Valorization of 3D/4D models of disappeared or ruined heritage sites in 3D modeling environments

Mathieu Koehl [1], Jade-Emmanuelle Heitz [1], Etienne Sommer [1], Matthieu Fuchs [2]

[1] Université de Strasbourg, CNRS, INSA Strasbourg, ICube Laboratory UMR 7357, Photogrammetry and Geomatics Group Strasbourg, France - (mathieu.koehl, jade-emmanuelle.heitz, etienne.sommer)@insa-strasbourg.fr
[2] Archéologie Alsace, Sélestat, France – matthieu.fuchs@archeologie.alsace

Keywords: 3D modeling, Cultural heritage, Valorization, Rendering, AR/VR, Blender.

Abstract

3D is an increasingly common tool in many fields, including archaeology. It allows creating what no longer or almost no longer exists based on many types of information: archaeological excavation, plans, photographs, archaeological or historical hypothesis and many others. 3D is a tool of popularizing knowledge by making it more visual (photorealistic renderings of 3D restitutions and virtual reality), appreciable in a targeted context, and even by making it palpable (3D printing). By associating 3D representations/modelling tools and archaeology, we then move to 4D, integrating the notion of time and allowing people to travel through the centuries in a simple and playful way. This study shows the aspect of valuation of archaeological 3D data through 3D textured and animated restitutions, put in their current urban context. The use of 3D modelling and animation software makes it possible to give a digital and virtual appearance and consistency to historical and heritage sites. These models allow for the restoration of shapes and dimensions, and their placement within a historical context. To help users better project themselves, it is also interesting to place them in current urban contexts. But a model is not always enough on its own, we must find ways to make them visible, to make them understandable, and to promote them to neophyte audiences on the one hand, but also to make them accessible to expert audiences such as historians and archaeologists in the other hand.
To illustrate the point, we used modelling projects on a Roman Castellum and a castle of the Renaissance period, both located in Horbourg-Wihr, Alsace, France. We also use models of Rhine castles carried out as part of an Interreg VI project – Châteaux Rhénans – Burgen am Oberrhein (Interreg, 2024).

1. Introduction

The historical numerical model is a powerful tool. Allowing to preserve and/or recreate, it is a vector of knowledge that is not always easy to interpret for non-experts. Whether it is a metric model or a simple representation of an archaeological hypothesis, digital historical restitution is a fun way to understand and convey knowledge. This model makes it possible to understand the morphology, color, texture, and even the environment of a historic site. This is why this study focuses on the promotion of digital heritage models for the public.
In the case of a digital reconstruction of a building that still exists, the public can rely on the elements still present on the site. It can recognize the shape, size, or materials used. However, in the case of a monument that has disappeared, such as the castle or the Gallo-Roman castellum of Horbourg-Wihr, the audience may find it difficult to project themselves into the scene because they have no element of comparison. In this case, it may be interesting to put the models in a contemporary context for the viewers. In our case study, we have therefore chosen to recontextualize the models using background maps such as the land registry or satellite view, using digital terrain models, or using digital models of the city with its buildings.
In the end, it is a transition from 2D to 3D for a better understanding of the audience and a more detailed and personalized visualization of 3D and even 4D renderings in their current context.

2. Context and historical elements at the modelled sites

The castellum of Horbourg-Wihr is a military camp of the Roman Empire dating from the end of the 4th century AD. It was built in the heart of an ancient city, which is located north of the city of the Rauraques, in the province of Grande Séquanaise (Maxima Sequanorum). Since the end of the 3rd century, this border region of the Empire has been the subject of Germanic incursions, the Alemanni in particular. The construction of this camp enabled the placement of an important military lock in the Alsace plain, in the second line, supporting the camps placed along the Rhine. It measures approximately 164 x 176 m on sides occupying 2,89 ha. (Fuchs, 1996). At the beginning of the 5th century, the region was conquered by the Alemanni, and the camp was abandoned. A community was established in its ruins at least from the 7th century and the remains of the walls are still visible in the 16th century. In 1543, the construction of a castle by the Wurtembergs on the northeast corner of the castellum reused the building materials. The layout of the camp was identified around 1780, and excavations uncovered several portions of the ramparts and towers in the 1880s. Recent excavations have found remains in several places and have made it possible to propose a reconstruction of the camp. It corresponds to a well-known model in the Rhine regions at the end of the 4th century, which has made it possible to draw comparisons with better-preserved camps. This is the basis of the 3D modelling that has been made.
Today, there are no visible remains, the entire camp being located under the contemporary city. The disappeared ramparts can be found in the streets of the city, which take up the layout, but the fortress has totally disappeared from the memory of the inhabitants. The georeferencing of this Roman camp in the current city makes it possible to better protect this heritage during work and to promote this prestigious historical past to the population, heritage enthusiasts, fellow citizens, and tourists.
The Wurtemberg Castle in Horbourg-Wihr is a Renaissance residence castle that Count George I had built in 1543 and that Duke Frederick I had enlarged in 1597 by his architect Heinrich Schickhardt (Fuchs, 1997). This castle is the seat of the Wurtemberg possessions on the left bank of the Rhine, in the Holy Roman Empire. It takes over from a medieval castle,

already attested in the 12th century, and is located at the northeast corner of the Roman castellum. It was dismantled in 1675 by order of Louis XIV, after Alsace had become French at the end of the Thirty Years' War. The buildings fell into disrepair and were sold as national property during the French Revolution. The remains were used as a stone quarry for building, and in the middle of the 19th century nothing was visible, the buildings were razed and the ditches partially filled in. The area of the residence buildings and the 3/4 of the castle's ditches are still visible on the cadastral plot. These elements enabled the architect Charles Winkler to position the entire castle in a perspective view engraved before the demolitions of 1675 around 1890.

