

VR and Natural Heritage

Subjects: Computer Science, Information Systems

Contributor: Partarakis Nikolaos , Emmanouil Zidianakis , Zacharias Pervolarakis , , Antonios Katzourakis , Theodore Evdaimon , Xenophon Zabulis , Constantine Stephanidis

Caves can be considered as features of the landscape that have been formatted through a long period and are signs of the past climatological conditions, of prehistoric animal and human inhabitation, and provide habitat for species. The sustainability and protection of cave heritage have been recently discussed mainly because over-tourism has been proven to have negative side effects on the preservation of the heritage site. Technology today has provided widely adopted inexpensive technical means to support immersive visiting experiences to sites of heritage interest that could support their valorization and sustainability in forms more friendly to the site. An alternative visiting approach is explored through a use case applied to the Alistrati cave near Serres, Greece, where a VR tour guide can support immersive visiting experiences to the heritage site. By employing means of digital preservation of heritage sites the VR solution aspires to offer immersive close to reality engaging visiting experiences.

natural heritage

preservation

valorization

sustainability

cave

Alistrati

Serres

1. Introduction

In history, caves have a broad range of uses and meanings, formed over long periods that include prehistoric, historic, and contemporary phases. These uses have generated distinct values and meanings for diverse groups, including local communities, international researchers, and tourists. Caves are unique in that they fulfill, at least to some extent, all of the criteria for natural, tangible, intangible, and historical heritage protection, making management of them difficult. The past, present, and future importance of caves and their multiple uses and meanings have not been consistently considered in measures to protect or manage them. The increasing pressure from economic development practices, including logging, mining, and tourism, also compounds management and conservation [1].

From a preservation perspective, the protection of cave heritage is extremely important. Underground sites have become attractive tourist destinations for an increasing number of visitors. This flow of visitors has made sustainability a major issue; that is, how tourism development ensures economic benefits for host communities and respects local identity without compromising the environmental resources [2]. Underground sites have an intrinsic scientific (e.g., geological record), cultural (e.g., superstitious rituals, archaeological artifacts), and recreational value (e.g., exploring inner caves) [3], leveraged for “cave tourism” including, among other forms of tourism, geo-tourism, mining tourism, and adventure tourism [4][5].

2. Alistrati Cave Model Implementation

2.1. Laser Scanning

A fixed laser scanner with a scanning range of 70 m was used to scan areas within the Alistrati cave. By placing the scanner in multiple places in the space, a series of scans were performed, for its full coverage. The Alistrati cave consists of eight points of interest (stops), which were covered with 29 grayscale scans. Special spheres identified by the Faro Scene software were used for automated registration. These spheres are placed in the scene in a manner that maximizes the visibility of each one from the different scanning points, and then the software recognizes the common spheres between the scans and performs the automatic registration. The data were imported into Faro Scene software for the process [6]. The scans were registered in a point cloud for each point of interest and exported. Each point of interest was subdivided into nine point clouds and each point cloud was imported to Faro Scene software again to create the mesh. Mesh models with highly detailed structures were generated from the point clouds and exported.

Individual scans were acquired without color information for two main reasons (a) to reduce the time needed to cover a path of almost 150 m within the cave and (b) to dismiss color information due to the peculiar lighting conditions within the cave. An example can be seen in **Figure 1**.

