

Preflight Summary Report for: SBern_DOI 9401850.pdf

Profile: Convert to PDF/A-1a (sRGB) (Processed pages 1 to 6)

Processed by Hoyett, Date: 6/28/2021 2:57 PM

Fixups

-  Convert to PDF/A-1a (1 object)
-  Make document XMP Metadata compliant with PDF/A-1 (1 object)
-  Flatten transparency (high resolution) (1 object)
-  Remove all alternate images (1 object)
-  Remove document structure compression (1 object)
-  Recompress LZW as ZIP (1 object)
-  Convert color to sRGB (10806 objects)
-  Embed fonts (1 object)

Results (Summary)

Error

-  CIDset in subset font missing (1 match on 1 page)
-  Glyphs missing in embedded font (1 match on 1 page)
-  MarkInfo missing
-  Structured PDF: Structure tree root entry missing
-  Text cannot be mapped to Unicode (1 match on 1 page)
-  Width information for glyphs is inconsistent (1 match on 1 page)
-  XMP property neither predefined nor defined in extension schema

Document information

File name: "SBern_DOI 9401850.pdf"
Path: "E:\FAMU_ADVANCE\ADV Teams\Fellows Program\Cohort 2\Accomplishments"
PDF version number: "1.7"
File size (MB): 1.4
Creator: "Appligent StampPDF Batch 6.5"
Producer: "pdfTeX-1.40.21; modified using iText® 5.5.6 ©2000-2015 iText Group NV (AGPL-version)"
Created: "1/22/2021 8:04 PM"
Modified: "6/28/2021 2:57 PM"
Trapping: "Unknown"
Additional actions
 Page open action
 Type: Go to page in current document
Number of plates: 4
Names of plates: "(Cyan) (Magenta) (Yellow) (Black) "

Environment

Preflight, 9.0.0 (057)
Acrobat version: 9.00
Operating system: Microsoft Windows (Build 9200)

Overview of Visualizing Historical Architectural Knowledge through Virtual Reality

1st Anahid Ehtemami

*Electrical and computer engineering
FAMU-FSU College of Engineering
Tallahassee, FL
aehtemami@fsu.edu*

2nd Sang Bum Park, PhD.

*Architecture and Engineering Technology
Florida A and M University
Tallahassee, FL
e-mail: sang.park@famu.edu*

3rd Shonda Bernadin, PhD.

*Electrical and computer engineering
FAMU-FSU College of Engineering
Tallahassee, FL
bernadin@eng.famu.fsu.edu*

4th Laurent Lescop Prof. STA Dr.hab arch.dplg

*CRENAU/ AAU UMR-CNRS1563
ENSA Nantes
Nantes, France
laurent.lescop@nantes.archi.fr*

5th Andrew Chin, PhD.

*Architecture and Engineering Technology
Florida A and M University
Tallahassee, FL
andrew.chin@famu.edu*

Abstract—Virtual Reality (VR) has become one of the emerging technologies over the past decade for improving the quality of life in human experiences. It has exciting and popular applications in entertainment, sports, education, and even digital documentation of notable or historical sites, allowing users to immerse themselves in an alternate reality. By combining the principles of software development and immersive VR, real-life VR experiences seek to transport users to an interactive environment where they can view, observe, and experience historical events and artifacts in a new way. There are several steps involved in VR development of cultural and historical sites that require a solid understanding for adaptable and scalable design. This paper is a review of the VR development process for notable historic preservation VR projects. This process can be used to create immersive VR experiences for other cultural sites.

