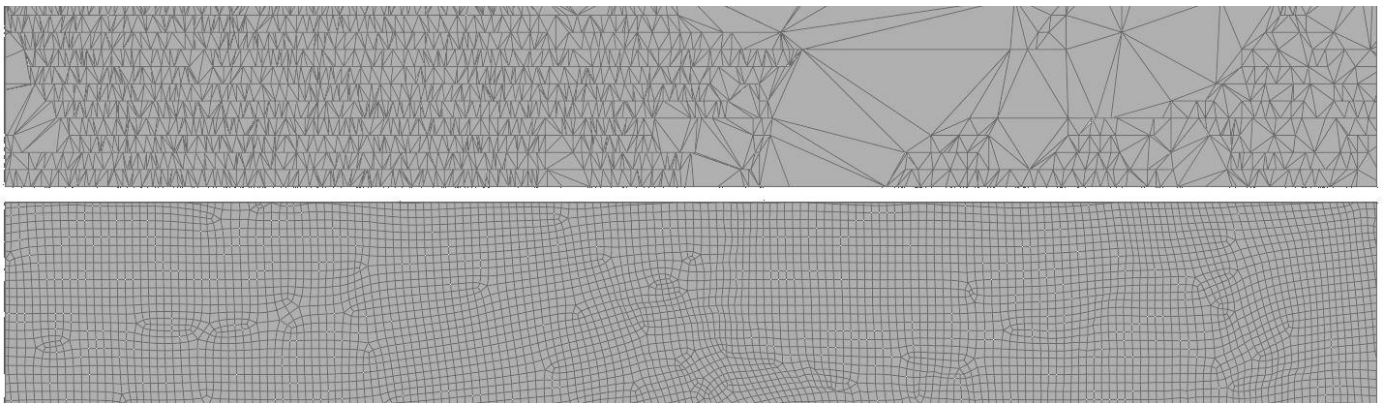


Figure 15. Top: initial mesh; bottom: remeshing with *Instant Field-Aligned Meshes*.

To populate the terrain with natural elements such as trees and ground elements, the instancing system was used. Figure 16 shows how the natural environment can be quickly created in this manner, ensuring a visually satisfying render that is adjustable in density and position.