Archaeological operations since the beginning of the 21st century have made it possible to find its location and to verify the relative accuracy of the engraving and plans. The Wurtemberg state archives also hold plans for a wing of the castle added by Heinrich Schickhardt. This data could be compared with other 16th century constructions, and particularly the castle of Riquewihr, built by Count George in 1540, as well as other buildings by the architect Schickhardt. 17th-century engraving formed the main basis for 3D modeling (Cartier, 2020).

Today there are no visible remains, except for the south-west corner of the counterscarp wall, the footprint of the castle being located under the contemporary city. The existence of this castle has also disappeared from the memory of the inhabitants.

The georeferencing of this castle in the current town makes it possible to better protect this heritage during development work and, as with the castellum to promote this prestigious historical past to the population, heritage enthusiasts, fellow citizens, and tourists. For these two examples, 3D modeling provides a scientific and architectural basis that can evolve as new discoveries and researchers' hypotheses evolve. For the public, the model resurrects a disappeared monumental heritage and becomes the marker of a heritage that is invisible on the surface but partially preserved in the subsoil of the city.

These models were carried out as part of a master’s degree-thesis by students (Cartier, 2020; Nivola, 2018).

3. 3D modeling platform

3.1 *Blender*: free and open source

There are many platforms and software for 3D modelling, rendering, and animation. Among these, free and open-source software have great roles to play, since their intensive development makes it possible to achieve performance equivalent to commercial software. This is the case of *Blender* (2024) for which *Blender*'s comprehensive array of modelling tools make creating, transforming, sculpting, and editing your models a breeze. The enhancement of models involves renderings that use the classic triptych: model - camera - light. The paper proposes to explain and analyze the different rendering methods used to valorize models of Roman camps and castles that have disappeared or still exist in the form of ruins. The first models were carried out in environments such as *Trimble Sketchup* or *Autodesk Maya*, and then were taken up in the *Blender* environment. The choice fell on this environment, which also includes a large community of users and developers providing a vast library of tools and add-ons, which is continuously enriched.

3.2 *Blosm – OSM* add-on for *Blender*

We use the *Blosm* add-on known as *Blender-OSM*. This add-on offered by Prochitecture (2024) allows the import of 2D and 3D geographic data such as digital terrain models, natural elements (lakes, rivers, vegetated areas, etc.), built elements (buildings, roads, etc.), or various images. This add-on provides data from *Open Street Map*, *Google 3D tiles*, *ESRI ArcGIS*, *Mapbox*, and others (Blosm, 2024).

3.2.1 Digital Terrain Models (DTM): The digital terrain model is imported (*Blosm*'s *Terrain* function) from the *Open Street Map* database. The tool then provides a terrain model with a resolution of 30 meters for a spatial footprint of up to more than 1500 square km. The *Extent tab* allows to select the desired plot of land by its coordinates. A select link to the Prochitecture website allows to retrieve these coordinates by selecting a rectangular area on an *Open Street Map* or *ArcGIS* satellite basemap and thus define the *Blosm Extent* parameter for *Blender* (note that this footprint will be kept for all subsequent processing carried out using *Blosm*). It is then possible to set up the import in *Terrain* mode before importing the data.

Blender is not initially designed to manage cartographic coordinate systems; a model can then be georeferenced according to the zero of the *Blender* coordinate system or according to an old import thanks to the *Relative to initial support* function. But *Blosm*'s *bpyproj* add-on offers the choice of assigning a map projection system for data imports made with *Blosm*.

3.2.2 Basemap: Basemaps can be imported using *Blosm* and the *Overlay Image* function. This function allows you to import images as textures applied to an object. The latter can be a previously imported 3D terrain model or any other object. The data provided can come from a variety of sources: *ArcGIS Satellite*, *Mapbox Satellite*, *OSM Mapnik*, *Mapbox Streets*, or a custom URL. In the case of *Mapbox* and *ArcGIS*, a token is required.

The procedure for selecting the geographical area remains the same. After selecting the data source, it is possible to define whether the image is applied to the model (*Set default material*) and/or saved as an external file (*Save overlay to file*). It should be noted that by importing several *overlay images*, each new implementation overwrites the previous image.

Finally, the user can intervene on the resolution of the imported image via the *Advanced* option of *Blosm* using the *Max number of tiles* parameter. This setting affects the final resolution of the image. Import in the form of tiles also optimizes performance during an import and during a re-import. Indeed, if a tile has already been downloaded by *Blosm*, it will be kept in memory and will not be downloaded again. For our case study, we import the overlay images on a simple plan, Horbourg-Wihr being not very hilly and the visualization being limited to the scale of the village. We then work on satellite views from *Mapbox* (Figure 1a) and *OSM* (Figure 1b). As a reminder, the idea here was to allow a better location of 3D reconstructions within the village of Horbourg-Wihr for the public and particularly the inhabitants, thus allowing them to understand the proximity they have to their municipal heritage.