Index Terms—Virtual reality, architecture, digital documentation, Immersive

I. INTRODUCTION

Over the past few decades, Virtual reality (VR) technologies have grown significantly and have been constantly evolving. Aside from VR games, it provided an interesting, attractive, and productive means to educate a wide range of audience about a variety of topics. Virtual reality can provide the opportunity for users to experience what might not be possible otherwise, such as exploring the high-risk vocational training, deep-sea levels, anatomy of human body, and cultural heritage projects that no longer exist [1]–[3]. “Cultural Heritage” refers to properties and sites with archaeological, aesthetic, and historical value (Tan, 2009). Diverse media and different digital tools help us recreate these areas in a virtual world and this experience is referred to as “Virtual Heritage” [4]. A virtual heritage environment enhances the knowledge and understanding of heritage and social history and the connection

between cultural sites and the wider public by providing a greater sense of engagement with social-historical memories, in addition to preserving the structure throughout time. Architectural heritage is now exposed to many dangers due to natural disasters or human invasive actions. On April 15, 2019, a fire broke under the roof of Notre-Dame de Paris cathedral, a historic French Gothic structure built in 1163, as described below. Most of the wood/metal roof and the iconic spire of the cathedral was destroyed in the fire. The estimated cost of reconstructions and restorations could surpass 1 billion dollars [5]. The Past project in partnership with Nottingham’s museums and cultural heritage institutions, is a narrative-based VR project that offers the mentioned advantages. In 2012, Hrozek discussed the processes of creation of VR experience of the historical building of The State Theatre of Kosice, Slovakia. Similar projects will be discussed later in this text. After the destruction of Bamiyan statues in Afghanistan in 2001, Ben Kacyra, the founding director of CyArk, a nonprofit engaged in 3D digital preservation of cultural heritage sites worldwide, began capturing historical landmarks using laser-scanning technology. Partnering with Google Arts and Culture was the start of launching the Open Heritage project, a new initiative dedicated to digital preservation of landmarks such as the 1000 year old Temple of Kukulkan in Chichen Itza, parts of the Roman city of Pompeii, and the Native American cliff dwellings at Mesa Verde in southern Colorado.

A. Related Work

The American Library Association describes digital preservation (DP) as “Combining policies, strategies, and actions that ensure access to reformatted and born digital content over time”, providing guidelines on content creation, content integrity, and maintenance for DP projects. VR technology,

one of the tools used for DP, facilitates the creation and visualization of the structure of historical buildings. 3D scanners are often used for accurately capturing the real-world picture that can be used for modeling and visualization. Finally, software development principles help turn the models into the interactive VR experience.

1) *The Past Project*: Nottingham's cultural heritage and social history was captured in The Past Project as a virtual heritage environment. The developers provided a space that the audience could inhabit, rather than just watching a performance. Visitors move around and interact with different elements in the space. They also encounter several embedded video and audio narratives that are designed to provide an immersive experience. The narratives are triggered as the visitors approach an object. The user experience was evaluated on different occasions where they either purely observed or observed and interacted with the VR environment. This evaluation showed that interactivity dramatically increased the positive response to the immersive experience [6].

2) *The State Theatre of Košice*: The historical building of The State Theatre of Košice was recreated in VR. For this project, the tasks are divided into 4 major steps: Preparation, Modelling, Verification, and Visualization. 3D scanning was used to improve the modelling and visualization.

3) *Notre Dame*: The historic Notre Dame Cathedral in Paris suffered a devastating fire in 2019. It permanently destroyed parts of the building. However, the site was recently immortalized in 3D by TARGO, a VR studio specializing in immersive documentaries.

II. METHODS

In this section, the main components of developing a VR historic preservation project are discussed.

A. Conceptual Model

A VR or Augmented Reality (AR) project includes several modules that interact at different levels. In figure 2, a project workflow is shown that is established by ChoroPhronesis, a research unit within Penn State's Department of Geography. This model divides the workflow into 3 main processes: Data Collection, Development, and Production. Many VR projects follow a similar structure

B. Program structure

For the development of the Nottingham museum project, Tuck designed a simplified program structure for the interactive VR projects as follows:

- Program Start: User navigation User input to control the navigation (mouse, keyboard, VR controllers, VR motion sensor)
- Event Controller: Player coordinates and event triggers (from user input through controller/raycast and colliders, promiximity)
- Post Processing: 3D Scan rendering