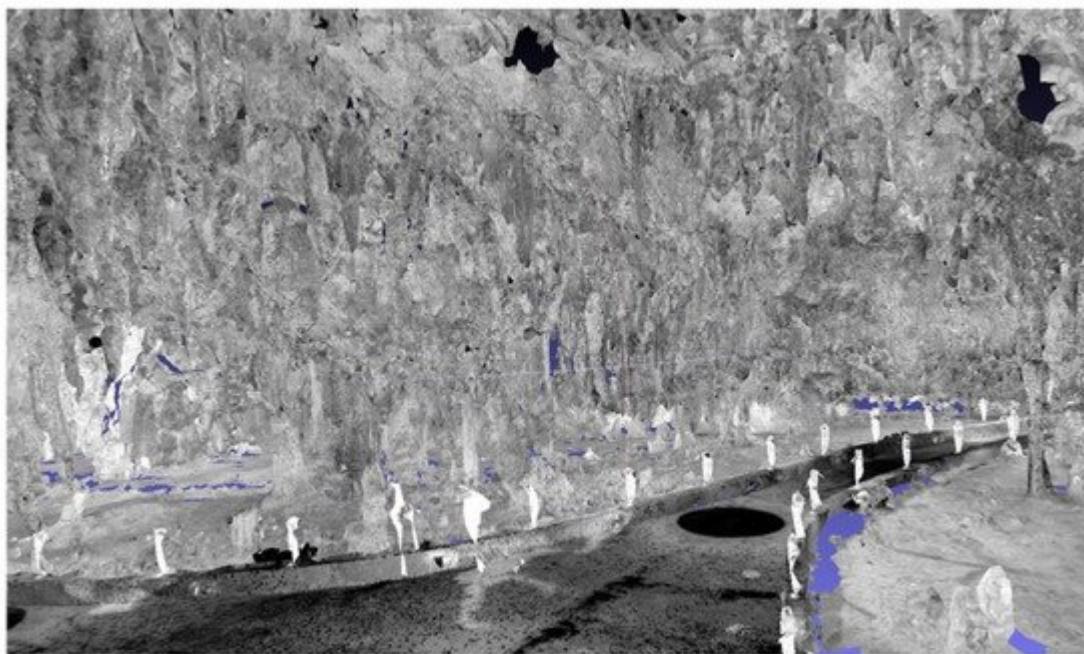


Figure 1. Mesh rendering of the combined point clouds.

2.2. Aerial Photogrammetry

For the aerial reconstruction of the exterior of the Alistrati cave the Phantom 4 Pro V2 was used, with which the collection of aerial photos and videos took place. The drone was equipped with a camera using a fixed 24 mm lens, capable of recording aerial photographs with a resolution of 20M pixels and the ability to record 4K video resolution

(also known as an ultra-high definition or UHD) in high-efficiency Video Codec (HEVC) H.264 and H.265. At the same time, three batteries were available, each of which offered a flying capacity of about half an hour. For the conduction of the aerial photography and video recording, specific orbits were created using the Pix4DCapture software, which is followed by the drone, and the appropriate filters are adjusted to the camera lens depending on the lighting of the day in which the flights were carried out. The flying height of the drone was calculated based on the morphology of the area and it was decided that an altitude of 40 m would be sufficient to capture a desired level of detail for the final photogrammetry model. More specifically, the “Double Grid” mission was selected in the Pix4DCapture, with a camera angle of 70 degrees, front overlap at 80%, and side overlap of 70%. This option is recommended by the software for point cloud reconstructions for surfaces with height fluctuations.

The data were post-processed using Pix4D Mapper, photogrammetry software [7]. Aerial photographs were uploaded to the photogrammetry software. Photographs were collected using Pix4D capture, which creates a fully autonomous flight and scan of the area of interest via a predetermined orbit. The reconstructed site can be seen in **Figure 2**.

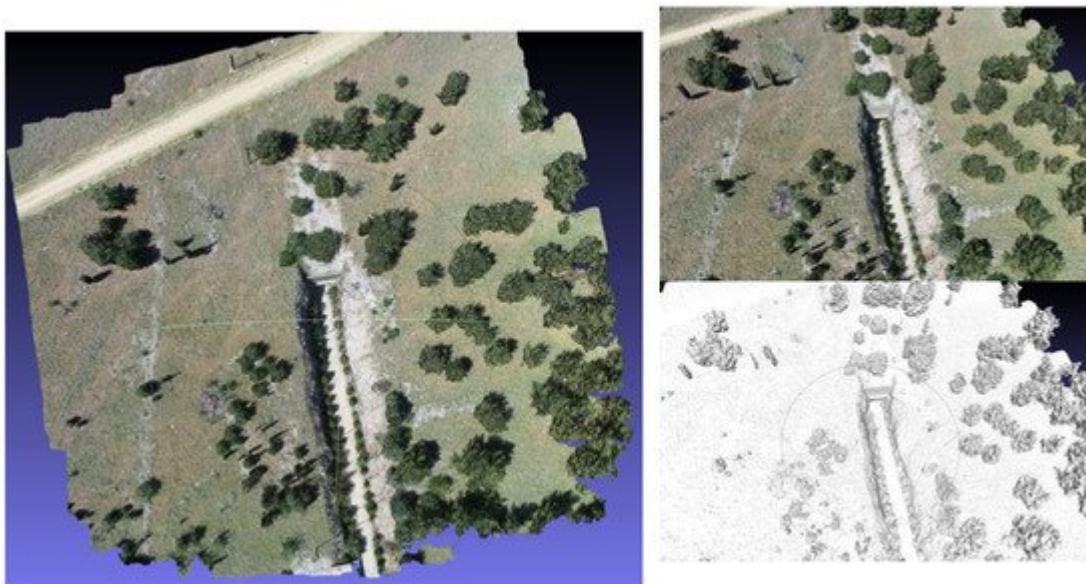


Figure 2. 3D reconstruction of the rural space and entrance of the Alistrati cave.