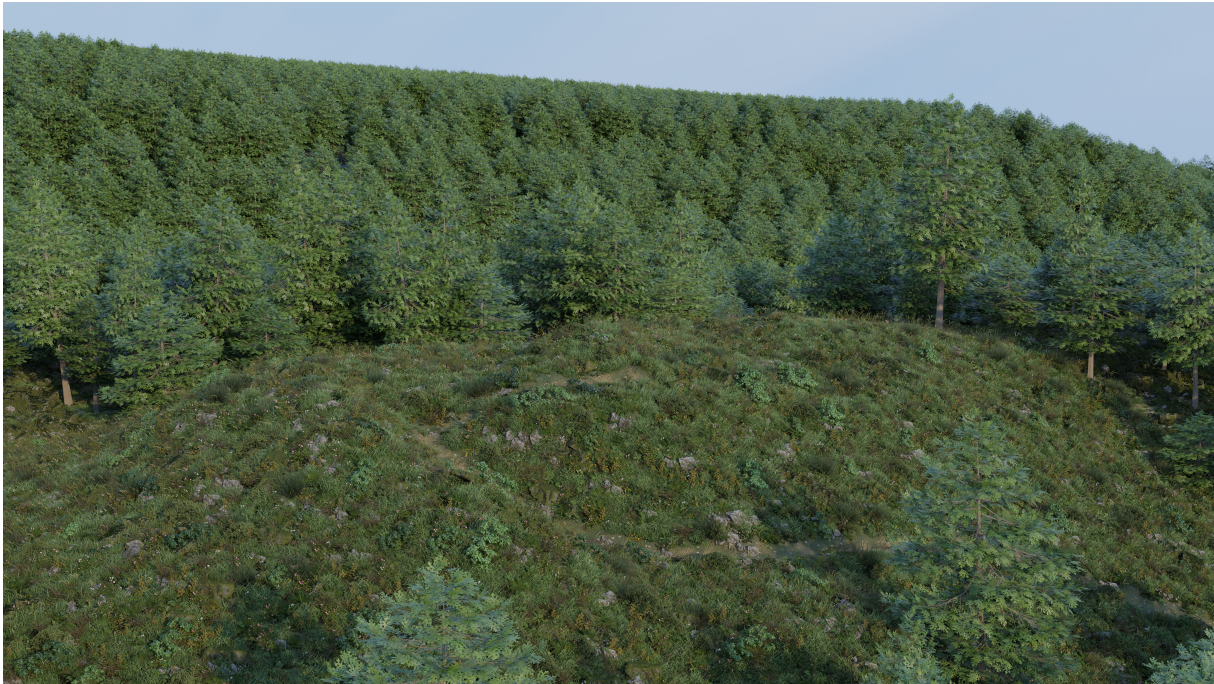


Figure 16. DTM around the castle with its natural environment.

8. Analysis of the Final Results

Once all the individual elements have been configured, they can be assembled to create the final model of the castle within its environment. Using the IGN's High-Density LiDAR data, it is possible to retrieve the points corresponding to the castle ruins. These points are used to define the footprint of the walls, and therefore to create the curve on which the stones will be automatically generated using the associated *Geometry Nodes* (as shown in Figure 5). Elements of the castle such as hoardings, arrow slits, battlements, etc., can be placed using the point cloud. It is rare for the details of the structures to still be present on the ruins. To obtain the details of the elements to be added, it is necessary to consult specialists. Changes can then be made quickly thanks to the parametric nature of the model. It is also important to correctly adhere to the coordinate shift that was made to create the DTM, to ensure that the model of the castle is accurately positioned in relation to its environment.

8.1. Proposals for Restoring the Historical State of the Castle

Cycles, the ray tracing engine integrated into Blender, is specifically designed to deliver photorealistic renderings through its path tracing algorithm. This advanced engine accurately simulates how light interacts with objects in a virtual environment, capturing complex effects such as reflections, refractions, and soft shadows. *Cycles* is highly optimized for GPU rendering, although it also supports CPU rendering, thus providing increased flexibility depending on the available hardware resources. Furthermore, *Cycles* benefits from deep integration with Blender's node-based material system, facilitating the creation of complex textures and fine-tuning the optical properties of materials. This ability to replicate realistic light interactions makes *Cycles* particularly effective for scientific visualisations, architectural animations, and film productions that require a high degree of realism. The number of iterations and noise reduction can be adjusted to obtain the right balance between rendering time per image and the desired final quality. Since this project aims to enhance the site for non-specialists, the visual aspect plays a crucial role. Therefore, a significant number of iterations are necessary for image calculations to ensure an attractive render, as illustrated by Figure 17.



Figure 17. Proposition of reconstruction of the Birkenfels castle.

8.2. Analysis of Flexibility in Restoration Proposals

Since the project aims to enhance Rhine castles while ensuring the accuracy of the results, it is necessary to present several renderings with the same quality in the final output, which will have been validated by archaeologists. Each parameterisation must then ensure the correct topology of the created objects, as well as a good application of textures. Since all the objects are parametric and have been designed as such, procedural textures are applied and do not require *UV* mapping. The models have been configured so that the modelling time is not dependent on the final size of the castle, as the dimensions and heights are determined only by curves and numerical values, allowing each object to be automatically modelled. For example, a model such as the one shown in Figure 17 can be produced in around 1 to 2 weeks, which is considerably shorter than the 2 to 5 months generally required for manual modelling that has already been carried out on this type of castle. In general, the main elements present in this model and their parameterisation methods are as follows:

- **Tiling:** The tiles are generated automatically using a *Geometry Nodes* setup. From a simple object representing the roof (a roof section is represented by a face), each face is separated according to its slope to determine the faces on which tiles are to be placed. Then, a bevelled cube is instantiated on each face to create the tiles. Since the object is instantiated, the number of triangles in the scene remains the same, and the computer resources can easily handle this model regardless of the number of tiles. Additionally, if the roof shape changes or if additional faces are added, the tiles are added or removed in real time. If a particular type of tile is to be applied, for example, following archaeological discoveries, it is sufficient to replace the input cube with a single hand-modelled tile. A random colour is assigned to each tile from a pre-selected set of colours to add realism to the scene.
- **Tiles on the roof joints:** The edges between each face of the surface used for the roofing are analysed to determine if the faces are concave or convex. If they are convex and have a slope, then linear tiles are applied on the edge to join the different sections of the roof. The initial tile model is instantiated along each selected edge, and each instance is oriented according to the normal and tangent of each edge to ensure proper

alignment. The modification of the input mesh and its impact on the tiling is shown in Figure 18, with the changes taking only a few seconds.

- **Castle stone walls:** Created automatically from a single hand-modelled stone, then applied to each edge of the curve, giving the castle's footprint.
- **Hoards:** Modifiable in number and dimension from a few manually modelled objects, and duplicated along a curve.
- **Vegetation density:** The vegetation models, their distribution, and their density are entirely parametric.
- **Windows:** Created with a specific *Geometry Nodes* setup, allowing their duplication and scaling.
- **Arrow slits:** They are integrated directly into the parametric structure of the wall, and can be modified in height, width, number and position.
- **Rock under the castle:** Created with *Geometry Nodes*, allowing noise to be applied to the input mesh that defines the approximate size of the rock.
- **Battlements:** Modifiable in width, height, and number of stones, always adjusting the size of the merlon.



(a)



(b)

Figure 18. Two different versions of automatic roof tiling. (a) Tiling inside the parapet. (b) Tiling over the merlons.

9. Conclusions

To conclude, this paper proposes a parametric modelling method using the Blender free and open-source software to create virtual environments representing Rhine castles from the Middle Ages. The presented methods allow for quick modifications of elements in size, number, and position, requiring only a limited number of manually modelled objects. In the context of Rhine castles, which are now in ruins, this method has the advantage of facilitating exchanges with archaeologists specialising in this period, whose role is to validate the models. Similarly, if the models are not historically accurate and contradict archaeological findings from the site, the established method allows for rapid modifications of the digital reconstruction proposals.

One of the major advantages of this approach is the significant time saving it offers compared to manual modelling techniques. This time is saved not only during the initial modelling process but also during subsequent modifications that arise from discussions with archaeologists, which can often be numerous. The time efficiency is achieved through the use of *Geometry Nodes*, which allow the same base structure to generate multiple elements that differ based on parameter adjustments. Furthermore, additional time is saved in the application of textures, as the method relies on procedural textures rather than requiring extensive preparatory work on *UV* mapping for each modification.

Additionally, since this project aims to promote the sites and share them with the public, visual rendering plays an important role. Applying textures to the created models adds colour, details to objects, and reflection to materials, thanks to the rendering capabilities of the software, while ensuring that the geometry remains light and manageable.

10. Implications and Future Prospects

The library of parametric objects used for this castle will need to be expanded to allow for modelling other castles from different periods or using different materials. The castles will be selected based on their historical importance and their need for promotion to encourage public visits. These parametric objects can only be used and duplicated for structures that share a minimum set of common features and rules. For more specific elements, a distinction must be made. Some unique elements, like certain sculptures, can be generated from simple geometries and parameterised. On the other hand, more complex geometries, such as those representing historical figures, events, or intricate coats of arms, are more challenging to integrate into a parametric workflow due to their complexity and the need to faithfully reflect real elements that do not conform to a general scheme. In such cases, parametric environments could be complemented by elements modelled directly from reality, particularly using photogrammetry, to ensure high-quality textures and accurate representation. Ref. [49] present unique elements found in the context of Rhine castles. Some of them, such as sculpted stones, cannot be parameterised, as they depict complex human figures that do not conform to specific criteria. On the other hand, other elements, such as lamps, can be recreated parametrically.

The method proposed here allows for the use of the actual *DTM* around the castle to integrate it into its environment. However, the *DTM* must be limited to avoid overloading the computer's resources. This creates a border in the environment during rendering in certain directions. Work is ongoing to solve this issue by proposing parametric modelling of an environment around the actual *DTM* to give a sense of a complete environment, combined with atmospheric and lighting renders.