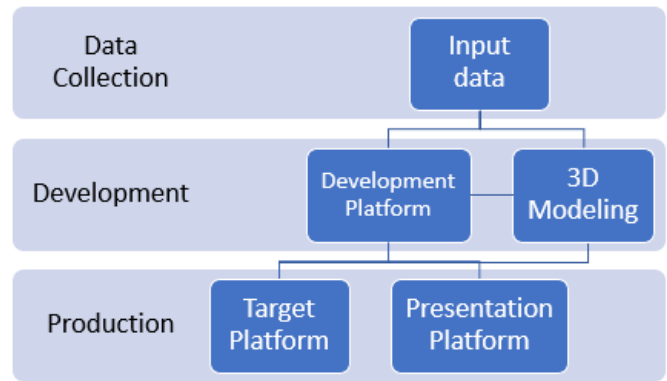


Fig. 1. General 3D/VR application building workflow [7]

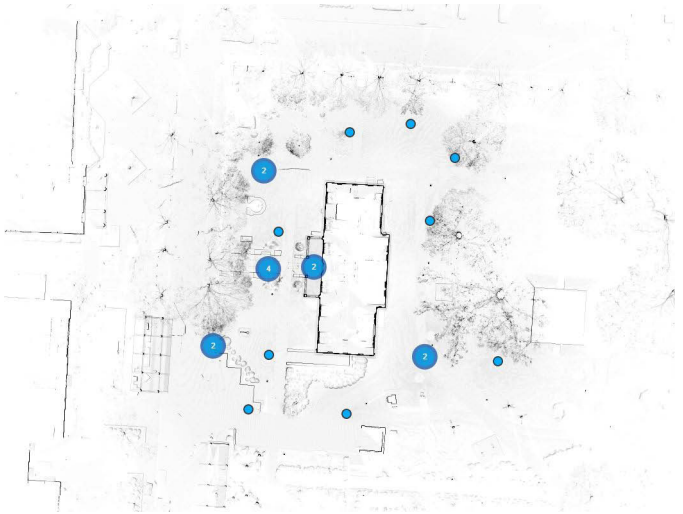


Fig. 2. Overview of the scanning map for FAMU campus scanning. The blue dots indicate the scanner positions.

C. 3D Scanning

It is important to find the right positions for the 3D scanner to require the least number of scans while covering the entire intended site. For a rectangular ground plan, the optimal number of scans is 8; one at each corner, and one in the middle of each edge. If necessary, it goes up to double for each position with two different heights in order to cover the low and/or top part of the site. In the production of The State Theatre of Košice project, the scanners were positioned 1.8m away from the side of the building. The steps for obtaining the scans:

- 1) Placing markers for scan aligning
- 2) Setting angles for scanning
- 3) Creating an image of the object, which later will be used as texture
- 4) Setting the scan resolution
- 5) 3D scanning
- 6) acquisition of position markers

Scanning time at each of the 8 locations took between 4-10 minutes, while it is dependent on how much details is scanned.

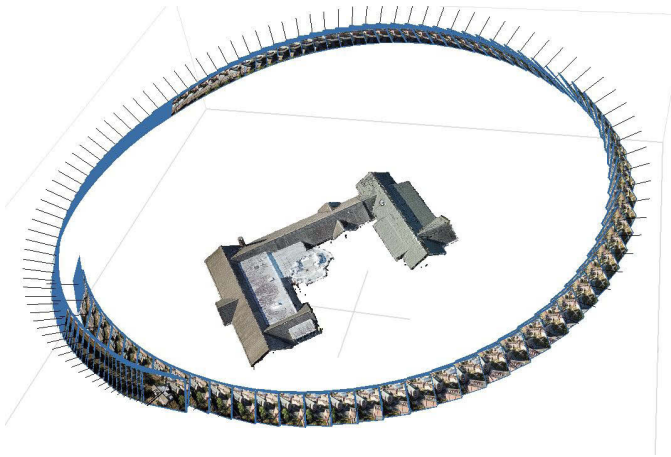


Fig. 3. Drone data obtained on FAMU campus.

After that, some processing is done for aligning point clouds and mesh creation. The mesh is then simplified and data that does not belong to the intended building is removed from the mesh. For visualization, in this project different methods have been used. 3D displays such as passive stereoscopic systems are able to render scenes up to 5 million polygons. Multiple projectors and special glasses and screens are required to view the content this way. Autostereoscopic 3D displays 3D images using the 2D-plus-depth method. Also, the 3D models can be printed using 3D printers by laying down successive layers of material.