2.3. 3D Model Synthesis and Coloring

In this stage, the registration of the two models (interior–exterior) was made to create the synthetic environment to be employed by the VR tour guide. For the interior, extensive reworking was conducted to achieve (a) filling of the areas with gaps resulting from visual occlusions during laser scanning, and (b) coloring of the interior using reference photography from the heritage site.

For this, the process followed an extensive artistic-driven approach for filling the missing parts using the 3D application, Blender [8]. Due to the chaotic nature of the stalactite and stalagmite cave, an abstract low-polygonal approach was used for the newly created meshes. For the best feature preservation with satisfactory performance

in a portable VR workload, half a million polygons were set for each point of interest (POI) scene on average. The main target mesh was generated by the point cloud at 350,000 faces using FARO Studio, filling as much surface area as possible. All unnecessary polygons, such as columns, ropes, etc., were deleted and were replaced by repeating a better version of them. The target mesh was further enhanced by manually modeling the blank parts to an equivalent topological structure of the neighboring LIDAR set. Modeling started by extracting the boundary loop edges; thus achieving polygon continuity. Every POI consists of overlapping scans of 10–15 million faces and all available features, such as albedo, normals, and coverage area, were extracted to the target mesh as multiple 8K textures, with a Ray Distance of 5 cm for the albedo and 2 cm for the normals to minimize negative angles from the polygonal gaps. To retain the best texture quality, new UVs were created by manually marking seams in areas near the main path the user is experiencing, combined with SmartUV at 30 degrees for the chaotic structure in the distant areas, with a 1/16,384 island margin. For every scan extraction, the bake output margin was set to 3 pixels, as it retains textures on acute-angled triangles while also avoiding texture bleeding to the non-overlapping parts.

A shader node tree was used to combine and enhance all of the textures. Multiple mix-color nodes set to lighten were used for the albedo, brightening the dark areas by contributing light from all of the scans, while the alpha channel was used to mask the secondary cloned-neighboring textures for the reimagined missing parts for both albedo and normals. Masks were used to separate the cave from the concrete pathway and a color filter with a shade of brown that uniformly shifted the hue to green and blue was applied, to resemble reference images of the decomposed granite and stalagmite synthesis of the cave. Finally, colored lights that are also present in real life were added to give an extra vibrancy to the image. The entire process is summarized in **Figure 3**.