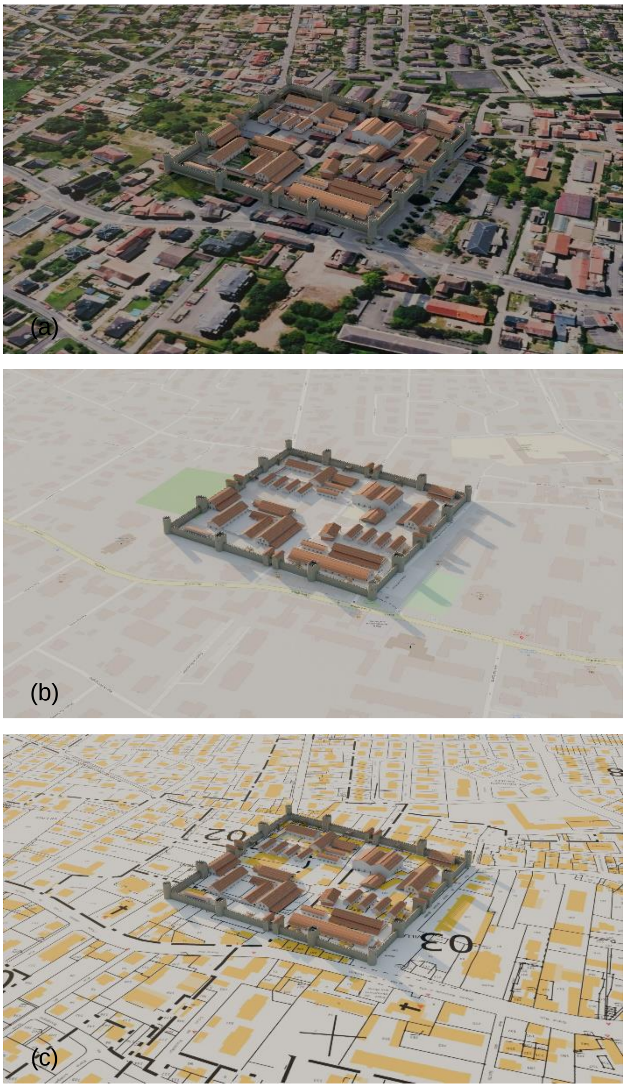
In a second step, we also used the city's cadastral data accessible on the platform (Cadastre, 2024) and available in digital or physical format. This data is in image format and must therefore be georeferenced before any import into *Blender*. As before, cadastral sheets are used as texture on a simple plane (Figure 1c).

3.3 Characteristic built and natural elements

In addition to the digital terrain model and *overlay images*, it is possible to import significant and recognizable elements of the Horbourg-Wihr environment, such as buildings and road layouts, among others. Two approaches are presented here.

The first uses *Blosm*. Indeed, the add-on also allows you to import many elements in 2D and 3D from the *OSM* database.

These elements are diverse: buildings, waterbodies, forest surfaces, vegetated surfaces, roads/paths, and railway tracks.



Figures 1: (a) Gallo-Roman Castellum of Horbourg-Wihr superimposed on satellite imagery, (b) on the *OSM* map, (c) on the cadastral map

Obviously, the decomposition of these classes depends on the detail of the *OSM* data. On the same footprint as before, the *OpenStreetMap* function of the *Blosm* add-on offers the possibility to choose the classes of elements to be imported by specifying their dimensions in 2D or 3D. Regarding the import of buildings, it is possible to choose different representation and organization parameters if these are not defined in the *OSM* database. Indeed, it is possible to indicate the shape of the roofs (flat or 2-sided, among others), the average height of the floors for those with no height information. It is also interesting to note that this average height can be randomized according to a customizable weight. Regarding the organization, it is possible to import all the buildings by merging them into one object or to separate them into individual buildings, the latter way allowing them to discern nature and even the name of the buildings if they are detailed in the *OSM* database (Figure 2).
The second method allows user to import 3D mesh and textured tiles sourced from *Google Maps* or *Google Earth*. This method requires the *MapsModelsImporter* (2024) add-on for Blender paired with the *Render Docs* software. Indeed, the method is broken down into two parts: the screenshot of the area to be imported as a 3D tile, then the import of the corresponding 3D tiles. To do this, *Render Docs* allows you to take .rdc format screenshots, a raster containing the 3D data of the specified Google Earth area. The *MapsModelsImporter* add-on then allows you to import .rdc files into *Blender*.

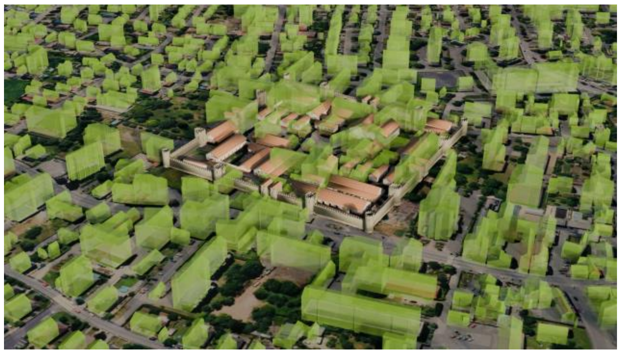


Figure 2: Gallo-Roman Castellum of Horbourg-Wihr superimposed on satellite imagery and *OSM* buildings

The resulting model contains the ground, buildings, but also trees, etc. Note that the detail of the generated 3D models is inversely proportional to the size of the geographical area, i.e. the resolution of the screenshot is correlated with the resolution of the mesh of the 3D tiles (Google Maps platform, 2024).