D. Raw Data

Data that is used in reconstruction of sites can be found in different types. In addition to 3D laser-scanning and photogrammetry [8], [9], data from Google Maps and older 2D images or videos, and on-site surveys can be used to improve the accuracy of the model. The two most common types of 3D scanning are:

- **LiDAR:** This method is mainly used in architecture and construction. A LiDAR scanner uses a laser moving horizontally and a mirror rotation around a vertical axis so the laser can cover a large area. The newest iPad's LiDAR uses infrared light with depth-sensing system is also an example of LiDAR. [10] This method also requires high processing power.
- **Photogrammetry:** This method produces the 3D models by using multiple pictures of an object. Software combines the images and creates the 3D model. Photogrammetry method works best when conducted during a bright but cloudy day to avoid shadows if the goal is to scan all sides of a building. A drone can be used to photograph the higher parts of the building. After capturing the images, the data is imported into a software. The software then can align them to extract 3D information out of the photographs and generate a point cloud and later dense point cloud from the point cloud data and the image data. This step can be repeated until the dense point



Fig. 4. Meshes reconstructed from data obtained on FAMU campus.

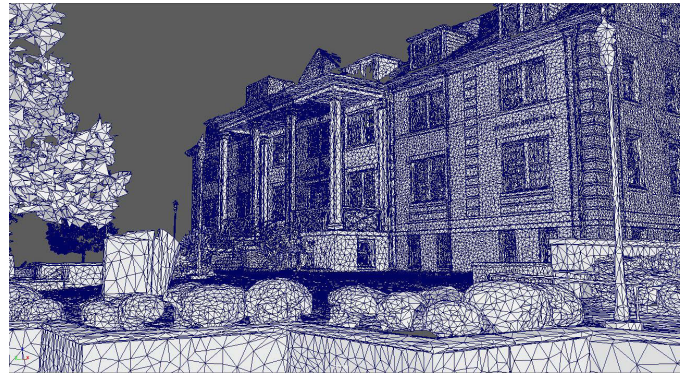


Fig. 5. Wireframe shading of a reconstructed building on FAMU campus.

cloud is satisfactory. There are tools that can be used for modifying the point cloud such as scaling, rotation, and translating. Next, a Mesh can be extracted from the point cloud or dense point cloud data. The textured mesh can be exported using the desired format for the intended use. For example, for applications developed in Unity platform, *.obj files, *.fbx, or *.prefab can be used to import the textured meshes [11].

1) **LiDAR vs. Photogrammetry :** The two aforementioned 3D model acquisition methods produce very different point densities. Also, resampling is sometimes done to reduce the size of the dataset. To compare the accuracy of different models, point-to-point, cloud-to-cloud, and cloud-to-mesh methods can be used [12].

2) **Point-to-point Distance:** In this method, a starting point and end point are selected to be the closest and farthest point to the origin of the coordinate system. The distance is computed between similar points and the maximum distance is considered. The number of points in the two datasets should be the same for this method.

3) **Cloud-to-cloud:** All points in the two sets are compared in cloud-to-cloud method. The computation time for this method can be very long if the datasets are large. Therefore, homologous octrees can be used to quickly calculate the Euclidean distances in a specific cube. An octree is a data structure where each node has eight children. Octrees are used to partition 3D spaces by recursive subdivision [13].

4) *Cloud-to-mesh*: A Delaunay triangulation on a set of discrete points is a triangulation $DT(P)$ that no point in the set is inside the circumcircle of any triangle in $DT(P)$. This method maximizes the minimum angle of all of the angles of the triangles in the triangulation [14]. In this method, Euclidean distances are calculated between points and meshes. The triangular meshes are computed from the point cloud generated by LiDAR scanners. To improve the accuracy, the quadratic mean of all Euclidean distances is considered. This method also can be time consuming based on the size of the datasets.

E. Data Transformation

After data collection, the next step is to use the data to create structured point clouds, meshes, and textures.