Additionally, for the finished models, the transfer to virtual or mixed reality systems has to be carried out. The study will need to analyse the different existing systems, compare them, and determine which methods are most suitable for meeting the visualisation and promotion needs of the sites. One issue that has already arisen is the transition of procedural

textures from Blender to other software. To solve it, texture baking works well for objects with a low number of faces. Texture baking converts a procedural texture, generated algorithmically, into a texture from a fixed image file. In this way, the texture file can be exported as a conventional image file (.PNG or .JPEG, for example) for use in other software without any compatibility problems. In addition to the diffuse map, Blender also enables light effects such as reflections or specularities of materials to be baked in. This saves a considerable amount of time in rendering, since the lighting effects are pre-calculated, which is useful, for example, for VR rendering, but it also limits the variations in light in the scene once the baking has been completed. The choice therefore has to be made on a case-by-case basis, depending on the specifications and the methods used to enhance the project. For objects with a larger number of faces, one idea is to break them down into multiple elements to create several texture files, being careful not to process the same face multiple times. This method is currently in development. The baking of procedural textures can only be performed on a specific and fixed geometry. As such, prior validation by archaeologists is essential to ensure that the geometry accurately reflects their interpretations and does not require further modifications. This validation step is crucial to avoid the need for repeated baking processes, which would significantly increase the workload and processing time.

This parametric modelling process then raises an important question relating to the valuation and diffusion of these models. The people targeted by this study do not necessarily have 3D modelling skills, so it is important to simplify access for them as much as possible. Ref. [50] proposes several possibilities for presenting models, parametric or not, in different environments. The importance of contextualisation is presented, in order to easily make the link between the past and what is visible today. In addition, the use of different sharing methods is important. Determining the most appropriate ones for enabling the public to easily understand how the models were created is a major challenge, whether using virtual reality or more traditional 3D renderings such as images or videos.

Author Contributions: Conceptualization, E.S. and M.K.; methodology, E.S.; software, E.S.; validation, E.S., M.K. and P.G.; formal analysis, E.S.; investigation, E.S.; resources, E.S.; data curation, E.S.; writing—original draft preparation, E.S.; writing—review and editing, E.S., M.K. and P.G.; visualization, E.S.; supervision, M.K. and P.G.; project administration, M.K. and P.G.; funding acquisition, M.K. and P.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by “Châteaux Rhénans: Burgen am Oberrhein” Interreg VI Project (2023–2025).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We extend our thanks to the “Châteaux Rhénans: Burgen am Oberrhein” Interreg VI Project (2023–2025) involving France, Germany, and Switzerland, coordinated by the European Collectivity of Alsace. Our work aligns with Action 4.6, focusing on 3D valorisation of heritage sites. We appreciate the opportunity to contribute to this innovative approach to digital heritage conservation. Moreover, the authors would like to thank Marthe PASSAT and Etienne FRITSCH for their valuable assistance with the archival documents and their explanations regarding the history of the Birkenfels castle. Finally, the authors would like to express their gratitude to Francesco Fassi, Stefano Campana and Fabio Remondino for kindly allowing us to publish in this special issue. This opportunity was made possible by the Best Paper Award obtained at the 10th International ISPRS/CIPA 3D-ARCH workshop on ‘3D Virtual Reconstruction and Visualization of Complex Architectures’, which was held in Siena (Italy) from 21 to 23 February 2024 [51–53].