4. Enhancement of restitution models

The enhancement of models involves different vectors. The simplest is photorealistic image rendering and video rendering. To improve realism, models with vegetation and realist context were designed.

4.1 Natural environment

First, it is possible to recreate the natural environment of the site. This stage is also a historical restitution, as it requires hypotheses about the shape of the natural terrain, the type of vegetation, but also its density and distribution.

4.1.1 Natural Digital Terrain Model: Regarding the natural terrain, it can be generated through field surveys such as LiDAR surveys or terrestrial laser scanners (Sommer et al., 2024) of the current state, considering knowledge or hypotheses of transformation of the places over time. The natural terrain can also be provided, in the form of a mesh as described above with add-ons like *Blosm*, by drawing on already existing databases like *OSM*. In both cases, the digital terrain model is a representation of the current reality. Depending on the history of the sites, it must be adapted based on archaeological hypotheses. With *Blender*, it is then possible to deform the terrain model using several tools, such as *sculpting* tools or simply to move the vertices of the model to adapt it. In addition, from the point of view of optimizing 3D scenes, it is also necessary to sample differently the parts of soil close to the site and those far away that will not be visualized closely. In our case studies, the terrain extends beyond the geographical boundaries of the historical sites studied. The terrain is an element of immersion in the model. Indeed, the terrain models are spread out several kilometers around the site. Again, *Blosm* provides simplified templates that are available immediately. The returned models can thus be immersed in a near, but also distant environment.

4.1.2 Near and far vegetation: Vegetation is a key point in the viewer's immersion. It offers a deeper dimension to the 3D scene by also providing a scale of comparison to the constructions. In this context, many solutions are offered in *Blender*. For a very realistic effect, the method generally proposed is the distribution of a 3D model on a given surface. However, this method is expensive in terms of performance during processing, but also during rendering. We will address this point in the rest of this section.

For all our projects, we have used the *GScatter* add-on for *Blender*. This add-on allows you to scatter 3D objects in a *Blender* scene by offering a set of tools for parameterizing the distribution, scale, rotation, and geometry factor. *GScatter* is therefore a powerful tool, allowing the user to control their 3D scene and offering a high level of customization. In addition, the add-on works with *Geometry Nodes* (Sommer et al., 2024-1). Each function created is associated with a new network of *Geometry Nodes*, which further increases the user's potential for control over their work. In our case, *GScatter* allows user to create vegetated soils quite quickly, while keeping control of the result. Indeed, the add-on offers many tools to optimize rendering and manipulation. The visualization of objects in the form of a proxy (a low-poly version of a model used to preview a scene) as well as camera culling (determination of faces not visible from the camera's point of view, so as not to calculate them during rendering or visualization) are the perfect examples. Beyond the tools for optimizing scene performance, *GScatter* offers many functions for scattering objects. These also allow the user to optimize their scene since they are applicable to all factors (distribution, scale, rotation, and geometry).

It may be interesting to detail the most frequently used tools in the following: The *Weight Mask* function is used to generate a weight map (drawn by the user using a *Blender Brush*) for which a weight of 0 means that the factor (e.g. distribution density) is at the minimum and a weight of 1 means that the factor is at the maximum. Figure 3 shows a cube on which we have applied a weight map for the scale and density factors of distribution. We can clearly see the correlation between the weights applied and the scale/density of the objects. For a weight around 1, the scale/density is around the maximum value entered.

Another very useful *GScatter* tool is the *Proximity* function, which allows you to manage the distribution of objects around other objects, as described in Figure 4. Other more basic tools, such as randomization of distribution, scale, or rotation, are often used. To go further, the add-on also offers a *Wind* function for the rotation factor, which allows you to add an animation to objects to simulate the wind by playing with the rotation parameters of the object. In addition to the many functions offered by *GScatter*, the add-on offers a library of objects for free registration.

This library is varied, offering low vegetation. It is then possible to download the objects individually or by "environment" bringing together plants likely to grow together and in the same place. This library can be found on the *GScatter* website, (2024) and offers download in .fbx, .gscatter and .abc format.

In our project, we therefore mainly use the *GScatter* library for low vegetation. However, we also use our own rock models, as well as tree models generated using the free *Sapling Tree Gen* add-on, (2024). The add-on allows the user to create custom curve trees (hence its creation via the *curve add* menu). *Sapling Tree Gen* offers many parameters to act on the geometry of the tree (distribution of branches, crown of the tree, height, curvature, etc.), the composition of the branches (diameters, divisions, etc.), the shape, density, and distribution of the leaves, as well as on the animation of the tree object. In addition, the add-on also offers presets corresponding to tree species (maple, pine, willow, etc.) to speed up the process of creating the tree. To optimize *Blender* scenes and therefore computation time during rendering, we also use the *Alpha Trees* (2024) add-on for the creation of forests. Indeed, as described above, the *GScatter* add-on allows you to distribute 3D models on a chosen medium. However, this method is very expensive in terms of performance. To overcome this problem, *Alpha Trees* allows the user to generate 2D models (a textured plan) of trees, then scatter them throughout the scene. This method therefore reduces the number of faces of the models to be dispersed, and therefore the memory required, while maintaining visual consistency and offering a wide range of parameters to the user. Basic tools such as randomization of distribution and scale are proposed. In addition, it is possible, just as *GScatter* proposes, to use weight maps to play with the distribution, density, and scale of trees. It is also possible to add wind to animating the trees.