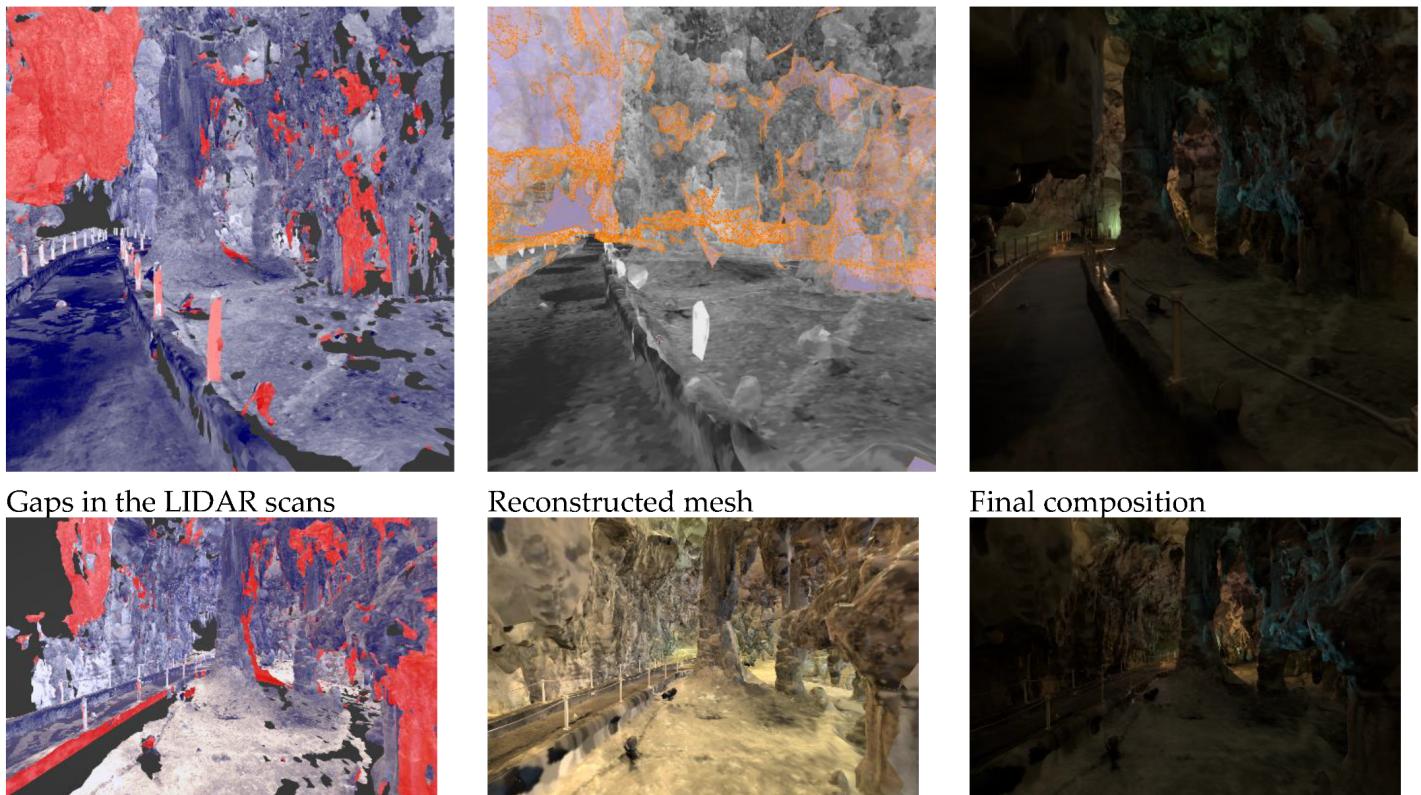


Figure 3. 3D model synthesis and coloring process.

2.4. Requirements for the VR Visit Scenario Definition

Before the implementation of the VR application for the Alistrati cave, specific system requirements had to be decided upon. To begin with, the VR tour should match the real-life one in terms of information; the means of providing the information about the cave could potentially vary. Ideally, researchers would use the 3D model in their app, but this proved to be a near-impossible task. Reconstruction techniques from point cloud data produce 3D models with highly unoptimized topology and complex UV maps that are simply unfit for VR purposes. Even on high-end computers simply loading the entirety of the cave in computer-aided design (CAD) software proved to be a rather cumbersome process. To make things worse, it was decided that the app should be able to run smoothly on stand-alone VR devices to further increase the potential target audience that cannot afford expensive desktop VR gear. Another requirement was to provide multi-language support for the VR tour and, last but not least, immerse the user in the experience with high fidelity visuals and environmental sounds.

3. The VR Tour Guide

For the implementation of the VR tour guide app, the Unity game engine was used [\[9\]](#). Unity is an all-purpose engine that provides fast-developing iterations with the high-fidelity 3D rendering of graphics. Specifically, the universal rendering pipeline (URP) was used that comes along with a node-based shader graph for creating custom shaders without the need for direct scripting for graphics processing units (GPU). In addition, Unity's own VR library XR SDK was the main library used in this project. By using XR SDK, the developing cycle of the VR app was greatly accelerated and vast portability between different HMDs was easily achieved through Unity's multi-platform support. While developing the application, the Oculus (Meta) Quest 2 was the main testing unit, which today is the industry's standard and most affordable stand-alone VR headset.

3.1. Digital Cave Game Level Optimization

VR development is a notorious field when it comes to performance optimizations since every frame has to be rendered twice in the system; specifically, one frame for each eye of the user to simulate spatial depth. The cost of rendering everything twice is well paid in performance, meaning that researchers have to compromise with lower fidelity graphics and optimize as much as possible. When researchers allow for low frame rates or frame drops in VR applications, it may cause the user to feel nauseous and ultimately create negative feelings both towards the tour of the cave and the experience of VR as a whole.