1) *Point Clouds*: Datasets that represent the external surface of objects or a space in the form of X, Y, and Z geometric coordinates in computer graphics are called point cloud [15].

2) *Meshes and Textures*: A polygon mesh in computer graphics is a collection of polygons that define the external surface of an object. The use of polygon meshes removes the risk of error-prone conversions such as CAD surfaces to mesh-based data from numerical simulations [16]. Higher number of polygons allows more precision in the surface representation but also increases the size of the dataset. When the point cloud is available, it is converted into a structured data in the form of a polygonal mesh to produce the best representation of the object or site [17]. 2D textures are images that are applied to 3D objects to show the colors and shades. Textures are usually images obtained from real world data that can sometimes be enhanced using image editing tools.



Fig. 6. 3D point cloud data from laser scanning done on FAMU campus.

F. 3D Modeling

3D modeling is usually used for terrestrial applications and in case of aerial domain, the term that is more frequently used is mapping [18].

Generating 3D models for the purpose of documentations and conservation requires the following properties:

- Accuracy, precision and reliability
- Portability for terrestrial acquisition
- Low cost

- Fast acquisition
- Flexibility due to the great variety and dimensions of sites and objects [19]

Often, one single technique cannot satisfy all of the requirements for a large and complex project. Therefore multiple sensors and techniques are usually used simultaneously to achieve a better model. Data fusion, or the integration of the data from different techniques can be done at sensor level or in the processing phase after data collection. Mobile mapping is an example of sensor integration and cost-effective acquisition of geo-referenced spatial data, combined with the usage of digital imaging devices, laser scanners, and Global Navigation Satellite System/Inertial Measurement Unit (GNSS/IMU) positioning sensors [20]. Challenges:

- Identifying the best techniques for capturing data and processing
- Selecting an appropriate workflow that results in high quality production
- Increasing the efficiency while maintaining the accuracy of the model
- Displaying the 3D model effectively using the desired method (VR headset, 3D printing, etc.)

G. Game Engine

Many platforms offer the ability to combine the visuals, physics, and interactivity. These platforms or game engines are development environments and consist of three smaller engines: Math engine, Rendering engine, and Physics engine. Some game engines like Unity and ApertusVR support 3D graphics that can be used for developing VR projects. As of now, Unity 3D has been rated the most high-performance VR game engine by G2 in 2020. G2 is the largest tech marketplace and uses an unbiased scoring methodology to rate products [21]. Performance and efficiency in VR development are key attributes, since VR usually includes large volumes of graphical data, and rendering modules require high computational power provided by Central Processing Units (CPU) or Graphics Processing Unit (GPU). Even if a module is not rendering, it could still represent a noticeable portion of the CPU consumption. Choosing the correct game engine according to the requirements of a project is the key, since it runs the different modules by prioritizing and multi-threading. Games are made up of different entities. The game engine holds each entity in a container such as scenegraph. Scenegraph is a generic tree that provides a fast way to traverse game entities [22]. A master loop (game loop) continuously renders and updates the coordinate space of each entity. If the game requires external forces such as gravity, physics engine is added. Gravity is an Ordinary Differential Equation (ODE) that will affect game entities. Motion equation integration with other attributes that affect the entities can be done through either of Euler, Verlet, Heun's, or Runge-kutta methods. The last two are the most complicated but exact method of integration [23].

Another well established game engine is Unreal developed by Epic Games. Unreal is free to download and the code for

TABLE I
POPULAR GAME ENGINES AND FEATURES.

Game Engine	Languages	Dev OS	Price
Unity	C#, JavaScript	macOS, Linux, Windows	Free
Unreal	C++, Blueprint	macOS, Linux, Windows	Free
CryEngine	C++, Flash, ActionScript, Lua	Windows	Subscription
Frostbite	C++	Windows	Proprietary

it exists on Github for cloning. For years, there has been a debate between artists and developers whether Unreal is more suitable for a specific project or Unity. To choose the game engine, there are several factors to keep in mind such as the levels of visuals, target platform, and team size. Both game engines offer similar contents, capabilities, and functionalities that are differently packaged [24].