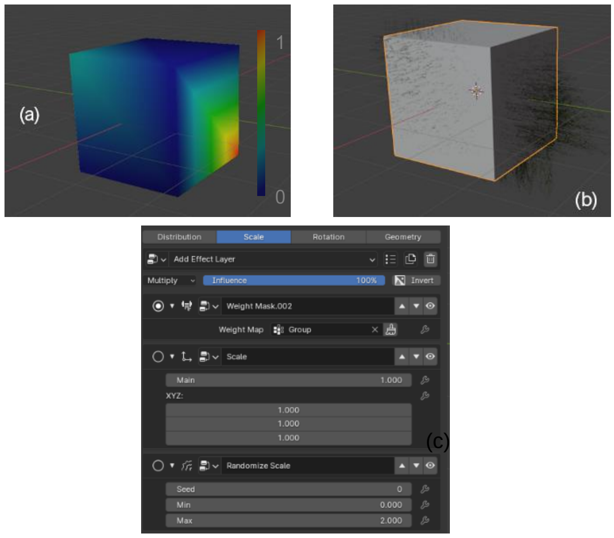


Figure 3: Applying *GScatter*'s *Weight Mask* function to a cube: (a) Weight map applied to the cube with its associated scale. (b) Objects distributed on the surface of the cube according to the weight map. (c) *GScatter* parameters for the distributed object with the various functions applied.

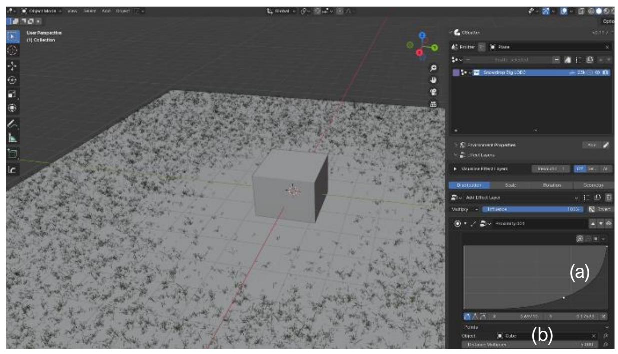


Figure 4: Application of the *Proximity* function on a plane with respect to a cube: here we see the distribution of objects on the plane around the cube according to a curve (a) as well as a distance parameter, here equal to 5 (b).

Beyond that, the add-on offers an automatic (and configurable) randomization of trunk colors, leaves, and even the model itself (with trees without leaves and with leaves). Regarding performance, *Alpha trees* also offer a camera culling module that allows the user to limit the calculation of trees to those visible to the camera. The fluidity during processing and the speed of rendering calculations make *Alpha Trees* a very interesting tool. Obviously, a comparison with *GScatter* is only relevant for renderings of a (distant) landscape, with hundreds of trees. In addition, for closer renderings, we opt for 3D trees for more advanced realism (Figure 5).

Figure 5: Photorealistic renderings of the vegetated areas of the Wasenbourg castle (Niederbronn-les-Bains, Bas-Rhin, France). Vegetation scattered with the *GScatter* add-on.

4.2 Lighting and position of the sun

Still in this quest for realism and immersion of the user, it is also possible to act on the lights of the scene. This work is facilitated by the implementation of *Blender*'s *Sun Position* add-on which offers many sun positioning parameters. The coordinates of the scene and the day of the exhibition can thus be entered to obtain and visualize the real sunshine of the site. *Sun position* offers two modes: *normal* and *Sun + HDRI texture*. Normal mode allows you to maintain control over sky settings (Figure 6).

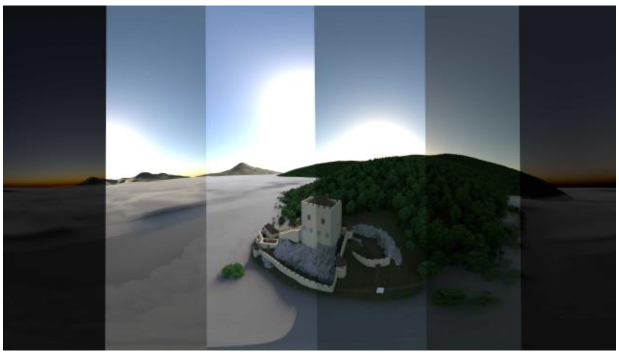


Figure 6: Panoramic render: Evolution of the sun on the Ramstein castle (Scherwiller, Bas-Rhin, France) for a given day in *Blender* using the *Sun Position* add-on.

4.3 Colors

Beyond the realistic texture aspect, it can sometimes be interesting to simplify the representations of the models. In a 4D reconstruction approach, several historical periods stand out for each site. This temporal phasing can then be schematized by a simple colored distinction of each historical epoch. This makes it possible to display the historical phase models simultaneously in the same scene and thus to better understand the extent of each in relation to the other. In the case of Horbourg-Wihr, for example, a transparent tint was applied to the sixteenth-century castle and the sixth-century Gallo-Roman castellum (Figure 7). We can then display the two historical periods in the same scene while ensuring that the historical phasing is understood.

Figure 7: Representation of the Gallo-Roman castellum (blue) and the castle (yellow) of Horbourg-Wihr on satellite imagery. Both historical elements feature their tinted realistic textures.

The application of this shade was automated using a *Python* script for *Blender*. This changes the shader's *Coat* parameter (Principled BSDF) to an entire collection. This parameter "simulates, for example, a varnish or a lacquer" (Blender, 2024). In particular, the parameters chosen by the user are *Tint* (RGB values) for the color to be applied, *Weight* (integer between 0 and 1) for color intensity, and *Roughness* (integer between 0 and 1).