3.1.1. Handling the Vast Amount of Data

The result of the reconstruction technique outputs a highly unoptimized 3D model, unfit for a stand-alone VR application. It would be possible for a 3D artist to recreate the entirety of the cave by hand, using the photo-scanned model as a base. By the end of this recreation, a final optimized model would be exported that contains far fewer, but more well-defined, vertices. This step, requiring a 3D artist to recreate a model with clean geometry, is a common technique that is used in game development when creating virtual characters that were initially

conceptualized using sculpting as a 3D modeling technique. However, re-modeling the complete set of the Alistrati cave data would be intense manual work and it was decided that it would not be worth the effort for this project. Researchers' solution to the problem was to divide and conquer; split the cave into multiple parts and each part would represent a game level on researchers' VR application. To traverse through each cave part the user will enter magical portals that will be positioned at the start and end of each part of the cave, respectively. This way, the information that needs to be handled in each scene is greatly reduced. Specifically, each part of the final application contains one or two points of interest where the virtual tour assistant will stop and provide information to the user. Then, the assistant will move towards the portal and wait until the user approaches before entering.

3.1.2. Baked Lighting

When rendering a frame in a computer graphics application, a lot of complex calculations take place behind the scenes. From applying the transformation matrix to each object (move, rotate, scale), to calculating how the light would realistically react to illuminate a scene. Naturally, the more complex the illumination model in question, the more realistic the final result will look. Thankfully, there is an exception to this rule that allows researchers to have realistic results with zero cost; if an object and a light source are static, meaning that they won't be moved for the rest of their lifetime, the lighting result will be the same for the entire duration of the application. So, researchers can avoid rerunning those calculations on each frame by simply calculating it once before the app runs. This technique is called baked lighting and it is a common technique that dramatically increases performance on real-time computer graphics applications.

On their implementation, researchers initially tried to use Unity's integrated baking pipeline, but it did not seem to be working well with the geometry of their cave 3D model, which was mostly created by automated point-cloud triangulation. Researchers' next solution was to use Blender to bake the lighting. The process involved placing the light sources by hand in the corresponding places that matched the photographic material researchers had available from the actual cave facility. By switching to the rendering viewport of Blender, researchers could see in semi-real time the results of the lighting placement. Finally, researchers baked the direct, indirect, and color passes of the diffuse light type into a new texture of 8192 by 8192 pixels. By repeating this process for each separated part of the cave, researchers ended up with image textures that researchers can use with unlit type shaders that use zero light calculations. You can see the results of the baking technique in **Figure 4**. It is important to mention that since there are no light sources in the scene, it is not possible to make use of normal maps to further increase the visual fidelity of the final result. There are specialized ways to apply normal maps even to such cases, but those options were not explored in this work.



Figure 4. Baked lighting result of the cave in Unity 3D.

3.1.3. Navigating the Cave and Controls

It is an industry standard that at least two ways of movement should be provided to the user in a VR environment: (a) joystick walk and (b) point and teleport. In the Alistrati cave application, the user has the option to move as if he was walking by pushing the left-hand joystick in the desired direction; by clicking the right-hand joystick he can change the direction he is looking at by 30 degrees; and lastly, by pressing the trigger of the right hand he can target a position on the floor and teleport there instantaneously. The reasoning behind offering both ways of movement lies in the fact that some people tend to feel nauseous when using only the walk method. The above happens because the user perceives that he is walking in the virtual environment, but his body knows that, in reality, he is not moving. Finally, since the user interfaces in the application are simple and only extend to allowing the user to choose his language, by either pressing the primary or the secondary buttons on his controllers the user can interact with the UI elements.

3.2. The Virtual Tour Assistant

To assist the user in the VR experience, a 3D model of a robot was created. This robot assistant was modeled to be a small, flying, and friendly robot to appeal to the user (final result shown in **Figure 5**). The job of the assistant is to work as a guide and escort the user into the cave while narrating the history and importance of specific points of interest.



Figure 5. The robot assistant 3D model.

3.2.1. Assistant's States

The robot assistant executes a simple logic loop while the game is running. First, it moves towards the designated position in the cave. Then, the robot waits until the user is close enough and when he arrives the robot starts playing the sound associated with the specific point of interest. After the sound is finished, the robot will repeat the execution of the above steps by moving toward the next stop. To handle the movement of the robot Unity's NavMesh system was used. To use this system a walkable path had to be designated for the AI agent to understand how it can potentially move around the space. For the walkable path, cutouts of the original paths of the cave mesh were used, and after baking those meshes into the NavMesh system, the robot assistant was free to move independently.