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BIM	Building Information Modelling
DTM	Digital Terrain Model
GIS	Geographic Information System
HBIM	Historic Building Information Modelling
IGN	Institut National de l'Information Géographique et Forestière
INSA	Institut National des Sciences Appliquées
LiDAR	Light Detection and Ranging
MEL	Maya Embedded Language
NURBS	Non-Uniform Rational Basis Splines

References

1. Sampaio, A.Z.; Tomé, J.; Gomes, A. Heritage Building Information Modelling Implementation First Steps Applied in a Castle Building: Historic Evolution Identity, Data Collection and Stratigraphic Modelling. *Heritage* **2023**, *6*, 6472–6493. [CrossRef]
2. Drap, P.; Gaillard, G.; Grussenmeyer, P.; Hartmann-Virnich, A. A stone-by-stone photogrammetric survey using architectural knowledge formalised on the ARPENTEUR Photogrammetric workstation. In Proceedings of the XIXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS), Geoinformation for All, Amsterdam, The Netherlands, 16–23 July 2000; Volume XXXIII, pp. 187–194.
3. Foschi, R.; Fallavollita, F.; Apollonio, F.I. Quantifying Uncertainty in Hypothetical 3D Reconstruction—A User-Independent Methodology for the Calculation of Average Uncertainty. *Heritage* **2024**, *7*, 4440–4454. [CrossRef]
4. Châteaux forts Alsace. Châteaux Forts Alsace-Château Fort. Available online: <https://www.chateauxfortsalsace.com/fr/ressources-pedagogiques/chateau-fort/> (accessed on 10 November 2024).
5. De Luca, L.; Véron, P.; Florenzano, M. A generic formalism for the semantic modeling and representation of architectural elements. *Vis. Comput.* **2007**, *23*, 181–205. [CrossRef]
6. Remondino, F.; Lo Buglio, D.; Nony, N.; Luca, L. Detailed Primitive-Based 3d Modeling of Architectural Elements. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, XXXIX-B5, 285–290. [CrossRef]
7. Volk, R.; Stengel, J.; Schultmann, F. Building Information Modeling (BIM) for existing buildings—Literature review and future needs. *Autom. Constr.* **2014**, *38*, 109–127. [CrossRef]
8. Banfi, F.; Barazzetti, L.; Previtali, M.; Roncoroni, F. Historic BIM: A New Repository for Structural Health Monitoring. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, XLII-5/W1, 269–274. [CrossRef]
9. Banfi, F. Building Information Modelling—A Novel Parametric Modeling Approach Based on 3D Surveys of Historic Architecture. In *Proceedings of the Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection*; Springer: Cham, Switzerland, 2016; pp. 116–127. [CrossRef]
10. Murtiyoso, A.; Grussenmeyer, P.; Suwardhi, D.; Awalludin, R. Multi-Scale and Multi-Sensor 3D Documentation of Heritage Complexes in Urban Areas. *ISPRS Int. J. -Geo-Inf.* **2018**, *7*, 483. [CrossRef]
11. Chevrier, C.; Perrin, J.P. Generation of architectural parametric components. In Proceedings of the CAAD Futures 09. Les Presses de l'Université de Montréal, Montreal, QC, Canada, 17–19 June 2009; p. 105.
12. Barazzetti, L.; Banfi, F.; Brumana, R.; Previtali, M. Creation of Parametric BIM Objects from Point Clouds Using Nurbs. *Photogramm. Rec.* **2015**, *30*, 339–362. [CrossRef]
13. López, F.J.; Lerones, P.M.; Llamas, J.; Gómez-García-Bermejo, J.; Zalama, E. A Review of Heritage Building Information Modeling (H-BIM). *Multimodal Technol. Interact.* **2018**, *2*, 21. [CrossRef]
14. Dore, C.; Murphy, M. Integration of Historic Building Information Modeling (HBIM) and 3D GIS for recording and managing cultural heritage sites. In Proceedings of the 18th International Conference on Virtual Systems and Multimedia, Milan, Italy, 2–5 September 2012; pp. 369–376. [CrossRef]
15. Yang, X.; Lu, Y.C.; Murtiyoso, A.; Koehl, M.; Grussenmeyer, P. HBIM Modeling from the Surface Mesh and Its Extended Capability of Knowledge Representation. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 301. [CrossRef]
16. Chevrier, C.; Charbonneau, N.; Grussenmeyer, P.; Perrin, J.P. Parametric Documenting of Built Heritage: 3D Virtual Reconstruction of Architectural Details. *Int. J. Archit. Comput.* **2010**, *8*, 135–150. [CrossRef]
17. Aicardi, I.; Chiabrando, F.; Lingua, A.M.; Noardo, F. Recent trends in cultural heritage 3D survey: The photogrammetric computer vision approach. *J. Cult. Herit.* **2018**, *32*, 257–266. [CrossRef]