5. Photorealistic renderings

Nowadays, everyone knows how to interpret and use a photo or a video. These supports are simple. Their distribution, transmission, and publishing are widespread and do not require significant computer resources. In addition, they represent an accessible and pleasant way to share. One of the central goals of this work is photorealistic rendering. It comes in several forms: photo rendering and video rendering. To stay in the same low-cost environment, these processes are carried out in *Blender* using the *Cycles* ray-tracing rendering engine.

5.1 Rendering settings for a good visual quality / computation time ratio

The difficulty associated with the rendering procedure is the optimization of the parameters. Indeed, users must play between visual quality and calculation time. First, the *resolution* of the renderings must be defined according to the use made in the end (viewing on a large screen, viewing in VR, printing on paper, etc.). The *sampling* parameter also plays a very important role in quality and computation time. This is because a *sample* is an attempt to trace rays from the camera in multiple directions. Each time the rays interact with an object in the scene, they collect information about the interaction of light with the scene. The higher the number of *samples*, the more detailed the rendering will be. A *sample* number that is too small can, for example, generate a noisy image (see example in Figure 8). This parameter works in conjunction with the *Time Limit*. Indeed, the latter attributes a limited time to the passage of the samples. We clearly see that for a smaller number of samples, the detail of the trees is reduced, and the noise of the fog is accentuated.

Figure 8: render with a *sample* parameter set to (a) 1 (for 7 seconds), (b) 1024 (for 7 seconds).

Another interesting parameter influencing the computation time is the *Light path*, i.e. the number of bounces of light rays on objects in the scene. This parameter can be modified on the Diffuse, Glossy, Volume, Transparent, etc. interactions. These parameters must be adapted according to the objects, and in particular, the types of materials present in the scene. For example, Figure 10 shows the same rendering of the Horbourg-Wihr vines for two *Transparent* parameters. These vines are modelled by textured planes with a diffuse map for color, and an alpha map for transparency (Figure 9). The alpha map is not sufficiently considered for a lower transparency coefficient (Figure 10).

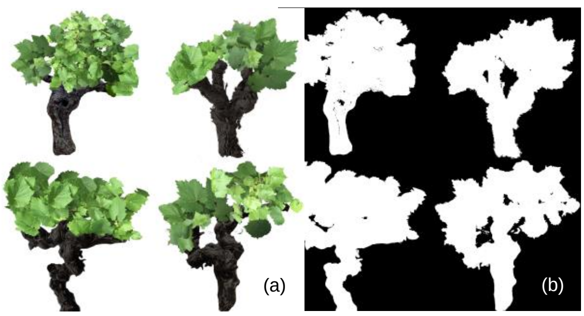


Figure 9: (a) Diffuse map; (b) Alpha map of the Horbourg-Wihr vineyards

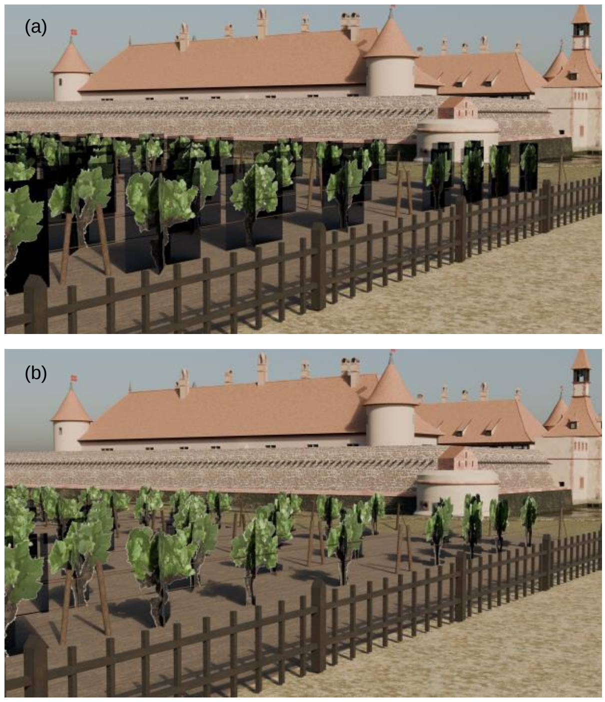


Figure 10: Proof of *Blender*'s Light path Transparent parameter: (a) Transparent coefficient equal to 1; (b) Transparent coefficient equal to 3. The alpha map is not sufficiently considered for a lower transparency coefficient.

Finally, to finish on the rendering parameters to optimize the quality and the computation time, we can state the *Performance tab* of *Blender*. This includes the *Memory* setting that allows you to change the number of tiles for a render. This is because rendering calculations are split into tiles. They are then performed tile by tile and cached as rendering. The higher the number of tiles (the more subdivided the rendering), the more the rendering performance is improved. We also find in the *Performance tab*, the *Persistent data* parameter, which once activated, allows you to remember the image rendering data (light interactions, etc.) and therefore to speed up future renderings.