1) *Unity 3D*: Here is a short summary of Unity features:
Pros

- Unity3D is free (but not open-source) and supports open-source technology.
- There is plethora of documentation available which can help with initial learning and troubleshooting throughout the project.
- It is free as long as the developer team earns less than \$100k for a project.
- It supports collaborations, Internet of Things (IoT), and distributed processing.
- Projects targeted for different platforms can be developed on it since Unity3D is platform independent [25].
- It can work with third-party Integrated Development Environments (IDEs)

Cons

- Memory garbage collection is automatic and can't be modified manually.
- Draw Call Batching is automatic. It also does not include Skinned Renderers (characters).
- For dynamic batching, meshes need to have less than 900 vertices.

H. Interactivity

Human-computer interface is one of the most important aspects of designing a system that users will be regularly interacting with. The goal of interaction design is to reduce the negative feedback that interacting with the system can cause for users [26]. In designing the interactive interface, these are the things to keep in mind:

- Easy to use: The best design is the one that users can interact with, with the least amount of "learning" time or frustration. The system should not cause confusion and be easy to break by a novice user.
- Effectiveness of use: users must be able to perform the task they need through the user interface (UI).

- Providing a pleasant experience: in addition to learnability and effectiveness, a good UI will provide user with a positive experience while interacting with the system. This is an even more important issue when designing an interface for an educational/entertainment project such as VR preservation of historical sites.

By controlling the path that the visitor can take in the VR, the narrative can progress steadily. It also can limit or neutralize the consequences of decisions that the user has made [27].

Common 3 Dimensional User Interface (3DUI) features include: head tracked view rendering, full body interaction, two-handed input, finger interaction, gesture recognition, navigation methods, stereo and surround sounds, and physics engine. In developing the UI, there are other important factors to keep in mind such as input device obstruction, programming interfaces, and error handling [28].

1) *Raycasting*: Raycasting is used in VR spaces as a type of user input and works similar to a laser beam detector. The laser is sent out by the user in a direction in 2D or 3D and if it hits elements that are designed to receive the raycast, it returns the information about what and where it hit [29]. Most VR gears (headsets) have predefined raycasting modules that can be reused and modified by the users. The attributes that can be modified are the origin, the direction, and maxDistance. Elements that are able to interact with the raycast require the Raycast Target attribute to be activated. A LayerMask can be defined to filter out any colliders that should not interact with the raycast.

2) *Events*: Unity handles the events through Bubbling. In Event Bubbling, when an event happens on an element, it first runs the handlers on that element, then on its parent, then all the way up on the ancestor [30]. In a VR project, events can be triggered by touching the collider on an element. Also, the user can be assigned a Raycast which will be used to interact with UI or other elements. A controller button press would then work as a mouse click. In Bubbling, the handler works until it handles that event all the way up. However, using Event Trigger can stop the bubbling and intercept all events. This can cause issues when there are nested UI elements with events such as OnPointer, OnPointerExit, and Click events, and should be used with caution.

III. CONCLUSION

In this paper, we have reviewed digital preservation using Virtual Reality. VR provides the means to transfer cultural, architectural, and historical assets in a way that is superior to the traditional methods. The use of audio, avatars, lighting, and different controllers have not been discussed here but are important components of realistic VR development. Basic steps of development and conceptual modeling of such projects were examined. However, considering the dynamic nature of project requirements, developers must continuously explore the new methods and tools in the field to provide the best end product.