18. Benazzi, T. Restitution 4D du Château du Kagenfels par Combinaison de l'Existant et d'Hypothèses Archéologiques pour une Visite Virtuelle du Site. Master's Thesis, INSA De Strasbourg, Strasbourg, France, 2018. Available online: <http://eprints2.insa-strasbourg.fr/3513> (accessed on 10 November 2024).
19. Heitz, J.E. Modélisation 4D du Château de l'Oedenbourg (Petit Koenigsbourg) par Combinaison de Relevés de Terrain et d'Hypothèses Archéologiques. Master's Thesis, INSA Strasbourg, Strasbourg, France, 2023. Available online: <http://eprints2.insa-strasbourg.fr/5069> (accessed on 10 November 2024).
20. Rigaud, D. Restitution 4D du Château de Wasenbourg dans le Cadre du Projet INTERREG VI. Master's Thesis, INSA De Strasbourg, Strasbourg, France, 2023. Available online: <http://eprints2.insa-strasbourg.fr/5073> (accessed on 10 November 2024).
21. Koehl, M.; Heitz, J.E.; Rigaud, D.; Guillemain, S. Process of Creating and Comparing 4D Models of Rhenish Castles in Different Modeling Environments. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *XLVIII-2/W4-2024*, 263–270. [CrossRef]
22. Barazzetti, L.; Banfi, F.; Brumana, R.; Gusmeroli, G.; Previtali, M.; Schiantarelli, G. Cloud-to-BIM-to-FEM: Structural Simulation with Accurate Historic BIM from Laser Scans. *Simul. Model. Pract. Theory* **2015**, *57*, 71–87. [CrossRef]
23. Garagnani, S.; Manferdini, A.M. Parametric Accuracy: Building Information Modeling Process Applied to the Cultural Heritage Preservation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *XL-5/W1*, 87–92. [CrossRef]
24. Remondino, F. Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. *Remote Sens.* **2011**, *3*, 1104–1138. [CrossRef]
25. Comte, F.; Pamart, A.; Réby, K.; De Luca, L. Strategies and Experiments for Massive 3D Digitalization of the Remains After the Notre Dame de Paris' Fire. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *XLVIII-2/W4-2024*, 127–134. [CrossRef]
26. Walmsley, A.; Kersten, T.P. Low-Cost Development of an Interactive, Immersive Virtual Reality Experience of the Historic City Model Stade 1620. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W17*, 405–411. [CrossRef]
27. Putra, G.B.; Agirachman, F.A. Experiencing Heritage through Immersive Environment using Affordable Virtual Reality Setup. In *Proceedings of the International Webinar on Digital Architecture 2021 (IWEDA 2021)*; Atlantis Press: Amsterdam, The Netherlands, 2022; pp. 51–57; ISSN: 2352-5398. [CrossRef]
28. Bolognesi, C.M.; Manfredi, V. Optimization of VR Application in Texturing Cultural Heritage. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *XLVIII-2/W4-2024*, 73–78. [CrossRef]
29. Ferdani, D.; Demetrescu, E.; Cavalieri, M. Interpreting and Visualizing the Past Through Virtual Archaeology: From Site to Museum Experience. In *Digital Restoration and Virtual Reconstructions: Case Studies and Compared Experiences for Cultural Heritage*; Trizio, I., Demetrescu, E., Ferdani, D., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 3–18. [CrossRef]
30. Boeykens, S. Using 3D Design software, BIM and game engines for architectural historical reconstruction. In *Proceedings of the 14th International Conference on Computer Aided Architectural Design Futures*, Liège, Belgium, 4–8 July 2011; pp. 493–509.
31. Banfi, F. BIM Orientation: Grades of Generation and Information for Different Type of Analysis and Management Process. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-2/W5*, 57–64. [CrossRef]
32. Banfi, F. HBIM, 3D drawing and virtual reality for archaeological sites and ancient ruins. *Virtual Archaeol. Rev.* **2020**, *11*, 16–33. [CrossRef]
33. Gyss, A.J. *Histoire de la Ville d'Obernai, I–II*; Lafitte Reprints: Marseille, France, 1978; Volume 1.
34. Biller, T.; Metz, B. *Die Burgen des Elsass, Band III: Der Frühe Gotische Burgenbau im Elsass (1250–1300)*; Deutscher Kunstverlag: München, Germany, 1995.
35. *Encyclopédie de l'Alsace*; Éditions Publitotal: Strasbourg, France, 1983; Volume 2.
36. Metz, B. *Association Pour la Sauvegarde de l'Architecture Médiévale*, Rapport de fouilles 67 3 18 368; 1974.
37. Charles, J. Birkenfels; les pierres de taille de la terrasse Nord Est. *Bull. L'Assoc. Pour Sauvegarde L'Archit. Médiévale* **1977**, *4*, 20–24.
38. Bischoff, G.; Rudrauf, J.M. *Les Châteaux-Forts Autour du Mont Sainte-Odile: La Couronne de Pierre des ducs d'Alsace*; I. D. l'édition: Bernardsviller, NJ, USA, 2019.
39. Carmona, C.; Trendel, G. *La région du Mont Sainte-Odile, Landsberg, Dreistein, Waldsberg, Kœpfel, Rathsamhausen, Lutzelbourg, Birkenfels, Kagenfels*; Imprimerie Régionale: Strasbourg, France, 1994; Volume 2.
40. Blender. Geometry Nodes-Introduction-Version 4.2. Available online: https://docs.blender.org/manual/en/latest/modeling/geometry_nodes/introduction.html (accessed on 10 November 2024).
41. Blender. Modifiers-Introduction-Version 4.2. Available online: <https://docs.blender.org/manual/en/latest/modeling/modifiers/introduction.html> (accessed on 10 November 2024).
42. Mathieu, G. Carcassonne's Hoardings. Available online: <https://pbs.twimg.com/media/FZTUa3AXEAAxi9.jpg> (accessed on 10 November 2024).
43. Perlin, K. An Image Synthesizer. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 22–26 July 1985; SIGGRAPH '85, pp. 287–296. [CrossRef]
44. IGN. Institut National de l'Information Géographique et Forestière–LiDAR HD Data. Available online: <https://geoservices.ign.fr/lidarhd/> (accessed on 18 November 2024).
45. CloudCompare (Version 2.11.3). Available online: <http://www.cloudcompare.org> (accessed on 10 November 2024).

46. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sens.* **2016**, *8*, 501. [[CrossRef](#)]
47. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson Surface Reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry processing (SGP '06), Cagliari, Sardinia, Italy, 26–28 June 2006; pp. 61–70. [[CrossRef](#)]
48. Jakob, W.; Tarini, M.; Panozzo, D.; Sorkine-Hornung, O. Instant Field-Aligned Meshes. *ACM Trans. Graph.* **2015**, *34*, 1–15. [[CrossRef](#)]
49. Sommer, E.; Koehl, M.; Grussenmeyer, P. Using Hyperfocal Distance in Close-Range Photogrammetry for Indoor Modelling. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *XLVIII-2/W7-2024*, 161–168. [[CrossRef](#)]
50. Koehl, M.; Heitz, J.E.; Sommer, E.; Fuchs, M. Valorization of 3D/4D models of disappeared or ruined heritage sites in 3D modeling environments. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *XLVIII-2/W8-2024*, 235–242. [[CrossRef](#)]
51. Sommer, E.; Koehl, M.; Grussenmeyer, P. Crafting and Modifying Rhine Castle Models with Parametric Modeling in Blender. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *XLVIII-2/W4-2024*, 405–412. [[CrossRef](#)]
52. Blender Foundation. Blender—A 3D Modelling and Rendering Package. Available online: <https://www.blender.org/> (accessed on 10 November 2024).
53. Visitgrandest. Drone View of the Birkenfels Castle. Available online: <https://visitgrandest.com/chateau-de-birkenfels> (accessed on 10 November 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.