5.2 Rendering

Renderings can be of various shapes. Optimizing rendering settings based on the type of rendering (static or dynamic) is a first step. The scene must be prepared, including the camera type and its path. Static rendering needs a simple placement of the camera in the desired view, dynamic rendering (video) needs to create the camera path. There are various methods for this: constraining the camera to a curve, placing Keyframes on the camera's transformation settings, enabling the *Auto Keyframe*, etc. The last step before starting the calculation is adjusting the output parameters of the renderings. Indeed, it is a question of defining the format and the recording path of the output image and the frames to be rendered. It should be noted that whether it is a photo or a video rendering, only images are rendered, alone or in sequences of images. This process makes it possible to control the output images, but also to avoid losing all the calculations in the event of a *Blender* crash. Indeed, image sequence rendering allows images to be saved as rendering because each frame is rendered one by one. This sequence of images is then recomposed with the right parameters, including fps (frame per second), bitrate, encoding, desired video format, and video resolution.

6. Transition to VR

To offer an immersive experience to the public, in this context, to go beyond the border of the screen, virtual and even augmented reality provides solutions. Modelling projects can thus be transposed into several VR media.

6.1 Rendering in the VR headset

The VR headset can be used in various cases. In the simplest case, it is possible to implement video and photo renderings made using a panoramic camera in *Blender*. These then offer a 360° guided immersion to the viewer (Figures 11 and 12).

Figure 11: Panoramic rendering of the north façade of the Oedenbourg castle (viewable with a VR device via the attached QR code).

For this project, MetaQuest 3 headsets with a storage capacity of 256 GB have been implemented. They allow the user to import 8K (8192 pixels x 4096 pixels) videos and images, panoramic or not. Note that the encoding of videos must respect certain parameters such as *codec*, *bitrate*, *resolution*, *format*, and number of *fps* (MetaQuest, 2024). For the project, the following parameters were used (Blender:Output, 2024):
Format: .mp4
Video Codec: H264
Container: MPEG-4
Output quality: Medium/Low

Figure 12: Viewing a panoramic rendering of Ramstein Castle within the Meta Quest 3 headset (with Wi-Fi replay of the headset content on a computer).

6.2 *Sketchfab* Platform

In a more interactive way, 3D scenes in VR are available through *Sketchfab*. This solution is simpler because the rendering engine is managed by the platform. Anyone with a VR headset and an internet connection can then walk around the scene. *Sketchfab* is a 3D visualization platform. It offers the possibility, on the one hand, to import, modify, annotate, and animate a 3D model, and on the other hand, to view, to move, and to upload models uploaded to the platform. *Sketchfab* is free to view and import for models weighing up to 100 MB. To upload models over 100 MB, the platform offers a subscription of up to 500 MB of import.
Textured or non-textured models can be uploaded. Note that if the models have been textured using procedural shaders in *Blender*, the textures must be baked and the shaders adapted for *Sketchfab* to interpret them. The platform is also available in normal viewing on computers and smartphones.

6.3 *Unity* Platform

As a third way of viewing 3D models using the VR headset, the use of game platforms like *Unity* or *Unreal*, *Faro Scene*, and even *Blender* (with the *VR Scene Inspection* add-on, for example) are interesting alternatives. However, regarding *Blender*, this type of visualization requires powerful computer hardware and, above all, a highly optimized scene (geometry and texture) for the VR and live visualization of a textured 3D model. Regarding *Unity*, it is possible to create interactive scenes offering a choice of several castles.
As part of the Châteaux Rhénans – Burgen am Oberrhein project, several interactive games are still under development.
Initially, a simple game offering the user the choice of the castle to browse using a home menu was implemented. This choice can be repeated once the game has started using a context menu that the user can open at any time. The viewable models here are the 3D models of castles in their current state.
Secondly, Wasenbourg castle was used to add interactions to the game. This game offers a 4D and guided immersion. Indeed, the user can first visualize different shapes of the castle: textured mesh of the current state, textured historical restitution, historical reconstruction evaluated by an archaeologist. In addition, there are built-in dialogue boxes, allowing the player to immerse themselves in the history of the site. This game was later enhanced with an interactive module, allowing the player to build/deconstruct a wall by manipulating bricks using the controllers.
Developments around the theme of games are being designed at INSA Strasbourg. These developments aim to increase the fun aspect of VR by implementing tools, functions, and mini-experiences accessible to as many people as possible.

7. 3D printing of castle models

Above, various ways of sharing 3D productions were presented. These latter methods all use digital means, however, for more inclusiveness, it is also interesting to have physical models on a reduced scale. This can therefore be done through 3D printing. These models are more accessible to the public and do not require any equipment or computer knowledge. In addition, they allow you to unlock another sense, touch, and therefore include people with visual impairments and disabilities. In addition, as part of the Châteaux Rhénans – Burgen am Oberrhein project, the aim of the German partners was to design tactile bronze models incorporating legend and Braille texts. To this end, INSA Strasbourg carried out additional processing following the modelling of the castles based on topographic data acquired in the field. The basis is therefore the cleaned and resampled mesh of the castles. These are then adapted for 3D printing (hole filling, poly-count optimization, smoothing, and visual enhancement) and then merged with the texts. Note that the final models' scale is known, and the Braille text size is adjusted to comply with the dimensional standards.

8. Conclusion

Using the example of different models, this paper developed different methods of rendering, adding natural or urban environments, making 360° images or videos that can be viewed in AR/VR headsets contributing to the analysis and enhancement of the models. The realization of these derivative products is described in the open-source *Blender* modeling environment and its add-ons. The Roman camp (4th century), the reconstruction of a Renaissance castle rebuilt on the same site, both of which have now disappeared, or the restitution of Rhine castles in their historical periods and the modeling in their current state are examples in a wide variety of situations that have made it possible to explore the multiple functionalities of this open-source 3D modeling platform.