To assign the positions where the robot should stop moving, wait for the user, and then play a specific audio file, an empty game object called the RobotPath was created. Inside the RobotPath exist multiple empty game objects positioned at places in the cave where the robot should stop moving and move to its next state. Another empty game object can be found inside RobotPath called PathSounds and it contains a public list with the sound files for every stop in every language available by the system. When the robot enters the moving phase of its state machine, it will use the NavMesh system to move to the position of the next RobotPath child. When it arrives and stops moving, it will play the sound from the PathSounds list that matches the index of the current stop. The advantage with the above setup is that a level designer can easily place as many stops and sounds as he pleases using Unity's editor while being completely agnostic of the code.

3.2.2. Animations

Up until this point, the robot assistant can execute its main function of roaming around the space and providing information to the user regarding the cave, but it lacks interactivity and appealing aesthetics, so simple animations were created to add to the overall experience. For starters, since the robot is levitating in the air, a discreet constant up and down motion was added. Then, when the robot is moving it bends forward, just like a helicopter would, to visually suggest that its thrusters are propelling it forward. In addition, custom shaders were used in the visor area of the robot to make it feel livelier. To create animated shaders, researchers used Unity's URP shader node graph and two shaders were implemented for the visor; one for an idle state in which the eyes of the robot would blink periodically, and another one where the eyes would transform into a sound wave visualizer while the robot would be speaking. Furthermore, a particle system has been created that emits particles in the form of sound waves to further emphasize the current status of the robot assistant, as seen in **Figure 6**. The advantage of using the URP rendering system in Unity is that it allows you to use the Shader Graph and build shaders visually. As Unity describes it: "Instead of writing code, you create and connect nodes in a graph framework. Shader Graph gives instant feedback that reflects your changes, and it's simple enough for users who are new to shader creation".



Figure 6. Robot assistant talking.

3.3. Sound Design

An important ingredient to further engage the user in a virtual world, are sounds. First of all, environmental sounds were added to the three-dimensional space of the scene; for example, the sound of water droplets dropping from

stalagmites with a light echo on the sound to emulate spatial depth. In addition, an ambient sound that imitates a light breeze blowing in the interior of the cave was attached. Lastly, a close-lipped booming sound was added to fake the robot assistant's thrusters that allow it to fly.

References

1. Roberts, N. The cultural and natural heritage of caves in the Lao PDR: Prospects and challenges related to their use, management and conservation. *J. Lao Stud.* 2015, 5, 113–139.
2. Buonincontri, P.; Micera, R.; Murillo-Romero, M.; Pianese, T. Where Does Sustainability Stand in Underground Tourism? A Literature Review. *Sustainability* 2021, 13, 12745.
3. Shavanddasht, M.; Karubi, M.; Sadry, B.N. An examination of the relationship between cave tourism motivations and satisfaction: The case of Alisadr cave. *Iran. Geoj. Tour. Geosites* 2017, 20, 165–176.
4. Parga-Dans, E.; Gonzalez, P.A.; Enriquez, R.O. The social value of heritage: Balancing the promotion-preservation relationship in the Altamira World Heritage Site. Spain. *J. Destin. Mark. Manag.* 2020, 18, 100499.
5. Torabi Farsani, N.; Reza Bahadori, S.; Abolghasem Mirzaei, S. An Introduction to Mining Tourism Route in Yazd Province. *Geoconserv. Res.* 2020, 3, 33–39.
6. FARO SCENE Software—For Intuitive, Efficient 3D Point Cloud Capture, Processing and Registration. Available online: <https://www.faro.com/> (accessed on 4 March 2022).
7. Pix4D SA—A Unique Photogrammetry Software Suite for Mobile and Drone Mapping. EPFL Innovation Park, Building F, 1015 Lausanne, Switzerland. Available online: <https://www.pix4d.com/> (accessed on 3 March 2022).
8. Community BO. Blender—A 3D Modelling and Rendering Package; Stichting Blender Foundation: Amsterdam, The Netherlands, 2018; Available online: <http://www.blender.org> (accessed on 2 March 2022).
9. Unity—A Real-Time Development Platform. Available online: <https://www.unity.com> (accessed on 8 March 2022).

Retrieved from <https://encyclopedia.pub/entry/history/show/55085>