REFERENCES

- [1] O. Soto-Martin, A. Fuentes-Porto, and J. Martin-Gutierrez, "A digital reconstruction of a historical building and virtual reintegration of mural paintings to create an interactive and immersive experience in virtual reality," *Appl. Sci.*, vol. 10, p. 597, 2020.
- [2] A. Ibañez-Etxeberria, C. Gómez-Carrasco, O. Fontal, and S. García-Ceballos, "Virtual environments and augmented reality applied to heritage education: an evaluative study," *Appl. Sci.*, vol. 10, p. 2352, 2020.
- [3] M. Obradović, I. Vasiljević, I. urić, J. Kićanović, V. Stojaković, and R. Obradović, "Virtual reality models based on photogrammetric surveys—a case study of the iconostasis of the serbian orthodox cathedral church of saint nicholas in sremski karlovci (serbia)," *Appl. Sci.*, vol. 10, p. 2743, 2020.
- [4] M. Roussou, "Virtual heritage: from the research lab to the broad public," *Bar International Series*, vol. 1075, pp. 93–100, 2002.
- [5] "Notre dame fire: Experts weigh in on timeline and cost of restoration." (Accessed on 11/05/2020).
- [6] D. Tuck, I. Kuksa, and R. Guest, "Narrating the past project: navigating around the story space," in *Proceedings of ICERI 09 International Conference of Education Research and Innovation*, 2009.
- [7] "Chorophrónesis — applied spatial intelligence." <https://chorophrónesis.psu.edu/>. (Accessed on 11/05/2020).
- [8] F. Remondino, "Heritage recording and 3d modeling with photogrammetry and 3d scanning," *Remote Sens.*, vol. 3, pp. 1104–1138, 2011.
- [9] R. Levy, "The role of laser scanning in the preservation and renovation of historic architecture," 01 2005.
- [10] K. Evangelidis, S. Sylaiou, and T. Papadopoulos, "Mergin' mode: Mixed reality and geoinformatics for monument demonstration," *Appl. Sci.*, vol. 10, p. 3826, 2020.
- [11] F. F. Ahmadi and H. Ebadi, "An integrated photogrammetric and spatial database management system for producing fully structured data using aerial and remote sensing images," *Sensors*, vol. 9, no. 4, pp. 2320–2333, 2009.
- [12] P. Charbonnier, P. Chavant, P. Foucher, V. Muzet, D. Prybyla, T. Perrin, P. Grussenmeyer, and S. Guillemin, "Accuracy assessment of a canal-tunnel 3d model by comparing photogrammetry and laserscanning recording techniques," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 12, pp. 2–6, 2013.
- [13] D. Meagher, "Geometric modeling using octree encoding," *Computer graphics and image processing*, vol. 19, no. 2, pp. 129–147, 1982.
- [14] L. P. Chew, "Constrained delaunay triangulations," *Algorithmica*, vol. 4, no. 1–4, pp. 97–108, 1989.
- [15] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.
- [16] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon mesh processing*. CRC press, 2010.
- [17] F. Remondino, "From point cloud to surface: the modeling and visualization problem," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, 2003.
- [18] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
- [19] F. Remondino, "Heritage recording and 3d modeling with photogrammetry and 3d scanning," *Remote sensing*, vol. 3, no. 6, pp. 1104–1138, 2011.
- [20] C. V. Tao and J. Li, *Advances in mobile mapping technology*, vol. 4. CRC Press, 2007.
- [21] "G2 research scoring methodologies." <https://research.g2.com/g2-scoring-methodologies>. (Accessed on 11/05/2020).
- [22] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5664–5673, 2019.
- [23] I. Millington, *Game physics engine development*. CRC Press, 2007.
- [24] S. D. Economics, "Developer economics: State of the developer nation q1 2016," 2016.
- [25] M. Farouk, G. Simon, and A. Ksentini, "Dissecting games engines: the case of unity3d," in *ACM Netgames 2015*, 2015.
- [26] J. Preece, H. Sharp, and Y. Rogers, *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2015.
- [27] M.-L. Ryan, "Immersion vs. interactivity: Virtual reality and literary theory," *SubStance*, vol. 28, no. 2, pp. 110–137, 1999.
- [28] T. M. Takala, P. Rauhamaa, and T. Takala, "Survey of 3d ui applications and development challenges," in *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 89–96, IEEE, 2012.
- [29] "Intro to vive development : Input through raycasting — circuit stream." <https://circuitstream.com/blog/htc-vive-tutorial-raycasting/>. (Accessed on 11/05/2020).
- [30] "Bubbling and capturing." <https://javascript.info/bubbling-and-capturing>. (Accessed on 11/05/2020).