Acknowledgements

Best thanks to the Châteaux Rhénans - Burgen am Oberrhein Interreg VI Project (2023-2025). Our work aligns with Action 4.6, focusing on 3D valorization of heritage sites. Best thanks to city of Horbourg-Wihr for proposing the modeling of disappeared heritage sites.

References

Alpha Trees: Render massive forest, fast in Blender. https://blendermarket.com/products/alpha-trees (accessed 10/2024)

Astuti, I.A., Purwanto, I.H., Hidayat, T., Satria, D.A., Haryoko, Purnama, R., 2022. Comparison of Time, Size and Quality of 3D Object Rendering Using Render Engine Eevee and Cycles in Blender. In: *5th Intern. Conf. of Computer and Informatics Engineering* (IC2IE), pages 54-59.

Blender: the free and open-source 3D creation suite. https://www.blender.org/ (accessed 08/2024)

BlenderGIS: Blender add-ons to make the bridge between Blender and geographic data. https://github.com/domlysz/BlenderGIS (accessed 08/2024)

Blender: Output. https://docs.blender.org/manual/fr/4.2/render/output/properties/output.html (accessed 09/2024)

Blosm: Blosm for Blender, OpenStreetMap, Google 3D cities, Terrain. https://github.com/vvoovv/blosm (accessed 08/2024)

Cadastre, 2024. http://cadastre.gouv.fr (accessed 10/2024)

Cartier, L., 2020. Modélisation 3D du château disparu des Wurtemberg à Horbourg-Wihr et exploitation de la réalité augmentée pour une mise en valeur dans la trame urbaine contemporaine. In: *Revue XYZ* n°162, 1er trim. 2020, pp. 63-68.

Corns, A., Davidson, L., Namberi, S., O'Reilly, R., Shaw, R. 2024. Developing enhanced and enriched 3D models of cultural heritage assets as part of the EU co-funded 5Dculture project. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-2/W4-2024. doi.org/10.5194/isprs-archives-XLVIII-2-W4-2024-157-2024

Fuchs M., 1996. Le castellum, une étude lacunaire. In: *Fuchs M. (dir.), Horbourg-Wihr. À la lumière de l'archéologie. Histoire et nouveauté. Mélanges offerts à Charles Bonnet, Actes 2 d'Archihw*, Horbourg-Wihr, 1996, pp. 135-146.

Fuchs M., 1997. Antiquités et érudits sous les Wurtemberg ou l'aube de l'archéologie horbourgeoise. In: *Maurer P., Kauffmann M. (dir), Les Wurtemberg dans la seigneurie de Horbourg, Actes 3 d'Archihw*, Horbourg-Wihr, 1997, pp. 131-134.

Google Maps platform, 2024. https://mapsplatform.google.com/resources/blog/commonly-asked-questions-about-our-recently-launched-photorealistic-3d-tiles/ (accessed, 11/2024)

Gscatter, 2024. https://store.gscatter.com (accessed 11/2024)

Interreg VI – Châteaux rhénans – Burgen am Oberrhein, 2023-2025. https://www.chateaux-rhenans.eu/fr/ (accessed 10/2024)

Koehl, M., Fuchs, M., Nivola, T., Koch, J., Cartier, L. and Soussoko, S. 2020. When roman antiquity and Renaissance came together in virtual 3D environment: 3D modelling considerations. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLIII-B4-2020, 607–614, doi.org/10.5194/isprs-archives-XLIII-B4-2020-607-2020

MapsModelsImporter: tools for importing 3D models from wide maps in 3D content softwares. https://github.com/eliemichel/MapsModelsImporter (accessed 08/2024)

MetaQuest: 360/180Video requirements. https://creator.oculus.com/media-studio/documentation/video-spec/ (accessed 10/2024)

Nivola, T., 2018. Modélisation 3D du Castellum de Horbourg-Wihr et exploitation de la réalité augmentée pour une mise en valeur dans la trame urbaine contemporaine, In: *Master Thesis INSA*, 89 pages.

Prochitecture, 2024. https://prochitecture.com/ (accessed 10/2024)

Ramtohul, A., Kumar Khedo, K., 2022. Augmented reality systems in the cultural heritage domains: A systematic review. In: *Digital Applications in Archaeology and Cultural Heritage*, Vol. 32, 2024, ISSN 2212-0548, doi.org/10.1016/j.daach.2024.e00317

Sapling Tree Gen: Adds parametric tree in Blender. https://extensions.blender.org/add-ons/sapling-tree-gen/ (accessed 10/2024).

Schneikert, F., 2016. Horbourg-Wihr (Haut-Rhin), Village - Schlossgarten - Rue des Écoles. Plusieurs états d'occupation et de nouvelles données sur le château des Wurtemberg, *rapport de diagnostic, Archéologie-Alsace*, SRA Alsace, Région Grand-Est, 2016, 87 p.

Sommer, E., Koehl, M. and Grussenmeyer, P., 2024. Crafting and modifying Rhine castle models with parametric modelling in Blender. Int. *Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-2/W4-2024, 405–412, doi.org/10.5194/isprs-archives-XLVIII-2-W4-2024-405-2024

Sommer, E., Koehl, M., Grussenmeyer, P., 2024. Parametric Modeling Techniques for Rhenish Castle Models in Blender. In: *Heritage 2024*, to be published. 26 pages.